

INTERNATIONAL STANDARD

NORME INTERNATIONALE



**Adjustable speed electrical power drive systems –
Part 7-202: Generic interface and use of profiles for power drive systems –
Profile type 2 specification**

**Entraînements électriques de puissance à vitesse variable –
Partie 7-202: Interface générique et utilisation de profils pour les entraînements
électriques de puissance – Specification du profil de type 2**



THIS PUBLICATION IS COPYRIGHT PROTECTED

Copyright © 2007 IEC, Geneva, Switzerland

All rights reserved. Unless otherwise specified, no part of this publication may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm, without permission in writing from either IEC or IEC's member National Committee in the country of the requester.

If you have any questions about IEC copyright or have an enquiry about obtaining additional rights to this publication, please contact the address below or your local IEC member National Committee for further information.

Droits de reproduction réservés. Sauf indication contraire, aucune partie de cette publication ne peut être reproduite ni utilisée sous quelque forme que ce soit et par aucun procédé, électronique ou mécanique, y compris la photocopie et les microfilms, sans l'accord écrit de la CEI ou du Comité national de la CEI du pays du demandeur.

Si vous avez des questions sur le copyright de la CEI ou si vous désirez obtenir des droits supplémentaires sur cette publication, utilisez les coordonnées ci-après ou contactez le Comité national de la CEI de votre pays de résidence.

IEC Central Office
3, rue de Varembe
CH-1211 Geneva 20
Switzerland

Tel.: +41 22 919 02 11
Fax: +41 22 919 03 00
info@iec.ch
www.iec.ch

About the IEC

The International Electrotechnical Commission (IEC) is the leading global organization that prepares and publishes International Standards for all electrical, electronic and related technologies.

About IEC publications

The technical content of IEC publications is kept under constant review by the IEC. Please make sure that you have the latest edition, a corrigenda or an amendment might have been published.

Useful links:

IEC publications search - www.iec.ch/searchpub

The advanced search enables you to find IEC publications by a variety of criteria (reference number, text, technical committee,...).

It also gives information on projects, replaced and withdrawn publications.

IEC Just Published - webstore.iec.ch/justpublished

Stay up to date on all new IEC publications. Just Published details all new publications released. Available on-line and also once a month by email.

Electropedia - www.electropedia.org

The world's leading online dictionary of electronic and electrical terms containing more than 30 000 terms and definitions in English and French, with equivalent terms in additional languages. Also known as the International Electrotechnical Vocabulary (IEV) on-line.

Customer Service Centre - webstore.iec.ch/csc

If you wish to give us your feedback on this publication or need further assistance, please contact the Customer Service Centre: csc@iec.ch.

A propos de la CEI

La Commission Electrotechnique Internationale (CEI) est la première organisation mondiale qui élabore et publie des Normes internationales pour tout ce qui a trait à l'électricité, à l'électronique et aux technologies apparentées.

A propos des publications CEI

Le contenu technique des publications de la CEI est constamment revu. Veuillez vous assurer que vous possédez l'édition la plus récente. un corrigendum ou amendement peut avoir été publié.

Liens utiles:

Recherche de publications CEI - www.iec.ch/searchpub

La recherche avancée vous permet de trouver des publications CEI en utilisant différents critères (numéro de référence, texte, comité d'études,...).

Elle donne aussi des informations sur les projets et les publications remplacées ou retirées.

Just Published CEI - webstore.iec.ch/justpublished

Restez informé sur les nouvelles publications de la CEI. Just Published détaille les nouvelles publications parues. Disponible en ligne et aussi une fois par mois par email.

Electropedia - www.electropedia.org

Le premier dictionnaire en ligne au monde de termes électroniques et électriques. Il contient plus de 30 000 termes et définitions en anglais et en français, ainsi que les termes équivalents dans les langues additionnelles. Egalement appelé Vocabulaire Electrotechnique International (VEI) en ligne.

Service Clients - webstore.iec.ch/csc

Si vous désirez nous donner des commentaires sur cette publication ou si vous avez des questions contactez-nous: csc@iec.ch.

INTERNATIONAL STANDARD

NORME INTERNATIONALE



**Adjustable speed electrical power drive systems –
Part 7-202: Generic interface and use of profiles for power drive systems –
Profile type 2 specification**

**Entraînements électriques de puissance à vitesse variable –
Partie 7-202: Interface générique et utilisation de profils pour les entraînements
électriques de puissance – Specification du profil de type 2**

INTERNATIONAL
ELECTROTECHNICAL
COMMISSION

COMMISSION
ELECTROTECHNIQUE
INTERNATIONALE

PRICE CODE
CODE PRIX

XH

ICS 29.200; 35.100.05

ISBN 978-2-83220-844-1

**Warning! Make sure that you obtained this publication from an authorized distributor.
Attention! Veuillez vous assurer que vous avez obtenu cette publication via un distributeur agréé.**

CONTENTS

FOREWORD.....	9
INTRODUCTION.....	11
1 Scope.....	14
2 Normative references	14
3 Terms, definitions and abbreviated terms	15
3.1 Terms and definitions	15
3.2 Abbreviated terms	21
4 General	21
4.1 General	21
4.2 Control modes	22
4.2.1 General	22
4.2.2 Control methods	22
4.2.3 Control nomenclature	22
4.2.4 Position Control	23
4.2.5 Velocity Control	24
4.2.6 Acceleration Control	26
4.2.7 Torque Control	26
4.2.8 No Control	27
5 Data types	28
5.1 Data type overview	28
5.2 Conventions	29
6 CIP Drive Profile	29
6.1 Object model	29
6.1.1 Object overview	29
6.1.2 Object description	30
6.2 How objects affect behaviour	30
6.3 Defining object interfaces	30
6.4 I/O Connection messages	31
6.4.1 General	31
6.4.2 CIP Motion I/O connection	31
6.4.3 Controller-to-Device connection	34
6.4.4 Device-to-Controller connection	46
6.4.5 Fixed Device connection format	53
6.4.6 CIP Motion connection timing	54
6.5 Drive startup procedure	63
6.5.1 General	63
6.5.2 Motion I/O Connection creation	63
6.5.3 Motion Axis Object configuration	63
6.5.4 Time Synchronisation	64
6.6 Device visualisation.....	65
6.7 Ethernet Quality of Service (QoS).....	65
7 Motion Axis Object	66
7.1 General considerations.....	66
7.1.1 General	66
7.1.2 Object overview	66

7.1.3	Motion Axis Object abstraction	66
7.1.4	Motion Control Axis Object	67
7.1.5	Drive control classification	67
7.1.6	Required vs. Optional in implementation	68
7.2	Class attributes	75
7.2.1	General	75
7.2.2	Semantics	79
7.3	Instance attributes	82
7.3.1	General	82
7.3.2	Motion Control configuration attributes	83
7.3.3	Motor attributes	85
7.3.4	Feedback attributes	92
7.3.5	Event Capture attributes	98
7.3.6	Command reference generation attributes	102
7.3.7	Control mode attributes	106
7.3.8	Stopping & Braking attributes	116
7.3.9	DC Bus Control attributes	119
7.3.10	Power and thermal management attributes	120
7.3.11	Axis Status Attributes	122
7.3.12	Exception, fault, and alarm status attributes	126
7.3.13	Exception limit attributes	131
7.3.14	Exception action configuration attribute	132
7.3.15	Initialisation fault status attributes	134
7.3.16	Start inhibit status attributes	136
7.3.17	Axis statistical attributes	137
7.3.18	Axis info attributes	138
7.3.19	General purpose I/O attributes	138
7.3.20	Local Mode Attributes	139
7.4	Common services	139
7.4.1	Supported services	139
7.4.2	Service specific data	140
7.5	Object specific services	141
7.5.1	Supported services	141
7.5.2	Service specific data	142
7.6	Behaviour	151
7.6.1	State model	151
7.6.2	State behaviour	154
7.6.3	Fault and alarm behaviour	157
7.6.4	Start Inhibit behaviour	158
7.6.5	Visualization behaviour	158
7.6.6	Command generation behaviour	163
7.6.7	Feedback Interface Behavior	166
7.6.8	Event Capture Behavior	169
7.6.9	Control Mode Behaviour	170
	Bibliography	181
	Figure 1 – Structure of IEC 61800-7	13

Figure 2 – Open loop position control	23
Figure 3 – Closed loop position control	24
Figure 4 – Open loop velocity control	25
Figure 5 – Closed loop velocity control	25
Figure 6 – Acceleration control	26
Figure 7 – Torque control.....	27
Figure 8 – No control	28
Figure 9 – Object Model for a CIP Motion drive.....	29
Figure 10 – CIP Motion I/O connection model	31
Figure 11 – Controller-to-Device Connection Format.....	32
Figure 12 – Device-to-Controller Connection Format.....	33
Figure 13 – CIP Motion I/O Connection Channels	34
Figure 14 – CIP Motion Controller-to-Device Connection Format	34
Figure 15 – Connection Header	35
Figure 16 – Connection Format.....	35
Figure 17 – Connection Header	36
Figure 18 – Instance Data Block	37
Figure 19 – Instance Data Header	38
Figure 20 – Cyclic Data Header	38
Figure 21 – Cyclic Write Data Block	43
Figure 22 – Event Data Block	43
Figure 23 – Service Data Block	45
Figure 24 – CIP Motion Controller-to-Device Connection Format	46
Figure 25 – Connection Header	46
Figure 26 – Connection Header	47
Figure 27 – Adjustment of actual position data based on device time stamp	48
Figure 28 – Instance Data Block	48
Figure 29 – Instance Data Header	49
Figure 30 – Cyclic Data Block	49
Figure 31 – Cyclic Read Data Block	50
Figure 32 – Event Data Block	51
Figure 33 – Service Data Block.....	52
Figure 34 – Fixed Controller to Device Connection Format (fixed size = 16 bytes)	53
Figure 35 – Fixed Device to Controller Connection Format (fixed size = 16 bytes)	53
Figure 36 – CIP Motion I/O Connection timing model	54
Figure 37 – Controller-to-Drive Connection timing with fine interpolation.....	55
Figure 38 – Controller-to-Drive Connection timing with unequal update periods	57
Figure 39 – Use of Time Stamp to adjust actual position to the controller's timebase	58
Figure 40 – Coordination of two drives with different Update Periods.	59
Figure 41 – Coordination of multiple drive axes in case of delayed Controller-to-Device Connection packets.....	60
Figure 42 – Propagation of a step change in time	61
Figure 43 – Group Sync of CIP Motion drives	64

Figure 44 – Object components for CIP motion architecture	66
Figure 45 – Controller Consumed Connection Data Format	79
Figure 46 – Controller Produced Connection Data Format	80
Figure 47 – Control Mode bit field	84
Figure 48 – IEEE per phase motor model	85
Figure 49 – Feedback Configuration bit field	97
Figure 50 – Event Checking Control word field	100
Figure 51 – Event Checking Status word field	101
Figure 52 – Interpolation Control word field	105
Figure 53 – Get Axis Attribute List Request Format	142
Figure 54 – Get Axis Attribute List Response Format	143
Figure 55 – Get Axis Attribute List Response – Example 1	144
Figure 56 – Get Axis Attribute List Response – Example 2	144
Figure 57 – Get Axis Attribute List Response – Example 3	144
Figure 58 – Get Axis Attribute List Response – Example 4	144
Figure 59 – Set Axis Attribute List Request Format	145
Figure 60 – Set Axis Attribute List Request – Example 1	146
Figure 61 – Set Axis Attribute List Request – Example 2	146
Figure 62 – Set Axis Attribute List Request – Example 3	146
Figure 63 – Set Axis Attribute List Response Format	146
Figure 64 – Set Cyclic Write List Request Format	147
Figure 65 – Set Cyclic Write List Response Format	147
Figure 66 – Set Cyclic Read List Request Format	148
Figure 67 – Set Cyclic Read List Response Format	148
Figure 68 – Motion Axis Object State Model	152
Figure 69 – Command Generator	163
Figure 70 – Feedback Channels 1 and 2	167
Figure 71 – Feedback Channels 3 and 4	168
Figure 72 – Event Capture Functionality	169
Figure 73 – No Control (Feedback Only)	171
Figure 74 – Closed Loop Position Control	172
Figure 75 – Closed Loop Velocity Control	174
Figure 76 – Open Loop Frequency Control	176
Figure 77 – Acceleration Control	177
Figure 78 – Torque Control	178
Figure 79 – Closed Loop Current Vector Control	179
Table 1 – Data types	28
Table 2 – Objects present in a CIP Motion drive device	30
Table 3 – Object effect on behavior	30
Table 4 – Object interfaces	30
Table 5 – Time Data Set	36
Table 6 – Axis Control	39

Table 7 – Command Data Set	39
Table 8 – Command Data Element to Motion Axis Object Attribute mapping	40
Table 9 – Actual Data Set	40
Table 10 – Actual Data Element to Motion Axis Object Attribute Mapping	41
Table 11 – Status Data Set configuration	41
Table 12 – Interpolation Control.....	42
Table 13 – Axis Response	50
Table 14 – Event Type	52
Table 15 – Propagation of a step change in time (example 1)	61
Table 16 – Propagation of a step change in time (example 2)	63
Table 17 – CIP Motion visualisation components	65
Table 18 – Instance attribute implementation vs. Device Control Code	68
Table 19 – Class attributes for the Motion Axis Object	77
Table 20 – Node Control bit definitions	80
Table 21 – Node Status bit definitions.....	81
Table 22 – Node Faults code definitions	81
Table 23 – Node Alarms code definitions	82
Table 24 – Dynamic Unit vs. Feedback Configuration	83
Table 25 – Motion Control configuration attributes	83
Table 26 – Motor Control field enumeration definitions.....	84
Table 27 –Control Method field enumeration definitions	84
Table 28 – General Motor Info Attributes	86
Table 29 – General Motor Configuration Attributes.....	86
Table 30 – General PM Motor Configuration Attributes	89
Table 31 – General Rotary Motor Configuration Attributes	89
Table 32 – General Linear Motor Configuration Attributes	90
Table 33 – Rotary PM Motor Configuration Attributes.....	90
Table 34 – Linear PM Motor Configuration Attributes	91
Table 35 – Induction Motor Configuration Attributes.....	91
Table 36 – General Feedback Info Attributes	92
Table 37 – Feedback Configuration Attributes.....	93
Table 38 – Feedback Selection field enumeration definitions	98
Table 39 – Event Attributes.....	99
Table 40 – Event Checking Control bit definitions	100
Table 41 – Event Checking Status bit definitions.....	101
Table 42 – Command Generator Signal Attributes.....	102
Table 43 – Command Generator Configuration Attributes	103
Table 44 – Command Target Update enumeration definition	105
Table 45 – Position Loop Signal Attributes.....	106
Table 46 – Position Loop Configuration Attributes.....	107
Table 47 – Velocity Loop Signal Attributes	108
Table 48 – Velocity Loop Configuration Attributes	108
Table 49 – Acceleration Signal Attributes.....	110

Table 50 – Torque/Force Reference Signal Attributes	110
Table 51 – Torque/Force Reference Configuration Attributes	111
Table 52 – Current Loop Signal Attributes.....	112
Table 53 – Current Loop Configuration Attributes	114
Table 54 – Frequency Control Signal Attributes	114
Table 55 – Frequency Control Configuration Attributes	115
Table 56 – Drive Output Attributes	115
Table 57 – Stopping/Braking Attributes	116
Table 58 – Stopping Mode enumeration definitions	117
Table 59 – DC Bus Control Attributes	119
Table 60 – Power and Thermal Management Status Attributes	121
Table 61 – Power and Thermal Management Configuration Attributes.....	122
Table 62 – Axis Status Attributes	123
Table 63 – Axis Status bit definitions	124
Table 64 – Axis I/O Status bit definitions.....	124
Table 65 –Status Data Set bit definitions	125
Table 66 – Exception, Fault and Alarm Status Attributes.....	126
Table 67 – Standard Exception Table	129
Table 68 – Exception Factory Limit Info Attributes	131
Table 69 – Exception User Limit Configuration Attributes	132
Table 70 – Exception Action Configuration Attribute.....	133
Table 71 – Exception Action Bit Definitions.....	133
Table 72 – Initialisation Fault Status Attributes	134
Table 73 – Standard Initialisation Fault Table	136
Table 74 – Start Inhibit Status Attributes.....	136
Table 75 – Standard Start Inhibit Table.....	137
Table 76 –Statistical Attributes	137
Table 77 – Axis Info Attributes	138
Table 78 – General Purpose I/O Attributes.....	138
Table 79 – Local Mode Configuration Attributes	139
Table 80 – Motion Axis Object – Common Services	139
Table 81 – Group_Sync Request Data Structure	140
Table 82 – Group_Sync Response Data Structure	140
Table 83 – Motion Axis Object – Object Specific Services.....	141
Table 84 – Run Motor Test Request Data Structure	149
Table 85 – Get Motor Test Response Data Structure	149
Table 86 – Run Inertia Test Request Data Structure	150
Table 87 – Get Inertia Test Response Data Structure	150
Table 88 – Run Hookup Test Request Data Structure	151
Table 89 – Get Hookup Test Response Data Structure	151
Table 90 – Axis Control Request Code	152
Table 91 – Axis Response Acknowledge Codes	153
Table 92 – Completion Criteria for Requested Operation	153

Table 93 – Successful Axis Control Request Cycle	154
Table 94 – Unsuccessful Axis Control Request Cycle	154
Table 95 – Axis State Mapping to Identity Object with LED Behavior	159
Table 96 – CIP Motion Device Seven-Segment Display Behavior	160
Table 97 – CIP Motion Multi-Character Alphanumeric Display Behavior	161
Table 98 – Multi-Axis Multi-Character Alphanumeric Display Behavior	162

IECNORM.COM: Click to view the full PDF of IEC 61800-7-202:2007

Withd2W

INTERNATIONAL ELECTROTECHNICAL COMMISSION

ADJUSTABLE SPEED ELECTRICAL POWER DRIVE SYSTEMS –**Part 7-202: Generic interface and use
of profiles for power drive systems –
Profile type 2 specification****FOREWORD**

- 1) The International Electrotechnical Commission (IEC) is a worldwide organization for standardization comprising all national electrotechnical committees (IEC National Committees). The object of IEC is to promote international co-operation on all questions concerning standardization in the electrical and electronic fields. To this end and in addition to other activities, IEC publishes International Standards, Technical Specifications, Technical Reports, Publicly Available Specifications (PAS) and Guides (hereafter referred to as "IEC Publication(s)"). Their preparation is entrusted to technical committees; any IEC National Committee interested in the subject dealt with may participate in this preparatory work. International, governmental and non-governmental organizations liaising with the IEC also participate in this preparation. IEC collaborates closely with the International Organization for Standardization (ISO) in accordance with conditions determined by agreement between the two organizations.
- 2) The formal decisions or agreements of IEC on technical matters express, as nearly as possible, an international consensus of opinion on the relevant subjects since each technical committee has representation from all interested IEC National Committees.
- 3) IEC Publications have the form of recommendations for international use and are accepted by IEC National Committees in that sense. While all reasonable efforts are made to ensure that the technical content of IEC Publications is accurate, IEC cannot be held responsible for the way in which they are used or for any misinterpretation by any end user.
- 4) In order to promote international uniformity, IEC National Committees undertake to apply IEC Publications transparently to the maximum extent possible in their national and regional publications. Any divergence between any IEC Publication and the corresponding national or regional publication shall be clearly indicated in the latter.
- 5) IEC provides no marking procedure to indicate its approval and cannot be rendered responsible for any equipment declared to be in conformity with an IEC Publication.
- 6) All users should ensure that they have the latest edition of this publication.
- 7) No liability shall attach to IEC or its directors, employees, servants or agents including individual experts and members of its technical committees and IEC National Committees for any personal injury, property damage or other damage of any nature whatsoever, whether direct or indirect, or for costs (including legal fees) and expenses arising out of the publication, use of, or reliance upon, this IEC Publication or any other IEC Publications.
- 8) Attention is drawn to the Normative references cited in this publication. Use of the referenced publications is indispensable for the correct application of this publication.
- 9) Attention is drawn to the possibility that some of the elements of this IEC Publication may be the subject of patent rights. IEC shall not be held responsible for identifying any or all such patent rights.

The International Electrotechnical Commission (IEC) draws attention to the fact that it is claimed that compliance with this document may involve the use of a patent concerning the following:

Publication/ Application serial number	Holder	Title
US 11/241,539	[RA]	Time Stamped Motion Control Network Protocol That Enables Balanced Single Cycle Timing and Utilization of Dynamic Data Structures

The IEC takes no position concerning the evidence, validity and scope of this patent right.

The holder of this patent right has assured the IEC that he is willing to negotiate licences under reasonable and non-discriminatory terms and conditions with applicants throughout the world. In this respect, the statement of the holder of this patent right is registered with IEC. Information may be obtained from

[RA]	Rockwell Automation, Inc. 1201 S. Second Street Milwaukee, WI 53204 USA Attention: Intellectual Property Dept.

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights other than those identified above. IEC shall not be held responsible for identifying any or all such patent rights.

The International Standard IEC 61800-7-202 has been prepared by subcommittee SC 22G: Adjustable speed electric drive systems incorporating semiconductor power converters, of IEC technical committee TC 22: Power electronic systems and equipment.

This bilingual version (2013-05) corresponds to the monolingual English version, published in 2007-11.

The text of this standard is based on the following documents:

FDIS	Report on voting
22G/184/FDIS	22G/192/RVD

Full information on the voting for the approval of this standard can be found in the report on voting indicated in the above table.

The French version of this standard has not been voted upon.

This publication has been drafted in accordance with the ISO/IEC Directives, Part 2.

A list of all parts of the IEC 61800 series, under the general title *Adjustable speed electrical power drive systems*, can be found on the IEC website.

The committee has decided that the contents of this publication will remain unchanged until the maintenance result date indicated on the IEC web site under "<http://webstore.iec.ch>" in the data related to the specific publication. At this date, the publication will be

- reconfirmed;
- withdrawn;
- replaced by a revised edition, or
- amended.

IMPORTANT – The 'colour inside' logo on the cover page of this publication indicates that it contains colours which are considered to be useful for the correct understanding of its contents. Users should therefore print this document using a colour printer.

INTRODUCTION

The IEC 61800 series is intended to provide a common set of specifications for adjustable speed electrical power drive systems.

IEC 61800-7 describes a generic interface between control systems and power drive systems. This interface can be embedded in the control system. The control system itself can also be located in the drive (sometimes known as "smart drive" or "intelligent drive").

A variety of physical interfaces is available (analogue and digital inputs and outputs, serial and parallel interfaces, fieldbuses and networks). Profiles based on specific physical interfaces are already defined for some application areas (e.g. motion control) and some device classes (e.g. standard drives, positioner). The implementations of the associated drivers and application programmers interfaces are proprietary and vary widely.

IEC 61800-7 defines a set of common drive control functions, parameters, and state machines or description of sequences of operation to be mapped to the profiles.

IEC 61800-7 provides a way to access functions and data of a drive that is independent of the used drive profile and communication interface. The objective is a common drive model with generic functions and objects suitable to be mapped on different communication interfaces. This makes it possible to provide common implementations of motion control (or velocity control or drive control applications) in controllers without any specific knowledge of the drive implementation.

There are several reasons to define a generic interface.

For a drive device manufacturer

- Less effort to support system integrators
- Less effort to describe drive functions because of common terminology
- The selection of drives does not depend on availability of specific support

For a control device manufacturer

- No influence of bus technology
- Easy device integration
- Independent of a drive supplier

For a system integrator (builds modules, machines, plants etc.)

- Less integration effort for devices
- Only one understandable way of modeling
- Independent of bus technology

Much effort is needed to design a motion control application with several different drives and a specific control system. The tasks to implement the system software and to understand the functional description of the individual components may exhaust the project resources. In some cases, the drives do not share the same physical interface. Some control devices just support a single interface which will not be supported by a specific drive. On the other hand, the functions and data structures are specified with incompatibilities. It is up to the systems integrator to write interfaces to the application software to handle that which should not be his responsibility.

Some applications need device exchangeability or integration of new devices in an existing configuration. They are faced with different incompatible solutions. The efforts to adopt a solution to a drive profile and to manufacturer specific extensions may be unacceptable. This will reduce the degree of freedom to select a device best suited for this application to the selection of the unit which will be available for a specific physical interface and supported by the controller.

IEC 61800-7-1 is divided into a generic part and several annexes as shown in Figure 1. The drive profile types for CiA 402¹, CIP Motion^{TM2}, PROFIdrive³ and SERCOS Interface^{TM4} are mapped to the generic interface in the corresponding annex. The annexes have been submitted by open international network or fieldbus organizations which are responsible for the content of the related annex and use of the related trademarks.

This part of IEC 61800-7 specifies the profile type 2 (CIP MotionTM).

The profile types 1, 3 and 4 are specified in IEC 61800-7-201, IEC 61800-7-203 and IEC 61800-7-204.

IEC 61800-7-301, IEC 61800-7-302, IEC 61800-7-303 and IEC 61800-7-304 specify how the profile types 1, 2, 3 and 4 are mapped to different network technologies (such as CANopen⁵, EtherCAT^{TM6}, Ethernet Powerlink^{TM7}, DeviceNet^{TM8}, ControlNet^{TM9}, EtherNet/IP^{TM10}, PROFIBUS¹¹, PROFINET¹² and SERCOS Interface).

- 1 CiA 402 is a trade name of CAN in Automation, e.V. This information is given for the convenience of users of this International Standard and does not constitute an endorsement by IEC of the trade name holder or any of its products. Compliance to this profile does not require use of the trade name CiA 402.
- 2 CIP MotionTM is a trade name of Open DeviceNet Vendor Association, Inc. This information is given for the convenience of users of this International Standard and does not constitute an endorsement by IEC of the trademark holder or any of its products. Compliance to this profile does not require use of the trade name CIP MotionTM. Use of the trade name CIP MotionTM requires permission of Open DeviceNet Vendor Association, Inc.
- 3 PROFIdrive is a trade name of PROFIBUS International. This information is given for the convenience of users of this International Standard and does not constitute an endorsement by IEC of the trade name holder or any of its products. Compliance to this profile does not require use of the trade name PROFIdrive. Use of the trade name PROFIdrive requires permission of PROFIBUS International.
- 4 SERCOSTM and SERCOS InterfaceTM are trade names of SERCOS International e.V. This information is given for the convenience of users of this International Standard and does not constitute an endorsement by IEC of the trade name holder or any of its products. Compliance to this profile does not require use of the trade name SERCOS and SERCOS interface. Use of the trade name SERCOS and SERCOS interface requires permission of the trade name holder.
- 5 CANopen is an acronym for Controller Area Network open and is used to refer to EN 50325-4.
- 6 EtherCATTM is a trade name of Beckhoff, Verl. This information is given for the convenience of users of this International Standard and does not constitute an endorsement by IEC of the trademark holder or any of its products. Compliance to this profile does not require use of the trade name EtherCATTM. Use of the trade name EtherCATTM requires permission of the trade name holder.
- 7 Ethernet PowerlinkTM is a trade name of B&R, control of trade name use is given to the non profit organisation EPSG. This information is given for the convenience of users of this International Standard and does not constitute an endorsement by IEC of the trademark holder or any of its products. Compliance to this profile does not require use of the trade name Ethernet PowerlinkTM. Use of the trade name Ethernet PowerlinkTM requires permission of the trade name holder.
- 8 DeviceNetTM is a trade name of Open DeviceNet Vendor Association, Inc. This information is given for the convenience of users of this International Standard and does not constitute an endorsement by IEC of the trademark holder or any of its products. Compliance to this profile does not require use of the trade name DeviceNetTM. Use of the trade name DeviceNetTM requires permission of Open DeviceNet Vendor Association, Inc.
- 9 ControlNetTM is a trade name of ControlNet International, Ltd. This information is given for the convenience of users of this International Standard and does not constitute an endorsement by IEC of the trademark holder or any of its products. Compliance to this profile does not require use of the trade name ControlNetTM. Use of the trade name ControlNetTM requires permission of ControlNet International, Ltd.
- 10 EtherNet/IPTM is a trade name of ControlNet International, Ltd. and Open DeviceNet Vendor Association, Inc. This information is given for the convenience of users of this International Standard and does not constitute an endorsement by IEC of the trademark holder or any of its products. Compliance to this profile does not require use of the trade name EtherNet/IPTM. Use of the trade name EtherNet/IPTM requires permission of either ControlNet International, Ltd. or Open DeviceNet Vendor Association, Inc.
- 11 PROFIBUS is a trade name of PROFIBUS International. This information is given for the convenience of users of this International Standard and does not constitute an endorsement by IEC of the trade name holder or any of its products. Compliance to this profile does not require use of the trade name PROFIBUS. Use of the trade name PROFIBUS requires permission of PROFIBUS International.
- 12 PROFINET is a trade name of PROFIBUS International. This information is given for the convenience of users of this International Standard and does not constitute an endorsement by IEC of the trade name holder or any of its products. Compliance to this profile does not require use of the trade name PROFINET. Use of the trade name PROFINET requires permission of PROFIBUS International.

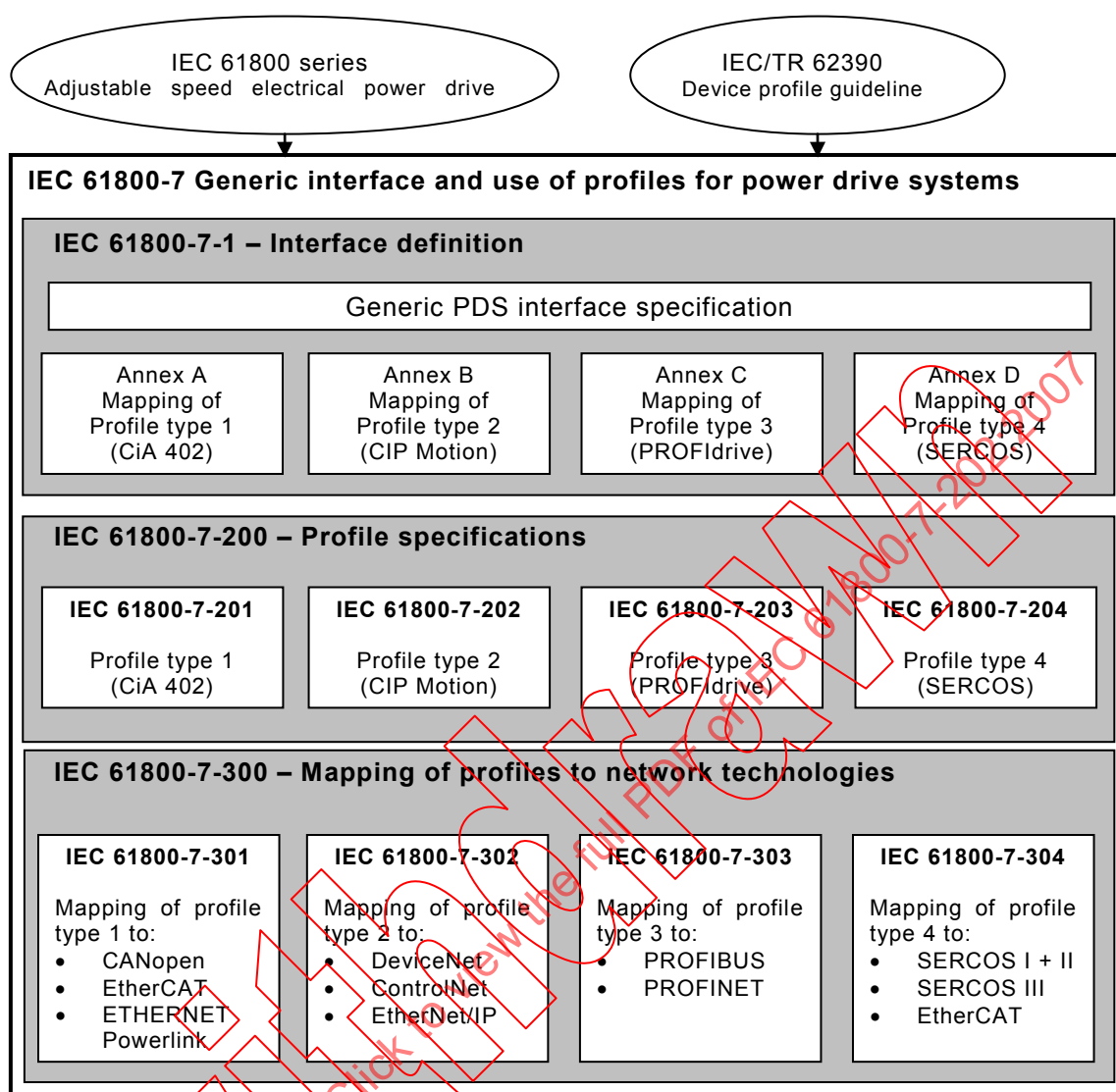


Figure 1 – Structure of IEC 61800-7

ADJUSTABLE SPEED ELECTRICAL POWER DRIVE SYSTEMS –

Part 7-202: Generic interface and use of profiles for power drive systems – Profile type 2 specification

1 Scope

IEC 61800-7 specifies profiles for Power Drive Systems (PDS) and their mapping to existing communication systems by use of a generic interface model.

The functions specified in this part of IEC 61800-7 are not intended to ensure functional safety. This requires additional measures according to the relevant standards, agreements and laws.

This part of IEC 61800-7 specifies profile type 2 (CIP Motion™) for Power Drive Systems (PDS). Profile type 2 can be mapped onto different communication network technologies.

2 Normative references

The following referenced documents are indispensable for the application of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

IEC 60204-1, *Safety of machinery – Electrical equipment of machines – Part 1: General requirements*

IEC 61131-3, *Programmable controllers – Part 3: Programming languages*

IEC 61158-5-2, *Industrial communication networks – Fieldbus specifications – Part 5-2 (Ed.1.0): Application layer service definition – Type 2 elements*

IEC 61158-6-2, *Industrial communication networks – Fieldbus specifications – Part 6-2 (Ed.1.0): Application layer protocol specification – Type 2 elements*

IEC 61588:2004, *Precision clock synchronization protocol for networked measurement and control systems*

IEC 61800-7 (all parts), *Adjustable speed electrical power drive systems – Generic interface and use of profiles for power drive systems*

IEC 61800-7-1, *Adjustable speed electrical power drive systems – Part 7-1: Generic interface and use of profiles for power drive systems – Interface definition*

IEEE 802.1Q, *IEEE standards for local and metropolitan area networks – Virtual bridged local area networks*

3 Terms, definitions and abbreviated terms

3.1 Terms and definitions

For the purposes of this document, the following terms and definitions apply.

3.1.1

algorithm

completely determined finite sequence of operations by which the values of the output data can be calculated from the values of the input data

[IEV 351-21-37]

3.1.2

application

software functional element specific to the solution of a problem in industrial-process measurement and control

NOTE An application may be distributed among resources, and may communicate with other applications.

[IEC/TR 62390:2005, 3.1.2, modified]

3.1.3

application mode

type of application that can be requested from a PDS

NOTE The different application modes reflect the control loop for torque control, velocity control, position control or other applications like homing.

3.1.4

attribute

property or characteristic of an entity

[IEC/TR 62390:2005, 3.1.3]

3.1.5

axis

logical element of a motion control system that exhibits some form of movement

NOTE 1 Axes can be rotary or linear, physical or virtual, controlled or simply observed.

NOTE 2 A physical axis may include one or more of the following components: a motion sensor, a motion control structure, a power amplifier, and a motion actuator.

3.1.6

bus regulator

any method used to limit the rise in DC bus voltage level that occurs when decelerating a motor

3.1.7

CIP Motion™¹³

extensions to the CIP services and protocol to support motion control over CIP networks

¹³ CIP Motion™ and CIP Sync™ are trade names of Open DeviceNet Vendor Association, Inc. This information is given for the convenience of users of this International Standard and does not constitute an endorsement by IEC of the trademark holder or any of its products. Compliance to this profile does not require use of the trade names CIP Motion™ or CIP Sync™. Use of the trade names CIP Motion™ or CIP Sync™ requires permission of Open DeviceNet Vendor Association, Inc.

3.1.8

CIP (Motion) drive

drive that complies with the CIP (Motion) specification

3.1.9

CIP Motion drive profile

collection of objects used to implement a CIP Motion drive device that includes the Motion Axis Object, as well as standard support objects like the Identity Object and the Time Sync Object

3.1.10

CIP Motion I/O connection

periodic bi-directional, class 1, CIP connection between a controller and a drive that is defined as part of the CIP Motion specification

3.1.11

CIP Sync^{TM13}

extensions to the CIP services and protocol to encapsulate IEC 61588:2004 time synchronisation functionality over a CIP Network (see Time Sync Object in IEC 61158-5-2 and IEC 61158-6-2)

3.1.12

class

description of a set of objects that share the same attributes, operations, methods, relationships, and semantics

[ISO/IEC 19501, modified]

3.1.13

closed loop

methods of control where there is a feedback signal of some kind that is used to drive the actual dynamics of the motor to match the commanded dynamics by servo action

NOTE In most cases, there is a literal feedback device to provide this signal, but in some cases, the signal is derived from the motor excitation (i.e. sensorless operation).

3.1.14

commands

set of commands from the application control program to the PDS to control the behaviour of the PDS or functional elements of the PDS

NOTE 1 The behaviour is reflected by states or operating modes.

NOTE 2 The different commands may be represented by one bit each.

3.1.15

control

purposeful action on or in a process to meet specified objectives

[IEV 351-21-29, modified]

3.1.16

control device

physical unit that contains – in a module/subassembly or device – an application program to control the PDS

3.1.17

converter

device that generally converts AC input to DC output

NOTE A converter is also commonly called the Drive Power Supply. In the context of a drive system, the converter is responsible for converting AC main input into DC bus power.

3.1.18

cyclic data block

high priority real-time data block that is transferred by a CIP Motion connection on a periodic basis

3.1.19

data type

set of values together with a set of permitted operations

[ISO/IEC 2382-15:1999, 15.04.01, modified]

3.1.20

device

field device

networked independent physical entity of an industrial automation system capable of performing specified functions in a particular context and delimited by its interfaces

[IEC 61499-1:2005, 3.30, modified]

entity that performs control, actuating and/or sensing functions and interfaces to other such entities within an automation system

[ISO 15745-1:2003, 3.11]

3.1.21

device profile

representation of a device in terms of its parameters, parameter assemblies and behaviour according to a device model that describes the device's data and behaviour as viewed through a network

NOTE This is a definition from IEC/TS 61915 which is extended by the addition of the device functional structure.

[IEC/TR 62390:2005, 3.19, modified]

3.1.22

drive

device designed to control the dynamics of a motor

3.1.23

event data block

medium priority real-time data block that is transferred by a CIP Motion connection only after a specified event occurs

NOTE Registration and marker input transitions are typical drive events.

3.1.24

functional element

entity of software or software combined with hardware, capable of accomplishing a specified function of a device

NOTE 1 A functional element has an interface, associations to other functional elements and functions.

NOTE 2 A functional element can be made out of function block(s), object(s) or parameter list(s).

[IEC/TR 62390:2005, 3.1.12]

3.1.25

get/read

operation that involves the retrieving of an attribute value from the perspective of the controller side of the interface

3.1.26

input data

data transferred from an external source into a device, resource or functional element

[IEC/TR 62390:2005, 3.1.14]

3.1.27

interface

shared boundary between two entities defined by functional characteristics, signal characteristics, or other characteristics as appropriate

[IEV 351-21-35, modified]

3.1.28

inverter

device that generally converts DC input to AC output

NOTE An inverter is also commonly called the Drive Amplifier. In the context of a drive system, the inverter is responsible for controlling the application of DC bus power to an AC motor.

3.1.29

model

mathematical or physical representation of a system or a process, based with sufficient precision upon known laws, identification or specified suppositions

[IEV 351-21-36]

3.1.30

motion

any aspect of the dynamics of an axis

NOTE In the context of this part of IEC 61800-7, it is not limited to servo drives but encompasses all forms of drive based motor control.

3.1.31

Motion Axis Object

object that defines the attributes, services, and behaviour of a motion device based axis according to the CIP Motion specification

3.1.32

Motion Control Axis Object

object that defines the attributes, services, and behaviour of a controller based axis according to the CIP Motion specification

3.1.33

open loop

methods of control where there is no application of feedback to force the actual motor dynamics to match the commanded dynamics

EXAMPLE Examples of open loop control are stepper drives and variable frequency drives.

3.1.34**operating mode**

characterisation of the way and the extent to which the human operator intervenes in the control equipment

[IEV 351-31-01]

3.1.35**output data**

data originating in a device, resource or functional element and transferred from them to external systems

[IEC/TR 62390:2005, 3.1.21]

3.1.36**parameter**

data element that represents device information that can be read from or written to a device, for example through the network or a local HMI

NOTE 1 Adapted from IEC/TS 61915.

NOTE 2 A parameter is typically characterised by a parameter name, data type and access direction.

[IEC/TR 62390:2005, 3.1.22, modified]

3.1.37**profile**

representation of a PDS interface in terms of its parameters, parameter assemblies and behaviour according to a communication profile and a device profile

3.1.38**service data block**

lower priority real-time data block associated with a service message from the controller that is transferred by a CIP Motion connection on a periodic basis

NOTE Service data includes service request messages to access Motion Axis Object attributes or perform various drive diagnostics.

3.1.39**set-point**

(in the context of a device) value sent to the drive that is used to directly control some aspect of the motor dynamics, which includes (but is not limited to) position, velocity, acceleration, and torque

3.1.40**set/write**

operation that involves the setting of an attribute to a specified value from the perspective of the controller side of the interface

3.1.41**shunt regulator**

specific bus regulator method that switches the DC bus across a power dissipating resistor to dissipate the regenerative power of a decelerating motor

3.1.42**status**

set of information from the PDS to the application control program reflecting the state or mode of the PDS or a functional element of the PDS

NOTE The different status information may be coded with one bit each.

3.1.43

synchronised

condition where the local clock value on the drive is locked onto the master clock of the distributed System Time

NOTE When synchronised, the drive and controller devices may utilise time stamps associated with CIP Motion connection data.

3.1.44

System Time

absolute time value as defined in the CIP Sync specification in the context of a distributed time system where all devices have a local clock that is synchronised with a common master clock

NOTE In the context of CIP Motion, System Time is a 64-bit integer value in units of nanoseconds with a value of 0 corresponding to the date 1970-01-01.

3.1.45

thermal model

any algorithm that attempts to model the thermal behaviour of the associated device during operation

3.1.46

time offset

System Time offset value associated with the CIP Motion connection data that is associated with source device

NOTE The System Time offset is a 64-bit offset value that is added to a device's local clock to generate System Time for that device

3.1.47

time stamp

System Time stamp value associated with the CIP Motion connection data that conveys the absolute time when the associated data was captured, or that can also be used to determine when the associated data shall be applied

3.1.48

type

hardware or software element which specifies the common attributes shared by all instances of the type

[IEC/TR 62390:2005, 3.1.25]

3.1.49

variable

software entity that may take different values, one at a time

[IEC/TR 62390:2005, 3.1.27]

NOTE The values of a variable as well as of a parameter are usually restricted to a certain data type.

3.1.50

Variable Frequency Drive

VFD

class of drive products that seek to control the speed of a motor, typically an induction motor, through a proportional relationship between drive output voltage and commanded output frequency

NOTE Variable Frequency Drives are therefore sometimes referred to as a Volts/Hertz drives.

3.1.51**vector drive**

class of drive products that seek to control the dynamics of a motor via closed loop control which includes, but is not limited to, closed loop control of both torque and flux vector components of motor stator current relative to the rotor flux vector

3.2 Abbreviated terms

Ack	Acknowledge
Act	Actual
Blk	Block
CIP™	Common Industrial Protocol (see IEC 61158 Type 2, IEC 61784-1 and IEC 61784-2 CPF2)
Cmd	Command
Cyc	Cyclic
HMI	Human Machine Interface
ID	Identifier
I/O	Input/Output
Opt.	Optional
PDS	Power Drive System
Req.	Required
RMS	Root mean square
RPM	Revolutions per minute
Sync'd	Synchronized
TCP	Transmission Control Protocol (see RFC 793)
UDP	User Datagram Protocol (see RFC 768)
VFD	Variable Frequency Drive

4 General**4.1 General**

A CIP Motion drive device controls the motion of one or more drive axes, typically consisting of a rotary or linear motor actuator and one or more optional feedback devices. Each associated motor is driven by an associated drive power structure that is also part of the device. Motor control functionality supported by the CIP Motion drive device profile can be applied to “open loop” Variable Frequency Drives or “closed loop” Vector Controlled Servo Drives. In either case, axis motion is controlled via a command reference that can be configured for position control, velocity control, acceleration control, or current/torque control. The CIP Motion drive device profile also supports use of additional drive feedback channels as master feedback sources.

This CIP Motion drive profile specification and the associated Motion Axis Object specification define the interface, the specific attributes, and command behaviours required to support motion control when connected to a CIP Motion compliant controller through a CIP network.

Applicable CIP Networks are DeviceNet™, ControlNet™, and EtherNet/IP™. EtherNet/IP is the network of choice for high performance, synchronised multi-axis control. ControlNet and DeviceNet would only be targeted for lower performance non-synchronised device applications such as simple variable frequency drives, velocity loop drives, and indexing drives.

4.2 Control modes

4.2.1 General

The Motion Axis Object covers the behavior of various motion control system devices that includes feedback devices and drive devices. For drive devices, the Motion Axis Object covers a wide range of drive types from simple variable frequency (V/Hz) drives, to sophisticated position control servo drives. Indeed, many commercial drive products have axes that can be configured to operate in any one of these different motion control modes depending on the specific application requirement.

The attributes of the Motion Axis Object are therefore organized to address this broad range of functionality, using the framework of “applications modes” defined in IEC 61800-7-1, and further detailed in the following subclauses.

IEC 61800-7-1 and this part of the IEC 61800-7 series are organised around the general philosophy that position control is the highest form of dynamic control. That is, position control implies velocity control, and velocity control implies acceleration control. Acceleration is related to torque or force by the inertia or mass of the load, respectively, acceleration control implies torque control. And finally, since motor torque or force is generally related to motor current by a torque or force constant, respectively, torque control implies current control. The torque or force constant can be a function of the motor magnets as in a Permanent Magnet motor, or the induced flux of an Induction motor.

Since acceleration, torque/force, and current are generally related by a constant, these terms are sometimes used interchangeably in the industry. For example IEC 61800-7-1 refers to a torque control loop rather than a current control loop. This specification attempts to differentiate between these control properties where applicable. This is particularly useful when the relationship between them is not static, such as when inertia/mass changes with position or time, or when the torque/force constant changes due to temperature change or motor flux variation.

4.2.2 Control methods

Within this basic control paradigm, there is latitude for different control methods, both closed loop and open loop. By closed loop, it is generally implied that there is a feedback signal that is used to drive the actual dynamics of the motor to match the commanded dynamics by servo action. In most cases, there is a literal feedback device to provide this signal, and in some cases, the signal is derived from the motor excitation (i.e. sensorless operation). By open loop, it is implied that there is no application of feedback to force the actual dynamics to match the commanded dynamics. While precision and performance are the hallmarks of closed loop control, simplicity and economy are the hallmarks of open loop control.

4.2.3 Control nomenclature

Finally, as evident in the description above, linear and rotary control applications can affect the control nomenclature. While rotary applications speak of Torque and Inertia, linear applications speak of Force and Mass. For the purposes of this part of the IEC 61800-7 series, when referring to rotary nomenclature, the defined behaviour can generally be applied to linear applications by substituting the terms, force for torque and mass for inertia. With that understanding, Figure 2 to Figure 8 follow the IEC 61800-7-1 nomenclature using torque rather than force without loss of generality.

4.2.4 Position Control

4.2.4.1 General concepts

In position control application mode, either the application control program (command execution function) or the motion planner (move trajectory control function) provide a set-point value to the CIP Motion device via the cyclic data connection. The position control method can be either open loop or closed loop.

4.2.4.2 Open Loop Position Control

A device configured for open loop position control applies to a class of drive devices called stepper drives. This type of drive is illustrated in Figure 2.

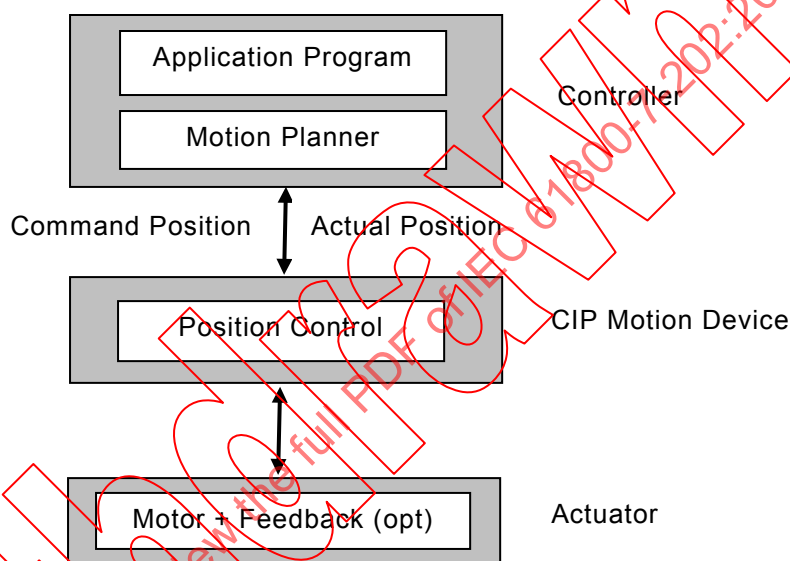


Figure 2 – Open loop position control

A feedback device for this configuration is optional. In the absence of a feedback device, actual position can be estimated by the drive and returned to the controller.

4.2.4.3 Closed Loop Position Control

A motion control device configured for closed loop position control is traditionally referred to as position loop drive or position servo drive. A position servo drive implies an inner velocity and torque control loop as shown in Figure 3. The presence of the torque/current control loop sometimes results in this kind of drive being referred to as a vector drive.

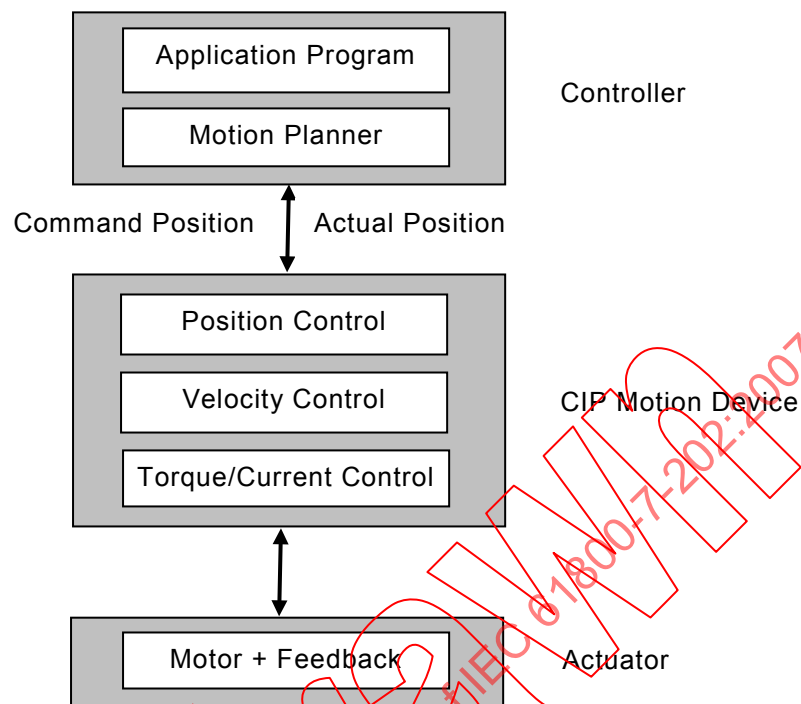


Figure 3 – Closed loop position control

A feedback device for this configuration is generally required to achieve good positioning accuracy. The feedback device may also be used to return Actual Velocity, and Actual Acceleration data to the controller via the cyclic data connection.

In addition to Command Position, the controller can pass Command Velocity and Command Acceleration for the purposes of forward control.

4.2.5 Velocity Control

4.2.5.1 General concepts

In velocity control application mode, the application control program and motion planner provide a set-point value to the CIP Motion device via the cyclic data connection. The velocity control method can be either open loop or closed loop.

4.2.5.2 Open Loop Velocity Control

A motion control device configured for open loop velocity control is traditionally referred to as Variable Frequency, or V/Hz, or VFD, drive. This type of drive is illustrated in Figure 4.

A feedback device for this configuration is optional. In the absence of a feedback device, actual velocity can be estimated by the drive and returned to the controller.

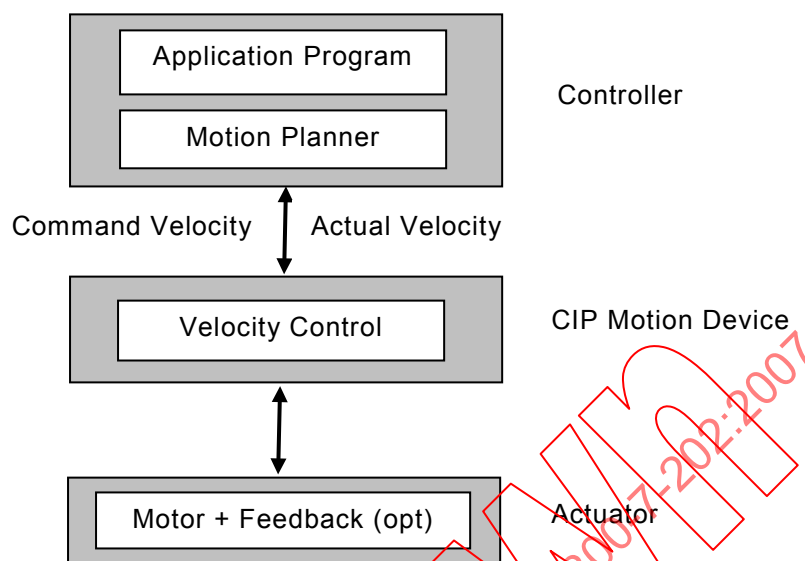


Figure 4 – Open loop velocity control

4.2.5.3 Closed Loop Velocity Control

A motion control device configured for closed loop velocity control is traditionally referred to as velocity loop drive or velocity servo drive. A closed loop velocity control drive implies an inner torque/current control loop (see Figure 5) and therefore is sometimes referred to as a vector drive.

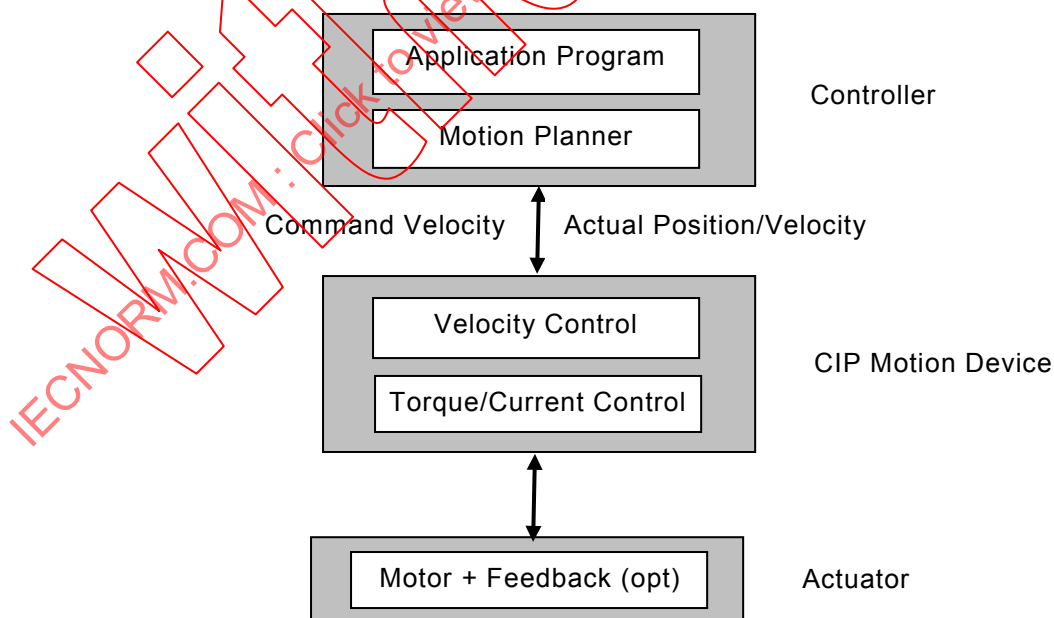


Figure 5 – Closed loop velocity control

A feedback device for the velocity loop drive configuration is optional. Tighter speed regulation is achieved when using a feedback device, particularly at low speed. When the feedback device is included, it may be used to return actual position, velocity, and

acceleration data to the controller via the cyclic data connection. When the feedback device is not included, only estimated velocity can typically be returned to the controller.

In addition to Command Velocity, the controller can also pass Command Acceleration for the purposes of forward control.

4.2.6 Acceleration Control

While neither a mainstream control mode in the industry nor mentioned in IEC 61800-7-1, the acceleration control mode is included here to complete the dynamic progression from velocity control to torque control and because the Motion Axis Object can support an Acceleration Command, potentially derived from the controller's motion planner. In the acceleration control mode, the application control program and motion planner provide acceleration set-point values to the CIP Motion device via the cyclic data connection. The drive converts the acceleration set-point into a torque command using the estimated system inertia. Acceleration control works in concert with the inner torque/current control loop as shown in Figure 6.

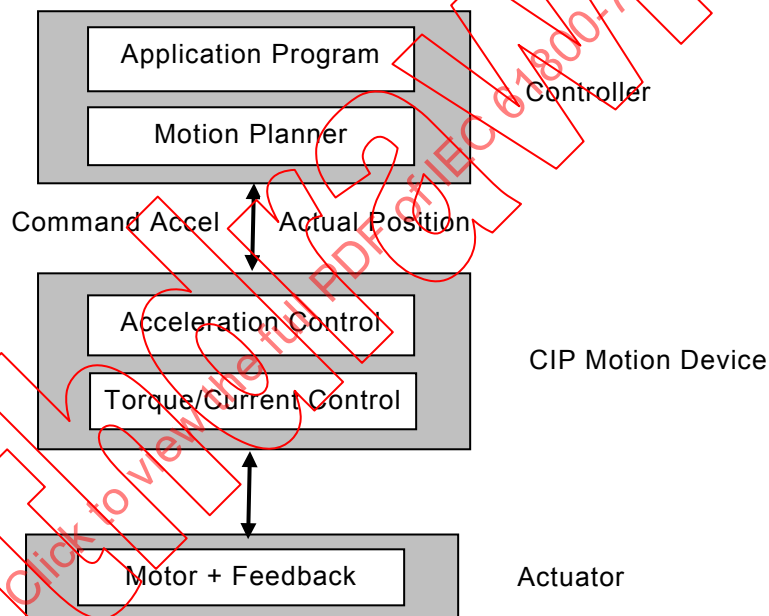


Figure 6 – Acceleration control

A feedback device for the acceleration control configuration is mandatory and may be used to return actual position, velocity, and acceleration data to the controller via the cyclic data connection.

4.2.7 Torque Control

In torque control application mode, the application control program or the motion planner provide torque set-point values to the device via the cyclic data connection (see Figure 7). Because motor current and motor torque are generally related by a torque constant, K_t , torque control is often synonymous with current control.

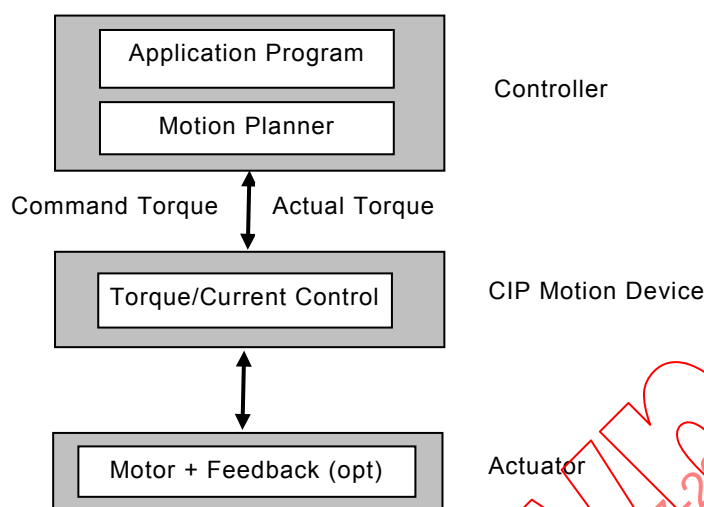


Figure 7 – Torque control

A position feedback device for this control mode is optional. If a feedback device is present, it may be used to return actual position, velocity, and acceleration data to the controller via the cyclic data connection.

4.2.8 No Control

The Motion Axis Object supports a “No Control” application mode. This configuration, also referred to as “feedback only” or “master feedback”, can be of value when a particular feedback channel in the CIP Motion drive device is to serve simply as a master feedback source to the rest of the control system. This application mode can also be applied to CIP Motion feedback devices. In this “No Control” configuration, no set-point value is supplied to the CIP Motion device via the cyclic data connection, but actual position, velocity, and acceleration can be supplied by the device to the controller via the cyclic data channel. The No Control drive configuration is illustrated in Figure 8.

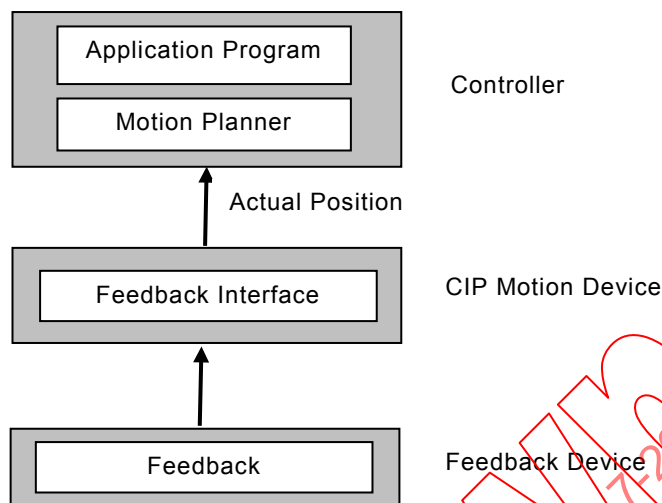


Figure 8 – No control

5 Data types

5.1 Data type overview

Table 1 shows references of data types used in this profile and the related definitions.

Table 1 – Data types

Data Types used in CIP Motion	Reference to definition
BOOL	Boolean (see 5.3.1.1.2 of IEC 61158-5-2)
SINT	Integer8 (see 5.3.1.4.2.2 of IEC 61158-5-2)
INT	Integer16 (see 5.3.1.4.2.4 of IEC 61158-5-2)
DINT	Integer32 (see 5.3.1.4.2.6 of IEC 61158-5-2)
LINT	Integer64 (see 5.3.1.4.2.8 of IEC 61158-5-2)
USINT	Unsigned8 (see 5.3.1.4.3.2 of IEC 61158-5-2)
UINT	Unsigned16 (see 5.3.1.4.3.4 of IEC 61158-5-2)
UDINT	Unsigned32 (see 5.3.1.4.3.6 of IEC 61158-5-2)
ULINT	Unsigned64 (see 5.3.1.4.3.8 of IEC 61158-5-2)
REAL	Float32 (see 5.3.1.4.1.2 of IEC 61158-5-2)
LREAL	Float64 (see 5.3.1.4.1.4 of IEC 61158-5-2)
SWORD/BYTE	Bitstring8 (see 5.3.1.2.2 of IEC 61158-5-2)
WORD	Bitstring16 (see 5.3.1.2.4 of IEC 61158-5-2)
DWORD	Bitstring32 (see 5.3.1.2.6 of IEC 61158-5-2)
LWORD	Bitstring64 (see 5.3.1.2.8 of IEC 61158-5-2)

5.2 Conventions

Hexadecimal numbers in this part of the IEC 61800-7 series may be represented by nn_{hex} (if their exact data type is not defined) or $0xnn$, $0xn\text{nnnn}$ (if their data type is specified).

6 CIP Drive Profile

6.1 Object model

6.1.1 Object overview

The object model in Figure 9 represents a CIP Motion drive device.

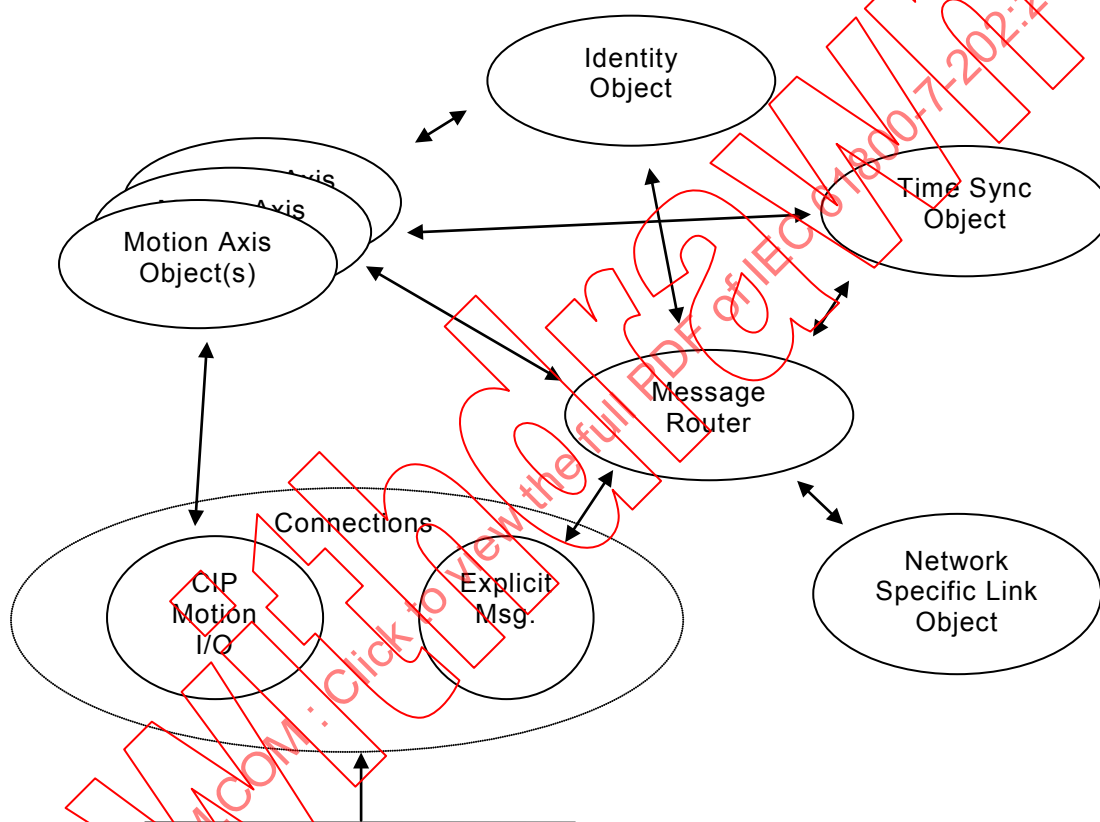


Figure 9 – Object Model for a CIP Motion drive

Table 2 indicates:

- the object classes present in this device,
- whether or not the class is required,
- the number of instances present in each class.

Table 2 – Objects present in a CIP Motion drive device

Object class	Optional or Required	Number of instances
CIP common required objects	Required	See IEC 61158-5-2 and IEC 61158-6-2
Motion Axis Object	Required	1 per axis
Time Sync Object	Optional	1

Refer to IEC 61158-5-2, IEC 61158-6-2 and 6.4.6.1 for more details about these objects.

6.1.2 Object description

The object model in Figure 9 shows the main functional components of the CIP Motion drive device profile defined as part of the CIP Motion extensions to CIP.

This object model also illustrates the use of multiple instances of a Motion Axis Object to implement a multi-axis drive device. Each Motion Axis Object instance governs the behavior of the associated drive axis. In this device profile, the term “drive axis” is synonymous with a “Motion Axis Object instance”.

A single bi-directional I/O connection to the Motion Axis Object class instance provides a cyclic data path between the controller and each individual Motion Axis Object instance. This connection passes on a special data structure whose self-defining format can be used to transfer cyclic, event, and service related data.

An optional Time Sync Object is included in the object model to facilitate accurate time synchronization between CIP Motion controllers and drive devices for high performance motion control. For lower performance drives such as V/Hz or VFD drives, the Time Sync Object is not necessary for drive operation.

6.2 How objects affect behaviour

The objects for this device affect the device’s behavior as shown in Table 3.

Table 3 – Object effect on behavior

Object class	Effect on behavior
CIP common required objects	See IEC 61158-5-2 and IEC 61158-6-2
Motion Axis Object	Provides dynamic control interface to drive, motor, and feedback components that comprise an axis.
Time Sync Object	Provide absolute time synchronization services between devices on the control network.

6.3 Defining object interfaces

Objects supported for the CIP Motion drive have the interfaces listed in Table 4.

Table 4 – Object interfaces

Object	Interface
CIP common required objects	See IEC 61158-5-2 and IEC 61158-6-2
Motion Axis Object	Message Router or Motion Axis Object class
Time Sync Object	Message Router or Motion Axis Object class

6.4 I/O Connection messages

6.4.1 General

The CIP Motion drive device profile supports a point-to-point bi-directional I/O connection between the controller and the Motion Axis Object class: this I/O connection is specifically referred to as the CIP Motion I/O connection.

The Motion Axis Object distributes the data in this connection to each instantiated Motion Axis Object instance. Therefore, the produced and consumed connection paths explicitly target Motion Axis Object class attributes (i.e. instance 0). The Consumed Connection path corresponds to the Controller-to-Device (C-to-D) connection, while the Produced Connection path corresponds to the Device-to-Controller (D-to-C) connection as detailed in the following subclauses.

The I/O Connection paths for this device are as follows:

Consumed Connection Path	=	Motion Axis Object class/Instance 0 / (Controller Consumed Connection Data attribute)
Produced Connection Path	=	Motion Axis Object class/Instance 0 / (Controller Produced Connection Data attribute)

6.4.2 CIP Motion I/O connection

6.4.2.1 Overview

The following subclause specifies the CIP Motion I/O Connection format that includes the Controller-to-Device (C-to-D) Connection and the Device-to-Controller (D-to-C) Connection for bi-directional data transfer between a CIP Motion Controller and a CIP Motion drive, as shown in Figure 10.

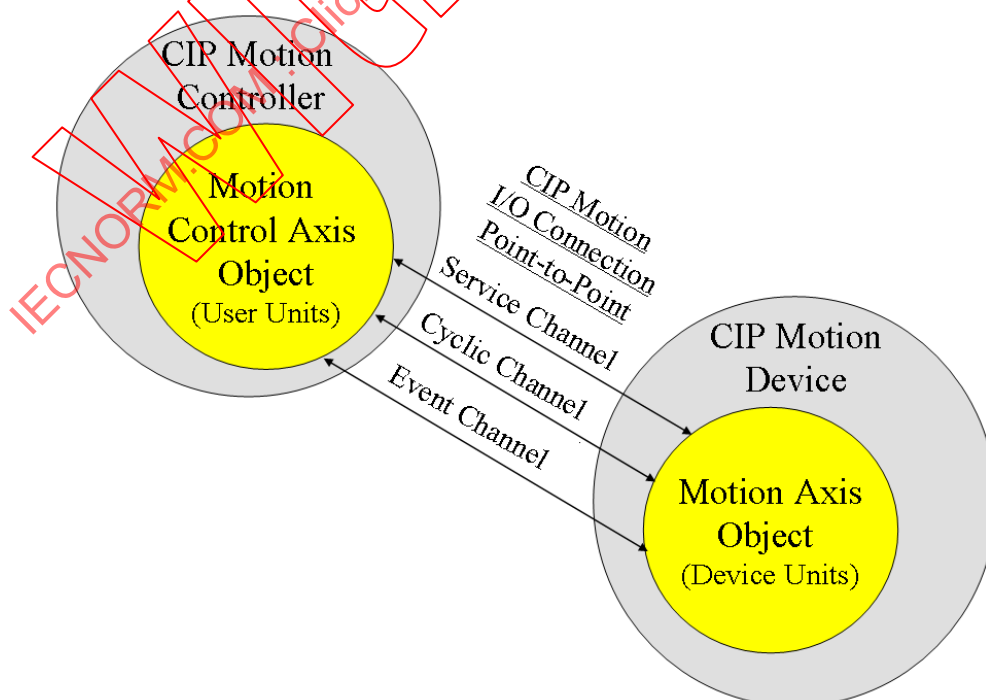


Figure 10 – CIP Motion I/O connection model

6.4.2.2 I/O connection formats

The Controller-to-Device connection format is shown in Figure 11 and the Device-to-Controller connection format is shown in Figure 12.

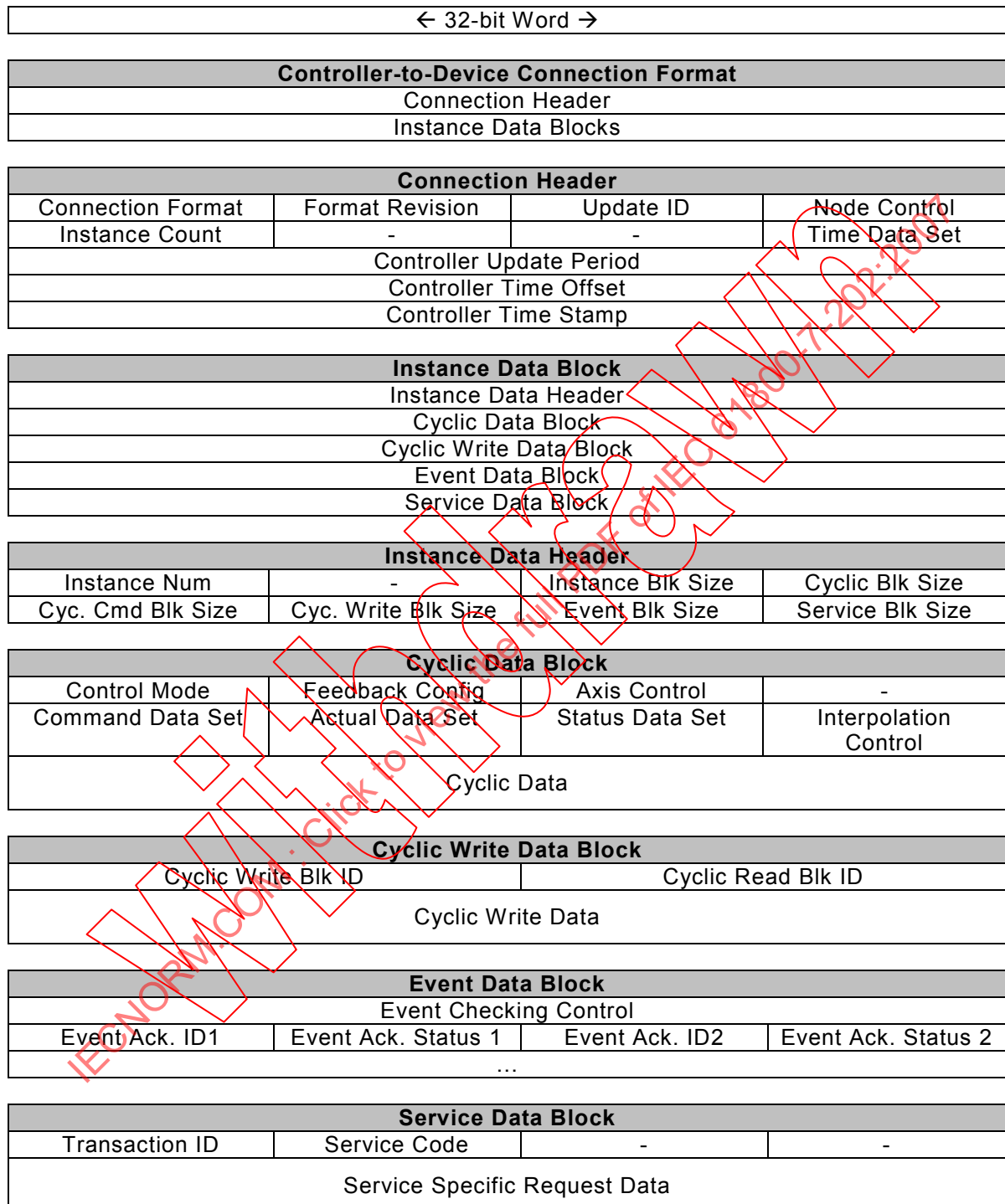


Figure 11 – Controller-to-Device Connection Format

← 32-bit Word →			
Device-to-Controller Connection Format			
Connection Header			
Instance Data Blocks			
Connection Header			
Connection Format	Format Revision	Update ID	Node Status
Instance Count	Node Alarms	Node Faults	Time Data Set
Device Update Period			
Device Time Offset			
Device Time Stamp			
Timing Diagnostic Data			
Instance Data Block			
Instance Data Header			
Cyclic Data Block			
Cyclic Read Data Block			
Event Data Block			
Service Data Block			
Instance Data Header			
Instance Num	-	Instance Blk Size	Cyclic Blk Size
Cyc. Act. Blk Size	Cyc. Read Blk Size	Event Blk Size	Service Blk Size
Cyclic Data Block			
Control Mode	Feedback Config	Axis Response	Response Status
-	Actual Data Set	Status Data Set	Axis State
Cyclic Data			
Cyclic Read Data Block			
Cyclic Read Blk ID	-		
Cyclic Read Data			
Event Data Block			
Event Checking Status			
Event ID	Event Status	Event Type	-
Event Position			
Event Time Stamp			
...			
Service Data Block			
Transaction ID	Service Code	General Status	Extended Status
Service Specific Response Data			

Figure 12 – Device-to-Controller Connection Format

6.4.2.3 CIP Motion connection structure

The CIP Motion I/O Connection actually consists of two unidirectional unicast connections; one passing data from the controller to the drive and the other passing data from the drive to the controller.

Both CIP Motion connection data structures (Controller-to-Device and Device-to-Controller) begin with a Connection Header that includes a 32-bit time stamp, followed by a series of data blocks for each axis instance supported by the drive node.

Each instance data block within the CIP Motion I/O connection packet consists of three sets of data blocks associated with the cyclic, event, and service data channels. The size of these data blocks for a given update is variable and determined by the connection and instance data block headers. From the drive's perspective, these three distinct data channels have different data processing priorities as illustrated in Figure 13.

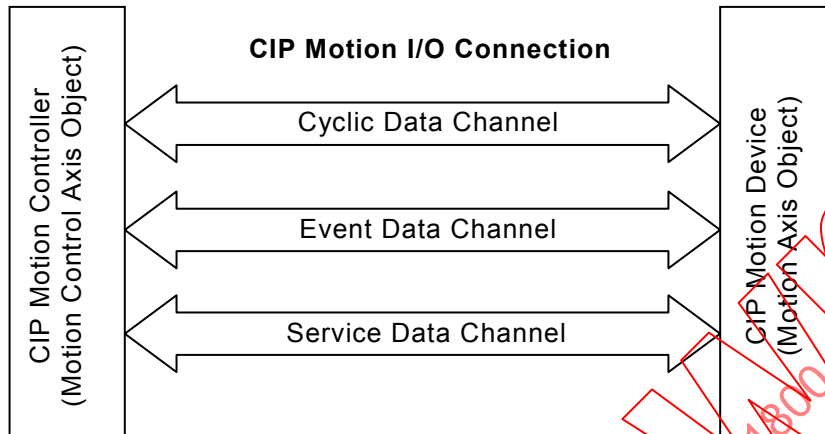


Figure 13 – CIP Motion I/O Connection Channels

The specific functionality of these three data channels is as follows:

- Cyclic Data Channel – carries cyclic data blocks that are sampled or calculated every Controller Update Period and synchronised with other nodes in the motion control system through use of distributed System Time. Cyclic data is high priority data that shall be immediately processed and applied to the drive axis within one Device Update Period.
- Event Data Channel - carries event data associated with drive event(s) (e.g. registration, homing, etc.) that have occurred within the last Controller Update Period. Event data is medium priority and shall be processed and applied within one Controller Update Period.
- Service Data Channel – carries data associated with service requests to read or write attribute values of the Motion Axis Object as part of on-line configuration and diagnostic functionality, as well as service requests to affect Motion Axis Object behaviour as part of controller instruction execution. Service data has lowest priority and is typically buffered and processed as a background task. There is no guarantee that a service request will be processed within Controller Update Period.

Taken together, these three data channels provide a comprehensive controller to drive data connection solution for industrial motion control.

6.4.3 Controller-to-Device connection

6.4.3.1 General

To facilitate a detailed description of each of its constituent data elements, the CIP Motion Controller-to-Device Connection is organised as shown in Figure 14.

CIP Motion Controller-to-Device Connection Format
Connection Header
Instance Data Blocks

Figure 14 – CIP Motion Controller-to-Device Connection Format

6.4.3.2 Controller-to-Device connection header

6.4.3.2.1 General structure

The Controller-to-Device Connection Header contains critical axis configuration information needed to parse the Instance Data Blocks. The fixed portion of the connection header is defined as shown in Figure 15.

Connection Header			
Connection Format	Format Revision	Update ID	Node Control

Figure 15 – Connection Header

- **Connection Format:** This enumerated byte determines what the format of CIP Motion connection is according to the definition in Figure 16.

Bit 7	Connection Format		Bit 0
	(Reserved)	Connection Type	

Figure 16 – Connection Format

- **Connection Type:** This 4-bit enumeration defines the CIP Motion connection type as shown below. Valid values for a Controller-to-Device Connection are 2 and 6. Fixed connections have a fixed connection size during operation to support lower-performance CIP networks like ControlNet and DeviceNet and typically associated with a simple single axis drive device that does not support time synchronisation services. Variable connections allow the connection data structure to vary in size during operation and are targeted for high-performance drives and CIP networks like EtherNet/IP.

- 0 = Reserved
- 1 = Reserved
- 2 = Fixed Controller-to-Device Connection
- 3 = Fixed Device-to-Controller Connection
- 4 = Reserved
- 5 = Reserved
- 6 = Variable Controller-to-Device Connection
- 7 = Variable Device-to-Controller Connection
- 8 to 15 = Reserved.

- **Format Revision:** This edition of IEC 61800-7-202 defines Format Revision 1. Devices utilizing this edition of IEC 61800-7-202 will only recognize Format Revision 1. This value is incremented by 1 for every revision of the Controller-to-Device Connection format that impacts the interface. The Format Revision allows newer drives to support the connection formats generated by an older controller. It also allows older drives to recognise a newer connection format from a controller that it cannot support and to generate a Node Fault Error Code 4 (Data Format Error).
- **Update ID:** This cyclic transaction number is incremented every update period. The Update ID is like the CIP sequence count and is used by the drive to determine whether the connection buffer contains fresh data. If the Update ID has not changed, the drive attempts to ride through the missed update by extrapolation based on previous trajectory until fresh data arrives. In the case where the drive axis is not synchronised, or does not support time synchronisation services, the time stamp data is either not included or invalid, so the Update ID is the only way for the drive to detect new connection data. In this case, the Controller-to-Device Connection Update ID is applied to the next Device-to-Controller Connection Update ID.
- **Node Control:** This element is applied to the Motion Axis Object class attribute “Node Control”, which is used to control the state of the associated drive communications node. See 7.2.2.3 (Motion Axis Object) for details.

6.4.3.2.2 Fixed connection header

If the Connection Format is a Fixed Controller-to-Device Connection, the above header is immediately followed by the instance data block.

6.4.3.2.3 Variable connection header

If the Connection Format is a Variable Controller-to-Device Connection, then the connection header contains additional fields related to multi-axis drive addressing and time stamping, as shown in Figure 17.

Connection Header			
Connection Format	Format Revision	Update ID	Node Control
Instance Count	-	-	Time Data Set
Controller Update Period			
Controller Time Offset			
Controller Time Stamp			

Figure 17 – Connection Header

- Instance Count: This value reflects the number of instance data blocks present in the CIP Motion Controller-to-Device Connection data structure.
- Time Data Set: This bit-mapped byte contains flags that determine the usage and format of the controller and drive timing information, as specified in Table 5. The Time Data Set value sent by the controller in the Controller-to-Device Connection becomes the Time Data Set value sent by the drive in the Device-to-Controller Connection. The value of this element is transferred to the Motion Axis Object attribute “Command Data Set”.

Table 5 – Time Data Set

Bit	Definition	Syntax
0	Update Period	0 = no update period 1 = update period included
1	Time Stamp	0 = no time stamp and offset 1 = time stamp and offset included
2	Time Diagnostic Data	0 = no diagnostic data 1 = diagnostic data included
3 to 7	(Reserved)	

- Controller Update Period: This 32-bit element specifies the update period of the Controller-to-Device Connection in nanoseconds. The value of this element is transferred to the Motion Axis Object class attribute of the same name. When the connection is synchronised, the Controller Update Period can be used by the drive along with the Controller Time Stamp to check for missed or late updates. If the difference between the last Controller Time Stamp and the current local drive time stamp exceeds the maximum value given by,

Max Delay = Controller Update Delay High Limit × Controller Update Period,

the Controller Update Fault bit is set in the Node Faults class attribute of the Motion Axis Object. An optional Controller Update Delay Low Limit attribute is also available. If the difference between the last Controller Time Stamp and the current local drive time stamp exceeds the maximum value given by,

Max Delay = Controller Update Delay Low Limit × Controller Update Period,

the Controller Update Alarm bit is set in the Node Alarm class attribute of the Motion Axis Object.

When the connection is synchronised and fine interpolation is enabled, the Controller Update Period is also used by the drive along with the Controller Time Stamp value to compute new coefficients to the interpolation/extrapolation polynomial.

In the case where the connection is not synchronised, no late update checks are performed by the drive. In this case, the command data should be applied immediately, which means applying the data immediately as a command reference.

- **Controller Time Offset:** This element represents the 64-bit System Time Offset value associated with the Controller Time Stamp that follows. The value of this element is transferred to the Motion Axis Object class attribute of the same name. The Controller Time Offset value is used by the drive to determine if System Time as defined in the controller is skewed relative to System Time in the drive. Normally, System Time between the drive and the controller are closely matched at any given time. But every few seconds, according to the IEC 61588:2004 protocol, the time master of the system can correct the System Time reference by a significant amount of time, perhaps an hour. Indeed, even the master itself can change as a higher quality time master is discovered by the system. These step changes to the System Time reference propagate through to the various devices on the network in such a way that the controller and the drive are running with skewed System Time values. The Controller Time Offset value can be used together with the System Time Offset for the drive to both determine if System Time between the drive and the controller is skewed and if so, to compensate for the skew.
- **Controller Time Stamp:** This element represents the 64-bit System Time value at the beginning of the Controller Update Period when the controller's update timer event occurred. It is calculated by the controller as the sum of the controller's local clock value when update timer event occurred and the controller's System Time Offset value. The Controller Time Stamp is therefore directly associated with the command data contained in the connection. The value of this element is transferred to the Motion Axis Object class attribute of the same name. Taken together with the Controller Update Period and the Command Target Time (discussed later), the drive has all the information it needs to compute command interpolation/extrapolation polynomials to correct command data values for differences between the drive and controller update timing. These differences can occur when the Controller Update Period is not an integer multiple of the Device Update Period or when the drive updates are phase shifted relative to the controller.

6.4.3.3 Instance data blocks

6.4.3.3.1 General structure

After the Connection Header are one or more Instance Data Blocks as determined by the above Instance Count. The Instance Data Block has the basic structure shown in Figure 18.

Instance Data Block
Instance Data Header
Cyclic Data Block
Event Data Block
Service Data Block

Figure 18 – Instance Data Block

6.4.3.3.2 Instance data header

The Instance Data Header contains critical axis configuration information needed to parse and apply the data contained in the three data channels (see Figure 19). This header is only included in the Variable Connection format to accommodate multi-axis drive applications. Information within the header can be used by the drive communications interface to copy the individual data blocks into separate fixed memory locations for processing.

If configured for a Fixed Connection format, only the Cyclic Data Block for a single axis instance is supported so there is no need for any information to specify instance number or block sizing. The Instance Data Header is therefore not included.

Instance Data Header			
Instance Num	(Reserved)	Instance Blk Size	Cyclic Blk Size
Cyc. Act. Blk Size	Cyc. Write Blk Size	Event Blk Size	Service Blk Size

Figure 19 – Instance Data Header

- **Instance Number:** This is the number that identifies the specific Motion Axis Object instance the following instance data block applies to. Motion Axis Object instances are created as a contiguous series of instance numbers starting with instance 1. Within the connection data structure, the Instance Numbers for each consecutive instance data block shall be an ordinal sequence, i.e. 1, 2, 3, etc. (instance 0 is defined as the class instance and not generally used as part of the connection data). Thus, in theory, up to 255 instances can be serviced by the CIP Motion I/O Connection.
- **Instance Block Size:** This value represents the size of the Instance Data Block in 32-bit words including the header. The Instance Block Size is useful when the drive wants to directly access the next instance data block without having to add the sizes of the cyclic, event, and service blocks.
- **Cyclic Block Size:** This value represents the size to the Cyclic Data Block in 32-bit word units including the header.
- **Cyclic Command Block Size:** This value represents the size to the Cyclic Command Data Block in 32-bit word units including the header.
- **Cyclic Write Block Size:** This value represents the size to the Cyclic Write Data Block in 32-bit word units including the header. If the Cyclic Write Block Size is 0, the drive does not need to support the Cyclic Read/Write functionality.
- **Event Block Size:** This value represents the size to the Event Data Block in 32-bit word units including the header. If the Event Block Size is 0, the drive does not need to support the event functionality.
- **Service Block Size:** This value represents the size to the Service Data Block in 32-bit word units including the header. A Service Block Size value of 0 indicates there is no service request to process.

6.4.3.3.3 Cyclic data block

The Cyclic Data Header at the top of the Cyclic Data Block is always included regardless of the connection format. This header contains key elements related to the content of the Cyclic Data Block of both the Controller-to-Device Connection and Device-to-Controller Connection, and the context of the data as determined by the Control Mode and Feedback Configuration (see Figure 20). The header also provides a mechanism to control the state of the targeted drive axis.

Cyclic Data Header			
Control Mode	Feedback Config	Axis Control	-
Command Data Set	Actual Data Set	Status Data Set	Interpolation Control
Cyclic Data			

Figure 20 – Cyclic Data Header

- **Control Mode:** This 8-bit enumerated value determines the control mode context of the command data as presently configured in the motion controller. This value can be changed while on-line and even while the drive axis is in the Running state. If a particular Control Mode transition is not supported by the drive, an Illegal Mode Change exception is generated that can be configured to perform any one of a number of actions in response to the illegal transition.

The value of this element is transferred to the Motion Axis Object attribute of the same name. The complete Control Mode attribute definition can be found in the attribute tables of the Motion Axis Object (see 7.3.2.2.1).

- **Feedback Configuration:** This 8-bit enumerated value indicates the feedback context of the command data as presently configured in the motion controller. Command data can be referenced to feedback counts of either Feedback 1 or Feedback 2, or motor units for sensorless operation. This value can be changed while on-line and even while the drive axis is in the Running state. If a particular Feedback Configuration transition is not supported by the drive, an Illegal Mode Change exception is generated that can be configured to perform any one of a number of actions in response to the illegal transition.

The value of this element is transferred to the Motion Axis Object attribute of the same name. The complete Feedback Configuration attribute definition can be found in the attribute tables of the Motion Axis Object (see 7.3.4.2).

- **Axis Control:** This 8-bit enumerated code can be used to directly initiate drive state change operations that do not require either passing or returning drive parameters, and therefore, do not require a CIP service to initiate. Valid enumerations for this data element are shown in Table 6.

Table 6 – Axis Control

Request Code	Requested Operation
0	No Request
1	Enable Request
2	Disable Request
3	Shutdown Request
4	Shutdown Reset Request
5	Abort Request
6	Fault Reset Request
7 to 127	(Reserved)
128 to 255	(Vendor Specific)

- **Command Data Set:** This bit mapped value has a bit defined for each possible real-time command reference (see Table 7). Command data appears in the same order in the Command Data Set as the bit numbers, so Command Position would appear before Command Torque in the real-time data structure of the Controller-to-Device Connection.

The value of this element is transferred to the Motion Axis Object attribute “Command Data Set”.

Table 7 – Command Data Set

Bit	Controller Command Data Element	Data Type
0	Command Position	DINT
1	Command Velocity	REAL
2	Command Acceleration	REAL
3	Command Torque	REAL
4	Position Trim	DINT
5	Velocity Trim	REAL
6	Acceleration Trim	REAL
7	Torque Trim	REAL

The above Controller Command Data Elements apply to the Controller-to-Device Connection's cyclic data structure and map to corresponding attributes in the Motion Axis Object as shown in Table 8.

The units of the Command Data Elements match the units defined for the associated Motion Axis Object attribute.

Table 8 – Command Data Element to Motion Axis Object Attribute mapping

Bit	Command Data Element	Motion Axis Object Attribute
0	Command Position	Controller Position Command
1	Command Velocity	Controller Velocity Command
2	Command Acceleration	Controller Acceleration Command
3	Command Torque	Controller Torque Command
4	Position Trim	Position Trim
5	Velocity Trim	Velocity Trim
6	Acceleration Trim	Acceleration Trim
7	Torque Trim	Torque Trim

It is the job of the controller to ensure that the necessary Command Data Elements are included in the connection data to support the specified Control Mode.

- **Actual Data Set:** This bit mapped value has a bit defined for each possible real-time actual data attribute that is to be included in the Actual Data Set of the Device-to-Controller Connection's Instance Data Block in the next update (see Table 9). Actual data appears in the same order as the bit numbers, so Actual Position would appear before Actual Torque in the Actual Data Set structure. Using this mechanism, the contents of the Actual Data Set may be changed at any time during drive operation. The value of this element is transferred to the Motion Axis Object attribute "Feedback Data Set".

Table 9 – Actual Data Set

Bit	Actual Data Element Produced	Data Type
0	Actual Position	DINT
1	Actual Velocity	REAL
2	Actual Acceleration	REAL
3	Actual Torque	REAL
4	Actual Current	REAL
5	Actual Voltage	REAL
6	Actual Frequency	REAL
7	(Reserved)	

The above Actual Data Elements map to corresponding attributes in the Motion Axis Object as shown in Table 10. The units of the Actual Data Elements match the units defined for the associated Motion Axis Object attribute.

Table 10 – Actual Data Element to Motion Axis Object Attribute Mapping

Bit	Actual Data Element	Motion Axis Object Attribute
0	Actual Position	Position Feedback
1	Actual Velocity	Velocity Feedback
2	Actual Acceleration	Acceleration Feedback
3	Actual Torque	Torque Reference
4	Actual Current	Iq Current Reference
5	Actual Voltage	Output Voltage
6	Actual Frequency	Output Frequency
7	(Reserved)	

- **Status Data Set:** This bit-mapped byte contains flags that determine the contents of the Status Data Set of the Device-to-Controller Connection's Instance Data Block in the next update (see Table 11). Status data appears in the same order as the bit numbers, so Axis Fault Status would appear before, for example, Axis Alarm Status data in the Status Data Set structure. Multiple attributes comprising a selected Status Data Element are transmitted in the order listed from top to bottom, so Axis Fault Code is transmitted before Mfg Axis Fault Code. The definitions of each of these Status Data Elements can be found by looking up the corresponding Motion Axis Object attribute (see 7.3). The value of this element is transferred to the Motion Axis Object attribute "Status Data Set".

Table 11 – Status Data Set configuration

Bit	Status Data Element Produced	Data Type
0	Initialization Fault Code Initialization Fault Code – Mfg Axis Fault Code Axis Fault Code – Mfg Axis Alarm Code Axis Alarm Code – Mfg Start Inhibit Code Start Inhibit Code – Mfg Axis Fault Time Stamp	USINT USINT USINT USINT USINT USINT USINT USINT LINT
1	Axis Fault Status Axis Fault Status - Mfg	LWORD LWORD
2	Axis Alarm Status Axis Alarm Status - Mfg	LWORD LWORD
3	Start Inhibit Status Start Inhibit Status - Mfg	WORD WORD
4	Axis Status Axis Status - Mfg Axis I/O Status Axis I/O Status - Mfg	DWORD DWORD DWORD DWORD
5	(Reserved)	
6	(Reserved)	
7	(Vendor specific)	

- **Interpolation Control:** The byte contains information needed to control the fine interpolation algorithm and determine the target time of the command data to the drive control structure (see Table 12). The value of this element is transferred to the Motion Axis Object attribute of the same name.

Table 12 – Interpolation Control

Bit	Definition	Syntax
0 to 1	Command Target Time	0 = Immediate 1 = Extrapolate (+1 Update Period) 2 = Interpolate (+2 Update Periods)
3 to 5	(reserved)	
6 to 7	(vendor specific)	

Command Target Time – This 2-bit integer defines a specific time relative to the Connection Time Stamp that the Command Data is targeted for, as defined by the controller's motion planner. The absolute command target time is the sum of the Controller Time Stamp from the controller and the product, Command Target Time x Controller Update Period.

- A Command Target Time of 0 implies that the Command Data is targeted for the beginning of the current update cycle and, thus, needs to be applied to the control structure Immediately. In this case, there is no need for any fine interpolation. This situation can occur when the Controller Update Period is significantly shorter than the Device Update Period, or when the controlled motion of the axis is nearly constant during the span of the Controller Update Period.
- A Command Target Time of 1 implies that the target for the Command Data is the next Connection Update timer event. In this case, the command interpolator functions primarily as an extrapolator that estimates the next Command Data value based on the present trajectory of the axis. This is a typical setting when the Controller Update Period is comparable to the Device Update Period or when the controlled motion of the axis is relatively constant during the span of the Controller Update Period.
- A Command Target Time of 2 implies that the target for the Command Data is two Connection Update timer events from the Connection Time Stamp. In this case, the command interpolator can compute a smooth trajectory based on the current dynamics of the motor to reach the Command Data value at the targeted time. This is true fine interpolation and is applicable when the Controller Update Period is significantly larger than the Device Update Period.
- **Cyclic Command Data:** The Cyclic Command Data contains high priority data that needs to be applied to the associated drive axis during the next drive update. This block consists of command data elements that are applied as references to the drive's control algorithms and explicitly determined by the Command Data Set element in the Cyclic Command Data Header.

6.4.3.3.4 Cyclic Write data block

The Cyclic Write Data Block can be used to synchronously update one or more targeted Motion Axis Object configuration parameters within the drive. This mechanism can be used in conjunction with a Function Block program to implement sophisticated outer loop control, gain scheduling, and dynamic limiting algorithms. Unlike service channel Set Axis Attribute service requests, which may take several drive update cycles to process, the Cyclic Write Data mechanism guarantees that the targeted parameter is applied at the next available drive update.

The Cyclic Write Data Block is only supported in the Variable Connection format (see Figure 21).

Cyclic Write Data Block	
Cyclic Write Blk ID	Cyclic Read Blk ID
Cyclic Write Data	

Figure 21 – Cyclic Write Data Block

The associated header for this block contains key elements related to the content of the Cyclic Write Data Block.

- **Cyclic Write Block ID:** This 16-bit ID determines the pre-defined Cyclic Write Block structure to apply to the Cyclic Write Data for this update. Cyclic Write Block structures are defined using the Set Cyclic Write Data List service. A successful response to this service includes a new Cyclic Write Blk ID that can be used in the next connection update to pass cyclic data in this format. The previous Cyclic Write Blk ID and its associated structure shall be maintained until the controller requests transmission of the new Cyclic Write Blk ID at which point the old Cyclic Write Blk ID and structure definition are no longer required.
- **Cyclic Read Block ID:** This 16-bit ID determines the pre-defined Cyclic Read Block structure to apply to the Cyclic Read Data for next Device-to-Controller Connection update. Cyclic Read Block structures are defined using the Set Cyclic Read Data Block service. A successful response to this service includes a new Cyclic Read Blk ID that can be used in the next connection update to allow the drive to use the new Cyclic Read Data format for next available Device-to-Controller Connection update.
- **Cyclic Write Data:** The Cyclic Write Data contains high priority data that needs to be applied to the associated drive axis during the next drive update. This block consists of parametric data elements that are applied to Motion Axis Object attributes that are used by the control algorithms. The contents of the Cyclic Write Data are explicitly determined by the structure identified by the Cyclic Write Blk ID found in the Cyclic Write Data Header.

6.4.3.4 Event data block

The Event Data Block is used to convey information regarding the event channel. In particular, the Event Data Block for the Controller-to-Device Connection is used to control the arming of event checking functions in the drive as well as acknowledge receipt of event notifications from the drive that are sent via the Device-to-Controller Connection's Event Data Block.

The Event Data Block for the Controller-to-Device Connection has the format shown in Figure 22.

Event Data Block			
Event Checking Control			
Event Ack. ID1	Event Ack. Status 1	Event Ack. ID2	Event Ack. Status 2
...			

Figure 22 – Event Data Block

- **Event Checking Control:** This 32-bit word is copied into the Motion Axis Object attribute of the same name that is used to enable various drive inputs, for example marker and registration inputs, to generate events. When these events occur, the drive captures both the time and exact axis position when the event occurred. The last 4 bits of the Event Checking Control is a binary value that specifies the number of active events, which is literally the number of Event Acknowledge IDs listed in this Event Data Block. A complete definition for the Event Checking Control attribute can be found in 7.3.5.2.1.

The Event Control mechanism works as follows:

- 1) Controller sets the appropriate Event Checking bit to look for a specific event condition to occur. Multiple bits in the Event Checking Control word may be set at any given time.

- 2) Drive detects that the Event Checking Control request bit is set and initiates the requested event checking action.
- 3) Drive sets the corresponding Event Checking Status bit to acknowledge that the drive is now actively checking for the specified event condition.
- 4) Controller sees that the Event Checking Status bit from the drive is set indicating that specified event trigger is “armed” at the drive. Starting with this update, the controller can process any event notifications from the drive that match the specified event condition.
- 5) Drive detects the specified event condition, increments the Event ID, and sends an Event Notification Data Block with this ID to the controller. If the associated Auto-rearm bit is clear in the Event Checking Control word, the drive discontinues checking for the specified event. However, the Event Checking Status bit remains set until the Event Checking Request bit is cleared by the controller. Notifications for other events can be sent in the same update.
- 6) Controller processes the Event Notification, sends the Event Acknowledge to the drive and, if Auto-rearm is not enabled, clears the associated Event Checking bit for the next Controller-to-Device Connection update.
- 7) Drive detects that the Event Checking Request bit is clear and clears the corresponding Event Checking Status bit prior to the next Device-to-Controller Connection update.
- 8) Controller is able to set the Event Checking bit again as soon as it sees the associated Event Checking Status bit has been reset. Therefore, the minimum time between when the Event Notification was received by the controller to the time when the drive is rearmed for the next event is one Controller Update Period.

In the case of a Registration event where Auto-rearm Event Checking is requested, the event handling sequence would be as follows:

- 1) Controller sets the appropriate Event Checking bit to look for a specific event condition to occur and also sets the Auto-rearm bit for that event condition.
- 2) Drive detects that the Event Checking Control request bit is set and initiates the requested event checking action.
- 3) Drive sets the corresponding Event Checking Status and Auto-rearm bits to acknowledge that the drive is now actively checking for the specified event condition.
- 4) Controller sees that the Event Checking Status bit from the drive is set indicating that specified event trigger is “armed” at the drive. Starting with this update, the controller can process any event notifications from the drive that match the specified event condition.
- 5) Drive detects the specified event condition, increments the Event ID, and sends an Event Notification Data Block with this ID to the controller. Since the corresponding Auto-rearm bit is set, the drive continues checking for the specified event. The Event Checking Status bit remains set until the Event Checking Request bit is cleared by the controller. If the Drive detects another specified event condition prior to transmission of the first event to the controller, it increments the Event ID again, and appends another Event Notification with this ID to the Event Data Block to the controller. Since the corresponding Auto-rearm bit is set, the drive continues checking for the specified event.
- 6) Controller processes the Event Notification, sends the Event Acknowledge to the drive but, since the Auto-rearm bit is enabled, it leaves the associated Event Checking bit set for the next Controller-to-Device Connection update.
- 7) Drive receives the Event Acknowledge from the controller. It can now send another Event Notification if the specified event has occurred again. Otherwise, go to the next step.
- 8) Drive continues checking for the specified event condition and sends an Event Notification Data Block to the controller whenever the event condition occurs. The

Event Checking Status bit remains set until the Event Checking Request bit is cleared by the controller. With the Auto-rearm feature, event checking is continuously enabled, ensuring that no registration events are missed during the normal one cycle delay in re-arming the event checking mechanism. The down side of Auto-rearm feature is that it can generate a multitude of uninteresting events to process, in fact, multiple specified events per Controller Update Period. Therefore, these events shall either be buffered in the controller in an event array attribute or filtered based on the event data. The later case is how Windowed Registration functionality is implemented; the Windowed Registration feature checks each registration event that arrives from the drive to see if the Event Position yields an absolute position value that is within the configured Registration Window. If the computed registration position is outside the window, the event is discarded. If the computed registration position is within the window, the registration position and registration time stamp is stored in the controller.

- **Event Acknowledge ID:** Transaction number assigned to this event by the original event notification. Each event is assigned a new Event ID by incrementing the current Event ID stored in the drive. Using the Event ID, the drive is able to match the event acknowledgement to the appropriate event notification to complete the event data transaction.
- **Event Acknowledge Status:** Enumerated value indicating controller response to the event. A value of 0 indicates that the event was successfully processed. A non-zero value indicates that an error occurred in the event processing and the event shall be resent.

6.4.3.5 Service data block

The service data block allows one service request per instance to be sent to the drive in a given update. The service request requires a specific service response from the drive indicating success or an error. In some cases, the response service contains requested data. In any case, the service request data persists in the Controller-to-Device Connection data structure until the controller receives the associated service response from the drive.

Each service request is represented by a block of data organized as shown in Figure 23.

NOTE By design, the first 4 bytes of the service data block do not follow the traditional CIP standard messaging format. That is primarily because this connection structure is, fundamentally, a CIP Implicit I/O connection, not an Explicit Messaging connection. However, in the case of a Fixed Connection format, the Service Specific Request Data defined below is sent via an Explicit Messaging connection and follows the CIP rules for explicit service request format.

Service Data Block			
Transaction ID	Service Code	-	-
Service Specific Request Data			

Figure 23 – Service Data Block

- **Transaction ID:** Transaction number assigned to this service request by the controller. Each service request is assigned a new Transaction ID by incrementing the current Transaction ID stored in the controller. Using the Transaction ID, the controller is able to match the service response to the appropriate service request and complete the service transaction.
- **Service Code:** Identifier that determines the object specific service request that follows. The list of supported Service Codes can be found in the Object Specific Services subclause of this part of the IEC 61800-7 series (see 7.5.1). CIP Common services are not applicable to the Service Data Block.
- **Service Specific Request Data:** The format and syntax of the Service Specific Request Data depends on the specified Service Code. This is true regardless of whether the service specific request data is passed in the Controller-to-Device connection or as part of an Explicit messaging connection.

6.4.4 Device-to-Controller connection

6.4.4.1 General

Like the Controller-to-Device Connection data structure described above, the CIP Motion Controller-to-Device Connection is organised as shown in Figure 24.

CIP Motion Controller-to-Device Connection Format	
Connection Header	
Instance Data Blocks	

Figure 24 – CIP Motion Controller-to-Device Connection Format

6.4.4.2 Device-to-Controller connection header

6.4.4.2.1 General structure

The Device-to-Controller Connection Header contains critical axis configuration information needed to parse the Device-to-Controller Connection data block. The fixed portion of the connection header is defined as shown in Figure 25.

Connection Header			
Connection Format	Format Revision	Update ID	Node Status

Figure 25 – Connection Header

- **Connection Format:** Same as Controller-to-Device definition except the required value for the Connection Type is either 3, indicating a Fixed Device-to-Controller Connection type or 7, indicating a Fixed Device-to-Controller Connection type.
 - 0 = Reserved
 - 1 = Reserved
 - 2 = Fixed Controller-to-Device Connection
 - 3 = Fixed Device-to-Controller Connection
 - 4 = Reserved
 - 5 = Reserved
 - 6 = Variable Controller-to-Device Connection
 - 7 = Variable Device-to-Controller Connection
 - 8 to 15 = Reserved.
- **Format Revision:** The Format Revision number is 1 for any device that utilizes this edition of IEC 61800-7-202. This value is to be incremented by 1 for every revision of the Device-to-Controller Connection format that impacts the interface. The Format Revision allows newer controllers to support the connection formats generated by older drives. It also allows older controllers to recognize a newer connection format from a drive that it cannot support and generate an appropriate error to its application.
- **Update ID:** The Device-to-Controller Connection Update ID should match the Update ID of the preceding Controller-to-Device Update ID and therefore shall be incremented every update period. In the case where the associated Controller-to-Device packet is lost or late, the Device-to-Controller Update ID shall be incremented as if the Controller-to-Device packet had arrived on time. This allows the CIP Motion control system to ride through a lost or missed Controller-to-Device packet and maintain synchronisation with matching Update IDs.

The Update ID is like the CIP message sequence count and is used by the controller to determine whether the connection buffer contains fresh data. If the Update ID has not changed, the controller attempts to ride through the missed update by extrapolation based on the previous trajectory until a fresh update arrives. In the case where the axis is not synchronised, or does not support time synchronisation services, the time stamp data is either not included or invalid, so no ride through extrapolation is attempted. In that case, the Update ID is also the only way for the controller to detect new connection data.

- **Node Status:** Contains bits used to indicate the status of the associated drive device communications node. The value of this element is derived from the Motion Axis Object class attribute of the same name. See 7.2.2.4 (Motion Axis Object) for details.

6.4.4.2.2 Fixed connection header

If the Connection Format is a Fixed Device-to-Controller Connection, the above header is immediately followed by the instance data block.

6.4.4.2.3 Variable connection header

If the Connection Format is a Variable Device-to-Controller Connection, then the connection header contains additional fields related to multi-axis drive addressing and time stamping (see Figure 26).

Connection Header			
<i>Connection Format</i>	<i>Format Revision</i>	<i>Update ID</i>	<i>Node Control</i>
Instance Count	Node Alarms	Node Faults	Time Data Set
Device Update Period			
Device Time Offset			
Device Time Stamp			
Data Received Time Stamp			
Data Transmit Time Stamp			
Sync Variance			

Figure 26 – Connection Header

- **Instance Count:** Same as Controller-to-Device definition.
- **Node Alarms:** Contains bits used to indicate the alarm conditions of the associated drive communications node. The value of this element is derived from the Motion Axis Object class attribute of the same name. See 7.2.2.6 (Motion Axis Object) for details.
- **Node Faults:** Contains bits used to indicate the fault conditions of the associated drive communications node. The value of this element is derived from the Motion Axis Object class attribute of the same name. See 7.2.2.5 (Motion Axis Object) for details.
- **Time Data Set:** Same as Controller-to-Device definition.
- **Device Update Period:** This element represents the current period between drive control calculations that apply the command data. This value is used by the controller to configure the behaviour of the fine interpolation blocks that are applied to the command data. If the Controller Update Period is much longer than the Device Update Period, fine interpolation is generally applied. When the Controller Update Period is comparable to the Device Update Period, fine interpolation is unnecessary and only extrapolation is required to perform time stamp corrections.
- **Device Time Offset:** This element represents the 64-bit System Time Offset value associated with the Device Time Stamp that follows. The Device Time Offset value is used by the controller to determine if System Time as defined in the drive is skewed relative to System Time in the controller. Normally, System Time between the drive and the controller are closely matched at any given time. But every few seconds, according to the IEC 61588:2004 protocol, the time master of the system can correct the System Time reference, sometimes by a significant amount of time, perhaps as much as an hour. Indeed, even the master itself can change as a higher quality time master is discovered by the system. These step changes to the System Time reference propagate through to the various devices on the network in such a way that the controller and the drive are running with skewed System Time values. The Device Time Offset value can be used together with the System Time Offset for the controller to both determine if System Time between the drive and the controller is skewed and if so, to compensate for the skew.
- **Device Time Stamp:** This time stamp value represents the 64-bit System Time value, in nanoseconds, when the drive's update timer event that is associated with the actual data in the connection structure occurred, for example when the actual data was captured. It is

calculated by the drive as the sum of the drive's local clock value when update timer event occurred and the drive's System Time Offset value. With the Device Time Stamp, the controller has all the information it needs to correct actual data values for differences between the drive and controller update timing that result when the Controller Update Period is not an integer multiple of the Device Update Period or when the drive updates are phase shifted relative to the controller. It is assumed in this timing model that the Device Time Stamp is registered to the beginning of the drive update period and is also the time when feedback was last captured. In the case where the Device Time Stamp does not match the local update time stamp of the controller, the controller extrapolates the actual response data value based on trajectory to correspond to the controller's time stamp. The timing diagram in Figure 27 illustrates how axis position data from the drive is adjusted by the controller based on the relative time stamps between the drive and the controller.

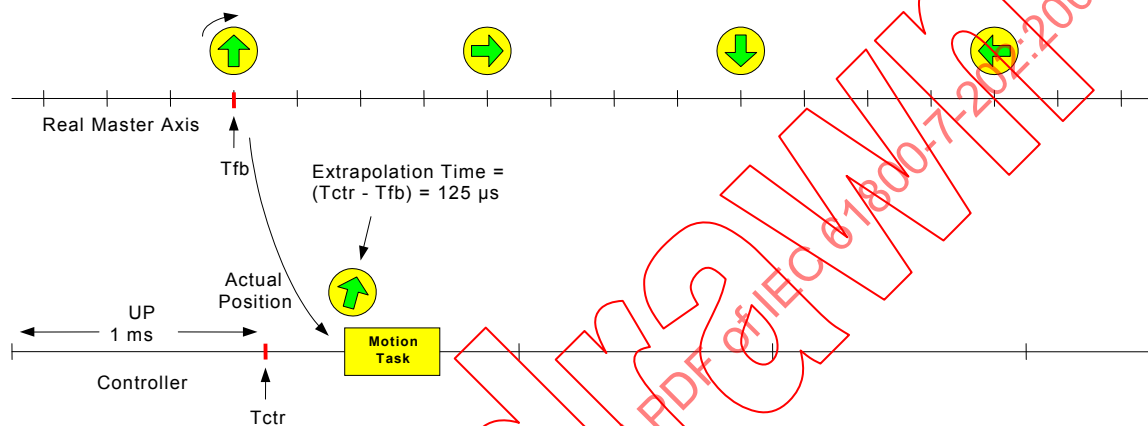


Figure 27 – Adjustment of actual position data based on device time stamp

- **Data Received Time Stamp:** This element represents the 64-bit System Time value at the moment that the last Controller-to-Device Connection data arrived at the drive device and was ready for processing. The time stamp units are nanoseconds. This value, when combined with controller's corresponding data transmit time stamp, can be used by the controller to generate Controller-to-Device Connection data delivery statistics.
- **Data Transmit Time Stamp:** This element represents the 64-bit System Time value at the moment that the Device-to-Controller Connection data is transmitted to the controller. The time stamp units are nanoseconds. This value, when combined with controller's corresponding data received time stamp, can be used by the controller to generate Device-to-Controller Connection data delivery statistics.
- **Sync Variance:** This element represents the current statistical variation of the System Time Offset value from the mean System Time Offset value.

6.4.4.3 Instance data blocks

6.4.4.3.1 General structure

After the Connection Header are one or more Instance Data Blocks as determined by the above Instance Count. The Instance Data Block is very similar to that of the Controller-to-Device Connection and has the basic structure shown in Figure 28.

Instance Data Block
Instance Data Header
Cyclic Data Block
Event Data Block
Service Data Block

Figure 28 – Instance Data Block

6.4.4.3.2 Instance data header

The Instance Data Header contains critical axis configuration information needed to parse and apply the data contained in the three data channels (see Figure 29). This header is only included in the Variable Connection format to accommodate multi-axis drive applications. Information within the header can be used by the drive communications interface to copy the individual data blocks into separate fixed memory locations for processing.

If configured for a Fixed Connection format, only the Cyclic Data Block for a single axis instance is supported so there is no need for any information on instance number or block sizing. Hence, the Instance Data Header is not included in the connection structure.

Instance Data Header			
Instance Num	-	Instance Blk Size	Cyclic Blk Size
Cyc. Act. Blk Size	Cyc. Read Blk Size	Event Blk Size	Service Blk Size

Figure 29 – Instance Data Header

- Instance Number: Same as Controller-to-Device definition.
- Instance Block Size: Same as Controller-to-Device definition.
- Cyclic Block Size: Same as Controller-to-Device definition.
- Cyclic Actual Block Size: Same as Controller-to-Device definition.
- Cyclic Read Block Size: Same as Controller-to-Device definition.
- Event Block Size: Same as Controller-to-Device definition.
- Service Block Size: Same as Controller-to-Device definition.

6.4.4.3.3 Cyclic data block

The Cyclic Data Header at the top of the Cyclic Data Block of the Device-to-Controller Connection is always included regardless of the connection format. This header contains key elements related to the content of the Cyclic Data Block and the context of the data within the block with respect to the drive (see Figure 30). Most of these elements are established by, and are therefore direct copies of, corresponding elements of the previous Controller-to-Device Connection Cyclic Data Block. Thus, the content of the Cyclic Data Block for the Device-to-Controller Connection is ultimately determined by the controller.

Cyclic Data Block			
Control Mode	Feedback Config	Axis Response	Response Status
-	Actual Data Set	Status Data Set	Axis State
(reserved)			
Cyclic Actual Data			
Cyclic Status Data			

Figure 30 – Cyclic Data Block

- Control Mode: Same as Controller-to-Device definition.
- Feedback Config: Same as Controller-to-Device definition.
- Axis Response: The 8-bit Axis Response is an enumerated value that is used for handshaking with the corresponding Axis Control element of the Controller-to-Device Connection to directly initiate drive operations that do not require a CIP service request. Valid Acknowledge Codes match the corresponding Request Codes of the Axis Control element, and are shown in Table 13.

Table 13 – Axis Response

Acknowledge Code	Axis Response
0	No Acknowledge
1	Enable Acknowledge
2	Disable Acknowledge
3	Shutdown Acknowledge
4	Shutdown Reset Acknowledge
5	Abort Acknowledge
6	Fault Reset Acknowledge
7 to 127	(Reserved)
128 to 255	(Vendor Specific)

This Axis Control/Axis Response mechanism for initiating state changes is fully described in the State Control subclause of the Motion Axis Object (see 7.6.1.2).

- **Response Status:** When there is a non-zero Acknowledge Code in the Axis Response, a Response Status value is also provided to indicate success or failure of the requested Axis Control operation. A Response Status of 0 indicates success, while a non-zero value indicates an error. The Response Status values comply with the CIP specification for General Status codes (see IEC 61158-6-2).
- **Actual Data Set:** Same as Controller-to-Device definition.
- **Status Data Set:** Same as Controller-to-Device definition.
- **Axis State:** This data element contains the enumerated Axis State value indicating the current state of this drive axis according to the Motion Axis Object State Model.
- **Cyclic Actual/Status Data:** The Cyclic Actual/Status Data contains high priority data associated with drive axis during the next drive update. This block consists of actual data elements and status data elements that are consumed by the controller as explicitly determined by the Actual Data Set and the Status Data Set elements in the Cyclic Data Header. See the Controller-to-Device definitions in 6.4.3 for details of Cyclic Actual/Status Data structure.

6.4.4.3.4 Cyclic Read data block

The Cyclic Read Data Block can be used to synchronously update one or more targeted Motion Control Axis Object attributes within the controller based on the current value of associated attributes in the drive. This mechanism can be used in conjunction with, for example, an IEC 61131-3-based Function Block program to implement sophisticated outer loop control based on a wide variety of available drive control signals. Unlike service channel Get Axis Attribute service requests, which may take several drive update cycles to process, the Cyclic Read Data mechanism guarantees the targeted parameter is updated every connection cycle.

The Cyclic Read Data Block is only supported in the Variable Connection format (see Figure 31).

Cyclic Read Data Block	
Cyclic Read Blk ID	-
Cyclic Read Data	

Figure 31 – Cyclic Read Data Block

The associated header for this block contains key elements related to the content of the Cyclic Write Data Block.

- **Cyclic Read Block ID:** This 16-bit ID determines the pre-defined Cyclic Read Block structure to apply to the Cyclic Read Data for this update. Cyclic Read Block structures are defined using the Set Cyclic Read Data List service. A successful response to this service includes a new Cyclic Read Blk ID that can be used in the next connection update to pass cyclic data in this format.
- **Cyclic Read Data:** The Cyclic Read Data contains high priority data that needs to be applied to the associated controller axis instance. This block consists of signal and status data elements that are to be scaled and applied to corresponding Motion Axis Object attributes. The contents of the Cyclic Read Data are explicitly determined by the structure identified by the Cyclic Read Blk ID found in the Cyclic Read Data Header.

6.4.4.4 Event data block

The Event Data Block allows multiple event notifications to be sent to the controller in a given update. Each event notification requires a specific event acknowledge indicating success or an error. The event notification data persists in the Device-to-Controller Connection data structure until the drive receives the corresponding event acknowledgement from the controller.

The Event Data Block for the Device-to-Controller Connection has the format shown in Figure 32.

Event Data Block			
Event Checking Status			
Event ID	Event Status	Event Type	-
Event Position			
Event Time Stamp			
...			

Figure 32 – Event Data Block

- **Event Checking Status:** This 32-bit word is a bit mapped parameter used to indicate if the drive is currently checking for events based on various drive inputs, for example marker, home, and registration inputs. Event checking is initiated when the corresponding Event Checking Control bit is set in the Controller-to-Device connection. When an event occurs, the drive captures both the time and exact axis position and passes this information to the controller in event data blocks. But for the controller to process the event data, the corresponding event checking status bit shall be set. For more detail on how this word is used in event operations, refer to the event mechanism sequence in 6.4.3.4. A complete definition for the Event Checking Status attribute can be found in 7.3.5.2.2.
- **Event ID:** Transaction number assigned to this event by the original event notification. Each event is assigned a new Event ID by incrementing the current Event ID stored in the drive. Using the Event ID, the drive is able to match the event acknowledgement to the appropriate event notification to complete the event data transaction.
- **Event Status:** Enumerated value indicating controller response to the event. A value of 0 indicates that the event was successfully processed. A non-zero value indicates that an error occurred in the event processing and the event shall be resent.
- **Event Type:** This enumerated value describes the type of event that occurred. Valid event types are as specified in Table 14.

Table 14 – Event Type

Event Type	Event Description
0	Registration 1 Positive Edge
1	Registration 1 Negative Edge
2	Registration 2 Positive Edge
3	Registration 2 Negative Edge
4	Marker Positive Edge
5	Marker Negative Edge
6	Home Switch Positive Edge
7	Home Switch Negative Edge
8	Home Switch-Marker + +
9	Home Switch-Marker + -
10	Home Switch-Marker - +
11	Home Switch-Marker - -
12 to 127	(reserved)
128 to 255	(vendor specific)

- Event Position: 32-bit integer representation of the axis position when the designated event occurred.
- Event Time Stamp: This element represents the 64-bit System Time value when the specified event occurred. The time stamp units are nanoseconds. Taken together with drive's System Time Offset value that is part of the Device-to-Controller Connection header, the controller has all the information it needs compute the absolute System Time when the event occurred.

6.4.4.5 Service data block

The service data block allows one service response per instance to be sent to the controller in a given update. Each service request requires a specific service response from the drive indicating success or an error. In some cases, the response service contains requested data. In any case, the service response data persists in the Device-to-Controller Connection data structure until the drive sees the associated service request removed from the Controller-to-Device connection instance data block (Service Block Size = 0) or a new service request is issued by the controller (incremented Transaction ID).

Each service response is represented by a block of data organised as shown in Figure 33.

NOTE Like the request structure, the structure of the service response does not follow the traditional CIP standard messaging format. That is primarily because this connection structure is, fundamentally, a CIP Implicit I/O connection, not an Explicit Messaging connection. However, the case of a Fixed Connection format, the Service Specific Request Data defined below is sent via an Explicit Messaging connection and follows the CIP rules for explicit service request format.

Service Data Block			
Transaction ID	Service Code	General Status	Extended Status
Service Specific Response Data			

Figure 33 – Service Data Block

- Transaction ID: Transaction number assigned to this service response derived from the Transaction ID of the original request. Each service request is assigned a new Transaction ID by incrementing the current Transaction ID stored in the controller. Using the Transaction ID in the response, the controller is able to match the service response to the appropriate service request.

- **Service Code:** Identifier that determines the specific service response that follows, which should match the Service code of the originating service request. A list of valid Service Codes for the Motion Axis Object is given in the Controller-to-Device Connection subclause.
- **General Status:** The General Status value is provided to indicate success or failure of the requested service request. A General Status of 0 indicates success, while a non-zero value indicates an error. The General Status values comply with the CIP specification for General Status codes (see IEC 61158-6-2).
- **Extended General Status:** The Extended General Status provides a method for defining vendor specific or service specific error codes. There is currently no standard definition for these codes.
- **Service Specific Response Data:** The format and syntax of the Service Specific Response Data depends on the specified Service Code.

6.4.5 Fixed Device connection format

By specifying a Fixed Connection Format, the CIP Motion I/O Connection can be reduced to a size that is readily applicable to lower-performance CIP Networks like DeviceNet and ControlNet. In the context of a DeviceNet or ControlNet network, the following features have been removed from the connection structure to support the requirements of a fixed connection size and limited network bandwidth.

- Time Stamping
- Node Faults/Alarms
- One Instance Only
- Dynamic Block Sizing
- Cyclic Read/Write Data Block
- Event Data Block
- Service Data Block

With Fixed Connection Format, service requests to the Motion Axis Object are supported only as an Explicit Messaging service.

Figure 34 and Figure 35 show an example of the Fixed Connection Format being used in a simple variable speed drive application requiring only a velocity command and returning actual velocity. In this case, the connection size has been reduced to 16-bytes, a size that is well suited for lower performance networks like DeviceNet.

Controller to Device Connection Structure			
Connection Format	Format Revision	Update ID	Node Control
Control Mode	Feedback Config	Axis Control	-
Command Data Set	Actual Data Set	Status Data Set	-
Command Velocity			

Figure 34 – Fixed Controller to Device Connection Format (fixed size = 16 bytes)

Device to Controller Connection Structure			
Connection Format	Format Revision	Update ID	Node Status
Control Mode	Feedback Config	Axis Response	Response Status
-	Actual Data Set	Status Data Set	Axis State
Actual Velocity			

Figure 35 – Fixed Device to Controller Connection Format (fixed size = 16 bytes)

6.4.6 CIP Motion connection timing

6.4.6.1 General

The general timing model for CIP Motion I/O Connection data exchange is described in this subclause. Data exchange between the drive and the controller is paced by the controller with one Device-to-Controller data packet sent for every Controller-to-Device data packet received. The Controller-to-Device Connection packets are sent periodically according to the configured Controller Update Period. The Device Update Period, i.e. the update period at which the drive performs its control calculations, is typically much faster than the Controller Update Period. The basic CIP Motion Connection Timing Model is illustrated in Figure 36.

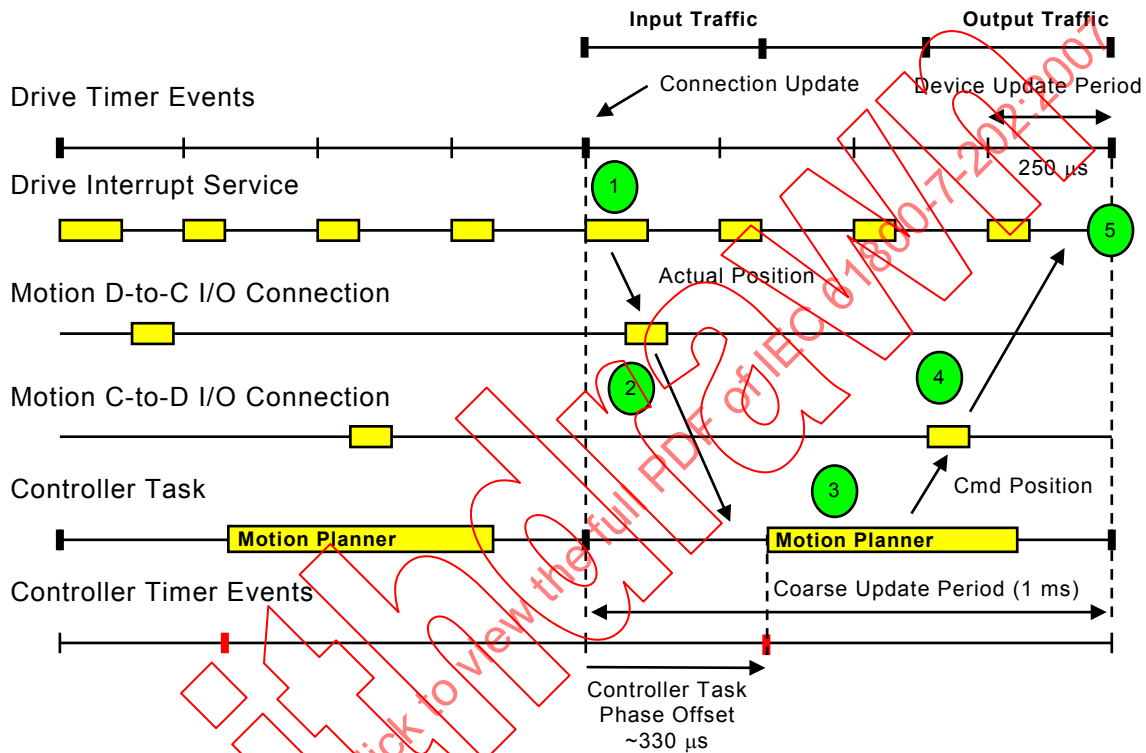


Figure 36 – CIP Motion I/O Connection timing model

The basic CIP Motion connection sequence consists of five basic steps which are shown in Figure 36 and described as follows:

- 1) A Drive Interrupt Service begins in response to a periodic Drive Timer Event and the drive determines that a Coarse Update transmission to the controller is required. The drive computes the Actual Position value (assuming Position Control Mode in this example) based on feedback count value captured in synchrony with the interrupt service timer event.
- 2) Once the Actual Position Input Data is loaded into the Device-to-Controller Connection structure, the drive transmits the data to the controller. The Input Data packet traverses the network and arrives at the controller prior to the start of the phase delayed Controller Task.
- 3) After a predetermined Phase Offset time from the start of last Controller Timer Event, which is typically set to be around 1/3 of the Coarse Update Period, the Controller Task starts. The task begins by reading the Actual Position Input Data from drive. The controller then runs a Motion Planner to compute a new Command Position value to send back to the drive. When gearing or camming operations are active, it may be necessary to use the Actual Position of a master axis as input to compute the Command Position of one or more slave axes.

- 4) Once the Command Position Output Data is loaded into the Controller-to-Device Connection structure, the controller transmits the data to the drive. The Output Data packet traverses the network and arrives at the drive prior to the start of the next Coarse Update Period.
- 5) The next Drive Interrupt Service begins and the drive determines that it has received fresh Output Data from the controller. The drive then reads the Command Position Data and applies the data along with the time stamp to the drive fine interpolator/extrapolator.

While other timing models are possible based on the controller configuration, all CIP Motion timing models begin with the drive sending actual data to the controller at the beginning of the Controller Update Period and end with command data arriving before the next Controller Update Period. However, it is not required that the command data sent to the drive in a given update period be the result of calculations performed during that period.

Most motion control protocols require the Controller Update Period to be an integer multiple of the Device Update Period. But because the CIP Motion I/O Connection packet includes a Time Stamp, the update period of the controller does not need to have any fixed relationship with the update period of the drive.

6.4.6.2 Controller-to-Device connection timing

The CIP Motion I/O Connection data exchange is initiated by the controller via the Controller-to-Device Connection packet. The inclusion of Time Stamp and Time Offset information along with the command data in this packet relieves the stringent timing requirements imposed by other motion control network protocols. Figure 37 illustrates how command data and time stamps delivered by the Controller-to-Device connection are applied to the drive axis when fine interpolation is required.

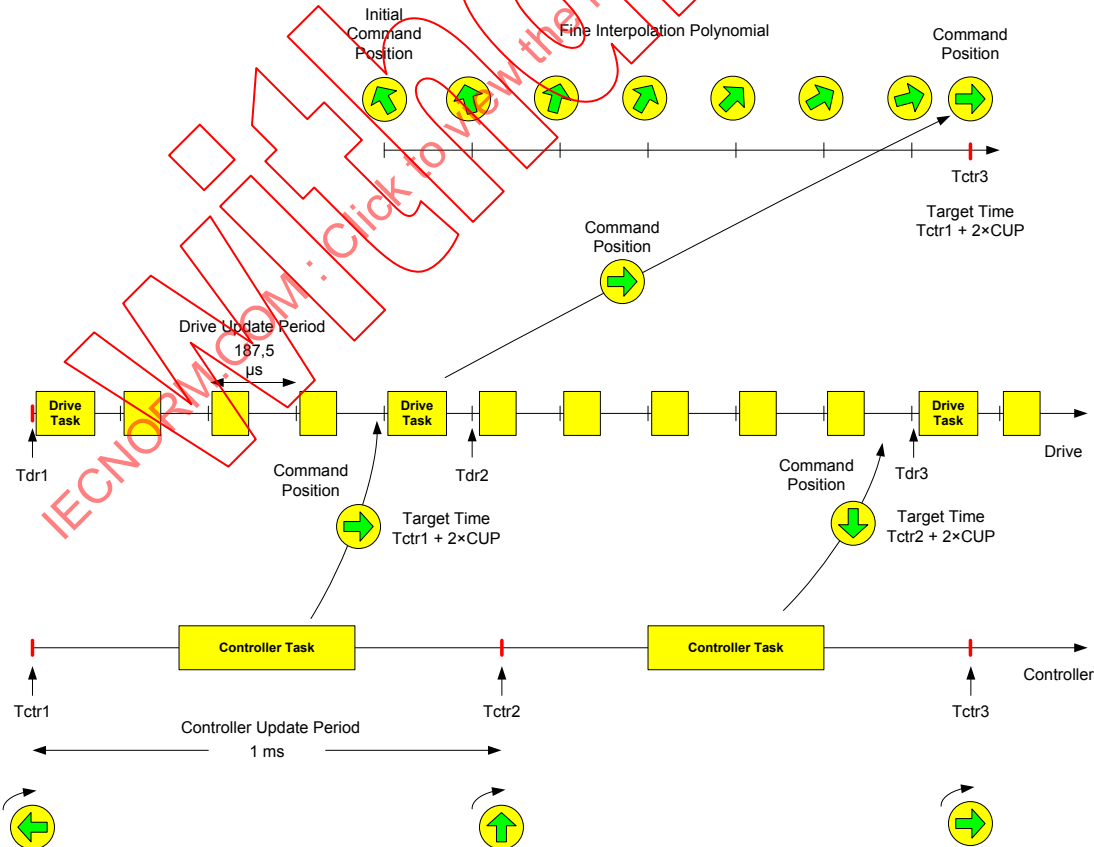


Figure 37 – Controller-to-Drive Connection timing with fine interpolation

The following steps describe in detail how connection data is transferred from the controller to the drive for fine interpolation during a typical connection cycle in the general case where the Controller Update Period (CUP) is not an integer multiple of the Device Update Period.

- 1) **Controller Transmit:** As part of the Control Task, the controller initiates transmission of a Controller-to-Device Connection packet with new command data to the targeted drive with an incremented Update ID and a new Controller System Time Stamp and Controller System Time Offset referencing the system time at the start of the current Controller Update Period. The instance data block for the targeted axis also contains the Command Target Time, which in this example is set to +2 to account for the one Controller Update Period (CUP) transport delay and the one Controller Update Period (CUP) delay for fine interpolation.
- 2) **Command Update ID Check:** The drive runs a periodic Drive Task that checks every Device Update Period for new Controller-to-Device Connection packet data. This can be easily done by checking for a changed Update ID. If the drive task discovers fresh data, then this is a command data update cycle and the command data shall be further processed according to the following sequence.
- 3) **Synchronous Operation Check:** Next, the drive checks if it is synchronised. If not synchronised, skip to the Apply Command Data step since there is no need to perform the System Time Offset Check or the Late Update Check. Bypassing these checks allows for control of the drive during start-up or even in the case where the drive does not have any time synchronisation services.
- 4) **System Time Offset Check:** Synchronous operation requires that System Time between the drive and the controller match to make the time stamps exchanged between the devices meaningful. Since System Time can be adjusted at any time by the time master and the adjustments can propagate to the different devices in the network at different times, it is possible for the System Time between the controller and the drive to be skewed. This skew shall be detected and, if present, the drive shall adjust the Controller Time Stamp to compensate for the skew. See 6.4.6.6 (System Time Offset Compensation) for details of this algorithm.
- 5) **Late Update Check:** Assuming synchronous operation, the drive computes the difference between the current drive update time stamp and the Controller Time Stamp passed in the Controller-to-Device Connection packet Stamp (adjusted, if necessary). If the difference is greater than (Controller Update Delay High Limit x Controller Update Period), the drive throws a Controller Update Fault. If the time difference has exceeded twice the Connection Update Period, the current fine interpolator polynomial has become, effectively, an extrapolator polynomial allowing the drive to ride through the late data condition until the new data arrives.
- 6) **Apply Command Data:** Assuming synchronous operation and a Command Target Time of +2 for fine interpolation, the drive computes coefficients for the fine interpolation polynomial based on the command reference being applied at the computed Target Time, which is the sum of the (adjusted) Controller Time Stamp (CUP), T_{ctr1} , and the product of the Command Target Time and Controller Update Period, or in this case, $2 \times CUP$. If the Target Time is less than the current System Time in the drive, new coefficients to the polynomial are still computed based on this command data to improve the accuracy of the extrapolation calculations. In general, whenever command data is late, the data still represents the freshest command data available and shall be applied as soon as possible. In the case of asynchronous operation, the command data is applied to the drive control structure immediately.
- 7) **Schedule Next Device-to-Controller Update:** Assuming synchronous operation, the drive schedules the time to send the next Device-to-Controller Connection packet to the controller. This transmission time target is the sum of the (adjusted) Controller Time Stamp, T_{ctrl} , plus the Controller Update Period (CUP). Since, in this general case, the Controller Update Period is not an integer multiple of the Device Update Period, the transmission time target shall be in the form timing window, defined as the Actual Update Window, that spans 1 Device Update Period and ends at the computed time of the next Controller Update. In the case of asynchronous operation, the drive simply proceeds to transmits the next Device-to-Controller Connection packet.
- 8) **Follow Device-to-Controller Timing sequence in 6.4.6.3.**

If the Command Target Time is set to +1, the computed polynomial in step 6 is not applied for the purpose of fine interpolation but rather for extrapolation; the extrapolation polynomial allows the drive to compute an accurate command data value at the time the drive performs its control calculations based on previous axis trajectory. Figure 38 illustrates this timing model in the general case where the Controller Update Period (CUP) is not an integer multiple of the drive update period.

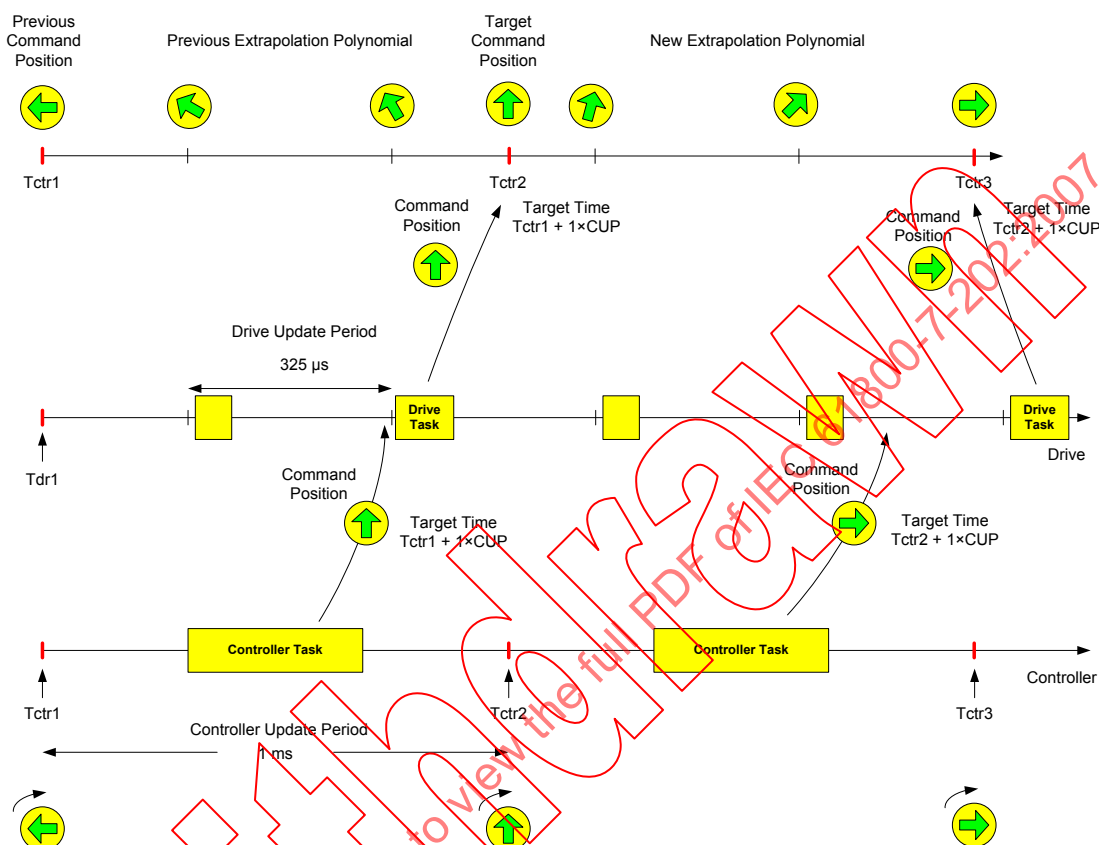


Figure 38 – Controller-to-Drive Connection timing with unequal update periods

In the above example, there are not many Device Update Periods in a given Controller Update Period. When this is the case, fine interpolation is not critical to drive performance and command data can be applied more directly to the drive's control structure without the extra delay required to support fine interpolation. Extrapolation has the disadvantage however that extrapolation error is manifested more directly to the command data resulting in rougher motion than when using fine interpolation.

All cyclic data associated with the Controller-to-Device Connection packet shall be applied in the Drive Task command update to make the earliest possible use of fresh command data, computing new interpolation/extrapolation polynomial coefficients.

6.4.6.3 Device-to-Controller connection timing

When in synchronous mode, the CIP Motion Device-to-Controller Connection includes a Device Time Stamp (and Device Time Offset) with the actual data to allow the controller to determine the position of the drive axis at the time the Controller Task update occurs. Time stamping allows the drive to sample feedback and compute actual data values based on its own Device Update Period that, unlike other motion control network protocols, does not need be strictly related to the Controller Update Period. Figure 39 illustrates how actual data and time stamps delivered by the Device-to-Controller Connection are used to adjust drive axis actual position, for example, to the controller's timebase.

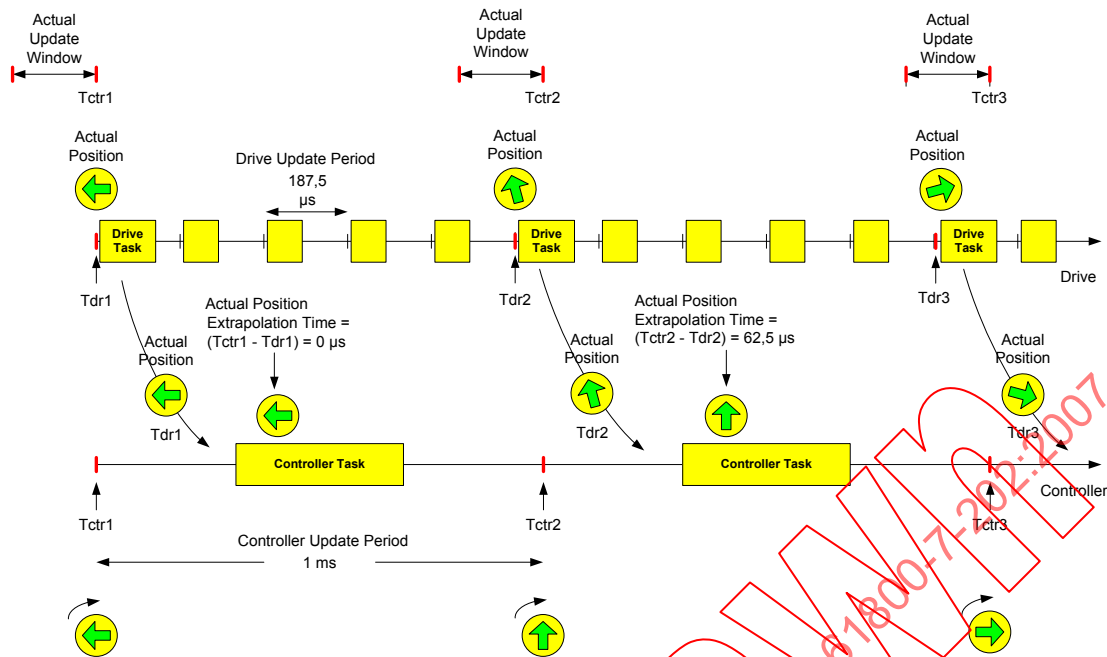


Figure 39 – Use of Time Stamp to adjust actual position to the controller's timebase

The following steps describe in detail how connection data is transferred from the drive to the controller during a typical connection cycle in the general case where the Controller Update Period (CUP) is not an integer multiple of the drive update period. This sequence of steps begins where the Controller-to-Device description ends as detailed in 6.4.6.2.

- 1) **Actual Update Window Check:** If the axis is synchronised, the drive compares current Drive Task time stamp with the Actual Update Window that is determined during the last command data update. The Actual Update Window has a duration of 1 Device Update Period and ends at the computed time of the next Controller Update. If the Drive Task time stamp is within the time window, this is an actual data update cycle. If the time stamp is before the window, then the drive shall wait for a subsequent drive task to send the actual data to the controller. (This prevents a condition where there is excessive time between the feedback capture and the start of the next Controller Task.) If the axis is not synchronised and a command update has just been received via the Controller-to-Device Connection, then this is also an actual update cycle, so the drive shall move on to the Drive Transmit step.
- 2) **Drive Transmit:** If this is an actual update cycle, then the drive sends the Device-to-Controller Connection packet to the controller with the latest Actual Data from this Drive Task, including, if synchronised, the current Device Time Stamp (and Device Time Offset), and an incremented Update ID. If not synchronised, just send the Update ID of the preceding Controller-to-Device Connection packet. All additional data sent to the controller in this packet may be derived from the previous Drive Task. This allows the drive transmission to occur at the earliest point in the Drive Task execution.
- 3) **Actual Update ID Check:** In the next Controller Task, the controller checks for new data from the drive by checking for a changed Update ID. The following steps are performed regardless of whether or not the Update ID has changed. The Update ID is the only way to detect for new actual data when the drive is not synchronised.
- 4) **Sync Flag Check:** Drive checks the Sync Mode bit of the Drive Node Status byte to determine if the drive axis is synchronised. If not, skip to the Apply Actual Data step to avoid Late Update checking and Time-Stamp Correction. Bypassing these subsequent steps allows the drive to operate during start-up or even in the case where the drive does not have any time synchronisation services.
- 5) **System Time Offset Check:** Synchronous operation requires that System Time between the drive and the controller match so as to make the time stamps exchanged between the

devices meaningful. Since System Time can be adjusted at any time by the time master and the adjustments can propagate to the different devices in the network at different times, it is possible for System Time between the controller and the drive to be skewed. This skew shall be detected, and if present, the drive shall adjust the Device Time Stamp to compensate for the skew. See 6.4.6.6 (System Time Offset Compensation) for details of this algorithm.

- 6) Late Update Check: The controller computes the difference between the current Connection Update Period time stamp and the Time Stamp in the Device-to-Controller Connection packet. If the difference is greater than (Missed Update Tolerance x Update Period), the controller throws a Controller Sync Fault.
- 7) Time-Stamp Correction: If the previously computed time difference is non-zero, then extrapolate the actual data value based on previous axis actual trajectory to line up with the controller's time stamp. This correction is necessary because the motion planner assumes that actual input data is implicitly time stamped to the beginning of the Controller Update Period.
- 8) Apply Actual Data: Controller applies actual data as inputs to the motion planner, which computes new command reference data for the next Controller-to-Device update.

6.4.6.4 Device Update Period independence

The timing diagram in Figure 40 illustrates how two drive axes can be tightly coordinated despite having different Device Update Periods and despite having an associated Controller Update Period that is not an integer multiple of either Device Update Period.

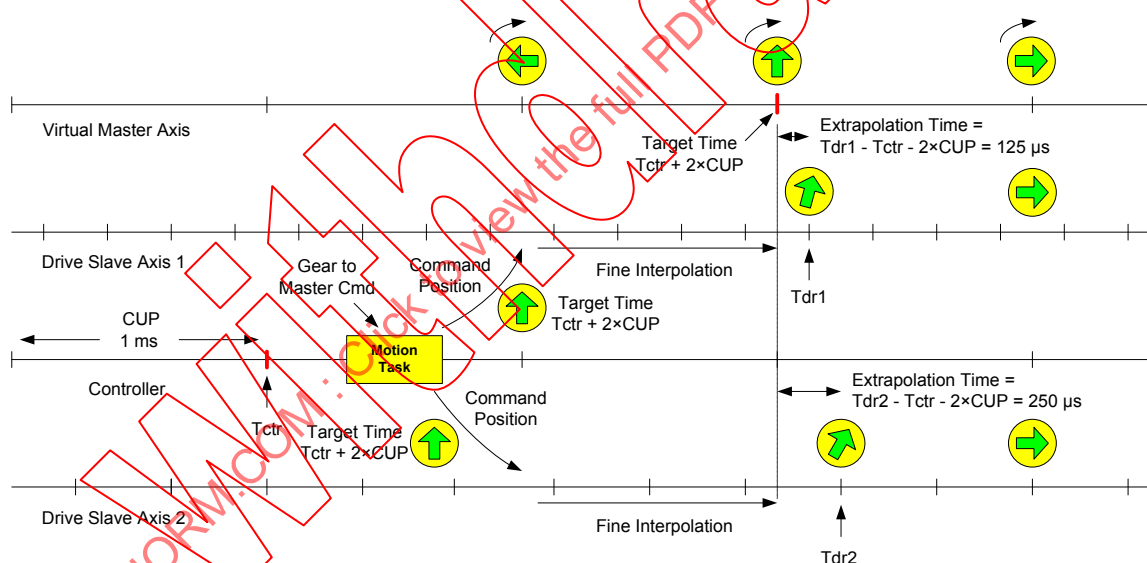


Figure 40 – Coordination of two drives with different Update Periods

In the above timing diagram, the controller's motion planner task sends identical command positions and time stamps to two slave drive axes that, while synchronised with System Time, are running at different drive update rates. When the command position data arrives at the two drives, they use the Controller Time Stamp, the Command Target Time, and the Controller Update Period to compute new coefficients to the interpolation polynomial based on the constraint that the polynomial value at time equal to (Controller Time Stamp + Command Target Time x Controller Update Period) is the specified Command Position value. Since there is no dependency on the drive update rate, the polynomial coefficients computed by each drive are identical. Since neither drive has an update that coincides with this target time, the drives use the fine interpolation polynomial to calculate the command position reference for each drive update until a fresh command position is received from the controller. If a new command position does not arrive until well after the target time, the drive continues to use the same polynomial equation to "extrapolate" command position for subsequent drive updates as shown in Figure 40. This extrapolation continues until fresh data arrives and new

coefficients can be calculated. In this way, whether by interpolation or extrapolation, each slave axis runs smoothly and the two axes stay phase locked with the master axis.

6.4.6.5 Transmission latency independence

Precise coordination of multiple CIP Motion drive axes can be maintained even when the Controller-to-Device connection packets incur significant delays while travelling across the CIP network. In Figure 41, the packet for Slave Drive Axis 2 has incurred a significant delay during transmission. As a result, the command position for this axis shall be extrapolated from the last fine interpolation polynomial. This allows the axis to move smoothly through a transmission latency disturbance. When the new command data does arrive, the new command value may not agree with extrapolated value due to extrapolation error. This error can result in a disturbance to the motion profile. The magnitude of the extrapolation error depends on the dynamics of the motion profile and the controller update rate. In most real-world applications, transmission latencies lasting several update periods can occur without any noticeable disturbance to the associated motion profile.

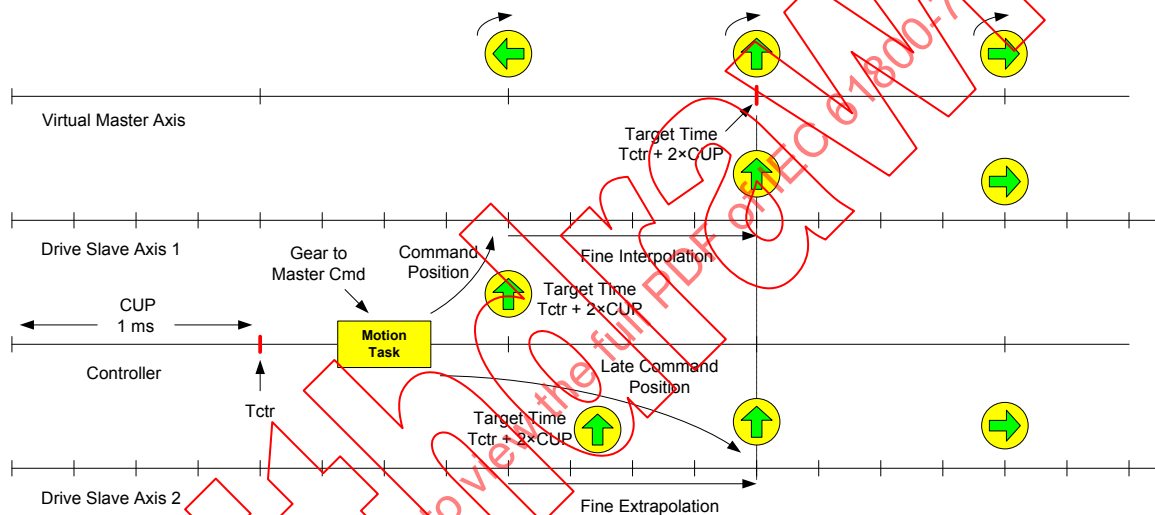


**Figure 41 – Coordination of multiple drive axes
in case of delayed Controller-to-Device Connection packets**

6.4.6.6 System Time Offset compensation

CIP Motion is built on the foundation of CIP Sync and its underlying IEC 61588:2004 time synchronisation protocol, which defines a mechanism to distribute and synchronise time for devices on a network. Unfortunately, IEC 61588:2004 makes no guarantee as to how soon all devices in the system responds to a step change in the master's time and has no provision for handling this condition. This step change in time may be caused, for example, by the user manually adjusting the master's clock due to correct drift in the clock or by a change in time mastership. According to IEC 61588:2004, step changes to the master clock time shall propagate through the distributed time system, and the time to propagate this step change in time through the system from the time master to the furthest time slave is a function of the sync packet interval (default is 2 s) and the number of hops (1 hop per IEC 61588:2004 boundary clock, for example a switch implementing the IEC 61588:2004 protocol). This is illustrated in Figure 42.

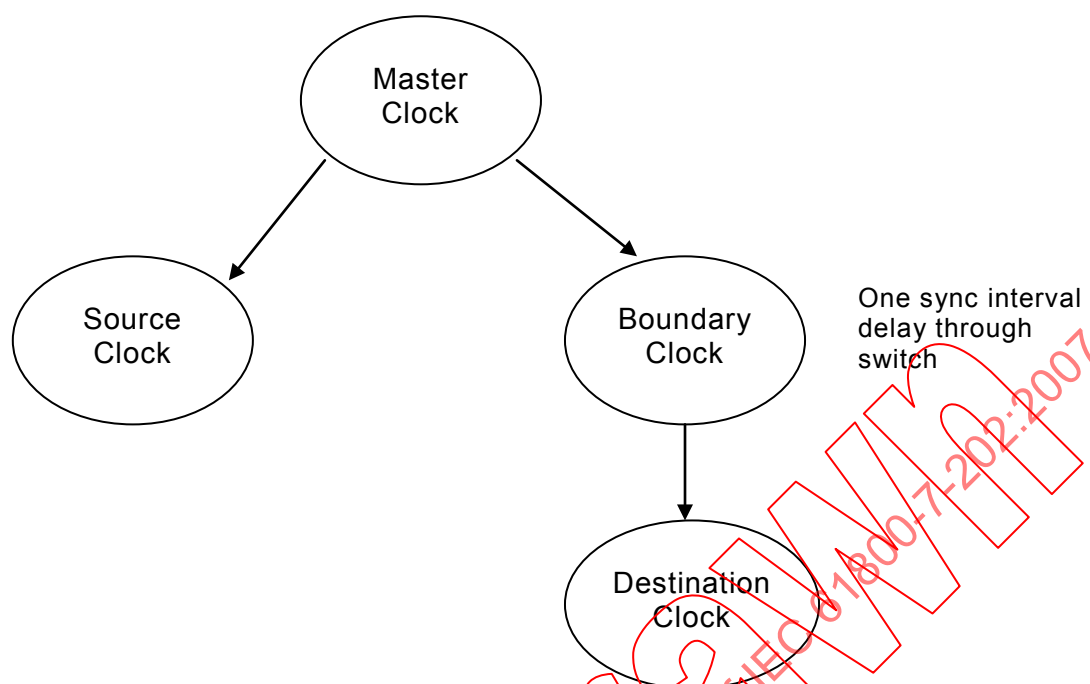


Figure 42 – Propagation of a step change in time

In Figure 42, the Source Clock is going to receive a System Time adjustment one sync interval before the Destination Clock. Thus, System Time for the Source Clock is going to be skewed relative to System Time for the Destination Clock for an entire sync packet interval.

The variability in propagating this time adjustment through the system creates a problem for time stamp based motion control in that the time stamp in the source device, say the controller, may not be applicable when received by the destination device, say a drive, because System Time for the source and destination devices is skewed. This effect is further illustrated in Table 15.

Table 15 – Propagation of a step change in time (example 1)

Time sync interval	Master time	Source time (controller)	Destination time (drive)
1	100	100	100
2	200 + 1 000	200	200
3	1 300	300 + 1 000	300
4	1 400	1 400	400 + 1 000
5	1 500	1 500	1 500

At Time Sync Intervals 1 and 2, both the timestamp source clock in the controller, and timestamp destination clock in the drive both have the same notion of time, i.e. System Time. At Sync Interval 2, a step change in time occurs at the master clock that propagates to the controller in Sync Interval 3, thus affecting the source clock time stamps sent to the drive device. It is not until Sync Interval 4 that the drive sees the step change. Any time stamp data sent from the controller to the drive between Sync Intervals 3 and 4 will not correlate with the drive's notion of time.

CIP Motion and CIP Sync extend the IEC 61588:2004 protocol to handle this condition. This is done by defining a run-time algorithm in the timestamp destination device that can detect this

time step condition and adjust the value of the received timestamp so that it is accurate. Two conditions are possible:

- 1) The source device has seen a step change in time but the destination device has not;
- 2) The destination device has seen a step change in time but the source device has not.

The key component of the algorithm is the System Time Offset value defined as part of the Time Sync Object. This offset value is added to the local clock time to generate System Time, that is,

$$\text{System Time} = \text{Local Clock} + \text{System Time Offset}$$

When the system is synchronised, at a given moment of time, the Local Clock values and the System Time Offset values may differ from device to device, but the System Time value should be the same within the accuracy of the IEC 61588:2004 protocol implementation.

A step change in time is indicated by a change in the System Time Offset value of either the source or destination devices. The source's System Time Offset is sent to the destination device as an offset along with the source's System Time as the data time stamp. The destination device compares the offset received to the previously received offset to determine if a step change has occurred and adjusts the received timestamp value accordingly.

The algorithm is stated as follows:

$$\begin{aligned} \text{Timestamp}_{\text{comp}} &= \text{Timestamp}_{\text{Rec}} + \\ &\quad ((\text{Offset}_{\text{dest}} - \text{Offset}_{\text{dest}(\text{last})}) - (\text{Offset}_{\text{src}} - \text{Offset}_{\text{src}(\text{last})})) \\ \text{If } |((\text{Offset}_{\text{dest}} - \text{Offset}_{\text{dest}(\text{last})}) - (\text{Offset}_{\text{src}} - \text{Offset}_{\text{src}(\text{last})}))| &\leq \text{Sync Threshold} \\ \text{Offset}_{\text{dest}(\text{last})} &= \text{Offset}_{\text{dest}} \\ \text{Offset}_{\text{src}(\text{last})} &= \text{Offset}_{\text{src}} \end{aligned}$$

where:

- TimestampRec = received time stamp.
- Offsetdest = the current value of the System Time Offset at the destination.
- Offsetdest(last) = the previous value of the System Time Offset at the destination.
- Offsetsrc = the received value of the System Time Offset from the source.
- Offsetsrc(last) = the previous value of the System Time Offset from the source.
- Sync Threshold = a number that defines that synchronization tolerance for the device application. For a typical CIP Motion vector drive, this number should be in the order of 1 µs.

Table 16 provides an example of how System Time Offset Compensation works. For illustration purposes, a packet is assumed to be sent at the sync interval with no propagation delay.

Table 16 – Propagation of a step change in time (example 2)

Time Sync interval	Master time	Timestamp source (controller)					Timestamp destination (drive)				
		Local clock	Offset	Last offset	Local time	Time stamp	Local clock	Offset	Last offset	Local time	Adj.time stamp
1	100	0	100	100	100	100	0	100	100	100	100
2	200 + 1 000	100	100	100	200	100	100	100	100	200	200
3	300	200	100 + 1 000	100	1 300	1 300	200	100	100	300	300
4	1 400	300	1 100	100	1 400	1 400	300	100 + 1 000	100	1 400	1 400
5	1 500	400	1 100	1 100	1 500	1 500	400	1 100	1 100	1 500	1 500

At Sync Interval 2 a step change of 1 000 occurs in the master clock and is propagated to the timestamp source clock, say the controller, in Sync Interval 3. The time stamp received at the destination clock, the drive in this example, is compensated for by 1 000 according to the above algorithm that results in an Adjusted Controller Time Stamp value that is consistent with the drive's Local Time even though it has not yet seen the step change. In Sync Interval 4, the drive sees the step change but no adjustment is made to the time stamp because in the compensation formula above the source and destination delta offset terms cancel out and the Last Offset values are updated as shown in sync interval 5 and no further offset compensation is required.

6.5 Drive startup procedure

6.5.1 General

In this subclause, the start-up process of a CIP Motion drive is discussed. For the sake of clarity, the exact messages and their formats will not be discussed. The objective is to discuss the steps and the exchanges that take place between the controller and the drive in order to correctly configure the CIP Motion drive.

The drive startup involves three distinct processes which are initiated by the CIP Controller and responded to by each CIP Motion drive on the network. They are:

- 1) Motion I/O Connection Creation;
- 2) Motion Object Configuration;
- 3) Time Synchronisation.

These processes may occur in parallel or consecutively depending on the particular implementation. Drive Configuration may occur via connected or unconnected explicit messaging. The startup process is outlined in the following subclauses.

6.5.2 Motion I/O Connection creation

Motion I/O Connections are created on EtherNet/IP using either the common CIP Forward_Open or Large_Forward_Open services (see IEC 61158-5-2 and IEC 61158-6-2). These services should specify the maximum size of the Control-to-Drive and Device-to-Controller Connection Data Structures applied to the Motion Axis Object. Configuration of the produced and consumed I/O connection paths are as described above.

6.5.3 Motion Axis Object configuration

Motion Axis Object configuration is accomplished using either the common Set_Attribute_List service or the object specific Set_Axis_Attribute_List service defined by the Motion Axis Object and supported by the CIP Motion I/O Connection.

6.5.4 Time Synchronisation

All CIP Motion drive device nodes that support time synchronisation services are required to be synchronised with a CIP Motion controller as part of a group, commonly embodied as a Motion Group. All devices in this group are sent a Group_Sync service that contains the Controller Update Period and the System Time Offset of the controller. The objective of this procedure is to guarantee that the two devices are synchronised or, in other words, that they have the same notion of time. This is accomplished by guaranteeing that no step changes to the drive device's System Time Offset value has occurred recently and the controller's System Time Offset and drive's System Time Offset values are based on a System Time reference that is consistent between the controller and the drive.

Figure 43 illustrates how the controller first waits until its local clock is synchronised to the master System Time clock and then waits a minimum delay period to make sure there are no step changes to the System Time Offset. Step changes are changes to the System Time Offset that are more than a predefined threshold defined by the Sync Threshold attribute.

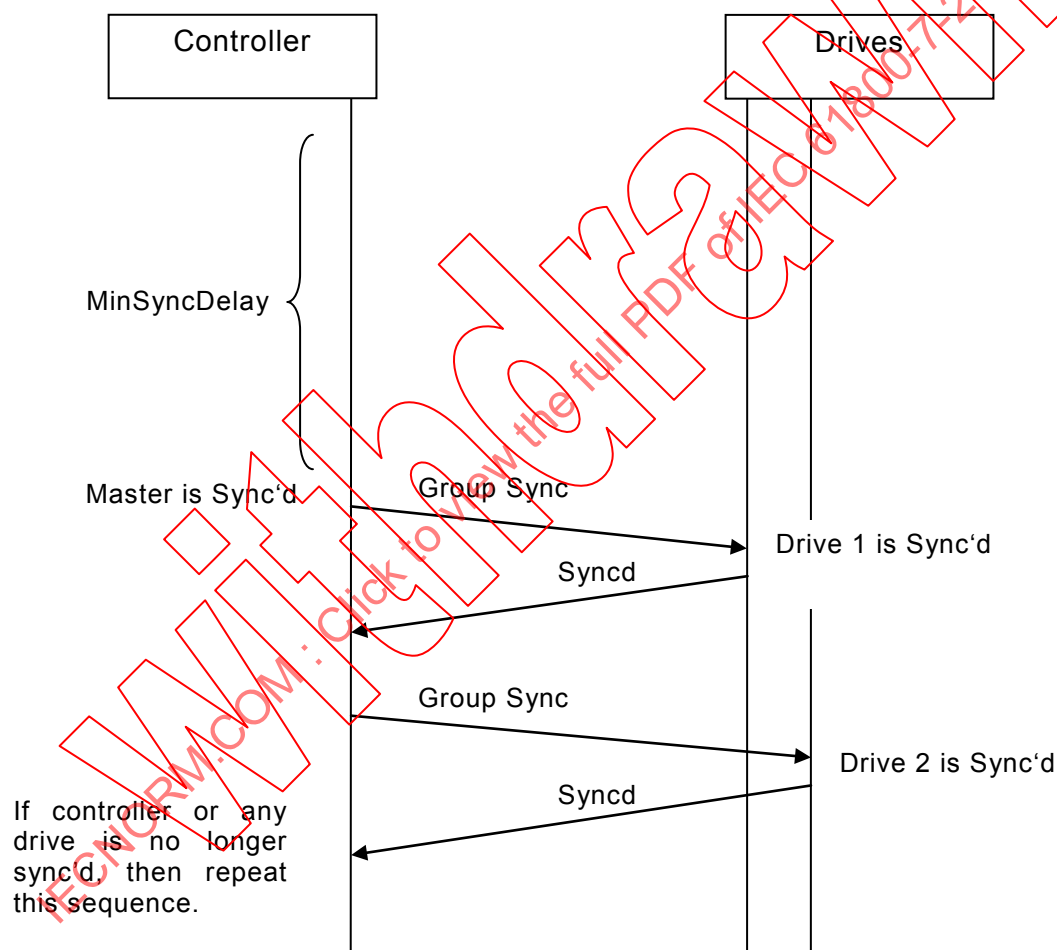


Figure 43 – Group Sync of CIP Motion drives

If there have been no System Time Offset steps during this period, the controller is considered to be synchronised and initiates a series of Group Sync services to each of the CIP Motion drive devices. Each drive device returns a response indicating whether it is also synchronised according to the same criterion described for the controller. If the controller and all drive devices indicate they are synchronised, then the Motion Group is considered to be synchronised.

There are two device specific parameters that are used by a CIP Motion device to determine if the device is group synchronised.

1. Sync Update Delay
2. Sync Threshold

The Sync Update Delay is calculated by the Motion Axis Object as the minimum time to wait after the last System Time Offset change greater than Sync Threshold before the device is considered to be group synchronised. This value is on the product of the number of networks hops the drive device is away from the grandmaster clock and the time synchronization interval. The Sync Update Delay attribute is typically in the order of seconds.

The Sync Threshold is a class attribute of the Motion Axis Object whose value is used to determine if the device's local clock is frequency synchronised to the master System Time clock. Sync Threshold depends on the IEC 61588:2004 implementation, but is typically in the order of microseconds for a hardware assisted clock implementation. Offset time changes greater than the Sync Threshold are considered step changes to the System Time Offset. Offset time changes less than the Sync Threshold are considered normal corrections to System Time to compensate for clock drift and network latency variation.

Once the controller and all the drives in the Motion Group are group synchronised, the controller can then set the Synchronous Control bit in the Controller-to-Device Connection header of the next cyclic packet sent to the drive devices indicating that the drive can schedule its next Device-to-Controller Connection update based on the Controller Time Stamp and Controller Update Period contained in the connection data. The drive device indicates that it is now sending scheduled connection data based on the Controller Update Period by setting the Synchronous Mode bit in the Device-to-Controller Connection header of the next cyclic packet sent to the controller. Data exchange between the controller and the drive proceeds thereafter according the CIP Motion Timing Model.

6.6 Device visualisation

The CIP Motion drive device requires visualisation components to quickly diagnose the condition of the drive. These components range from bicolor LEDs to multi-character alphanumeric displays. Table 17 defines the requirements for these visualisation components and appropriate labels for these components when employed by the drive. When applied to different CIP networks, the Network Status and Module Status LED behavior shall conform to the LED behavior of the associated CIP network to which the device is connected. The mapping of the device's Module Status LED states to the Motion Axis Object states is fully defined in the Motion Axis Object section on Visualisation.

Table 17 – CIP Motion visualisation components

Visualisation component	Need in implementation	Associated label
Network Status LED	Required	"Net Status" "NS"
Module Status LED	Required	"Mod Status" "MS"
Axis Status LED	Conditional ^a	"Axis Status" "XS" "Axis # Status" "XS#" ^b
Seven-Segment Display	Optional	N/A
Multi-char alphanumeric display	Optional	N/A
^a Requirement may be waived by using a multi-character alphanumeric display.		
^b Multi-axis drive devices need multiple Drive Status LEDs, one for each drive axis.		

6.7 Ethernet Quality of Service (QoS)

When a CIP Motion drive device that supports time synchronisation services is targeted for operation on EtherNet/IP, standard Ethernet QoS or Quality of Service protocol shall be utilised to guarantee that CIP Sync related IEC 61588:2004 packets and CIP Motion I/O Connection packets have timely delivery through the network components. Specifically, standard QoS on Ethernet utilise the IEEE 802.1Q tagged frame packet protocol that supports

a three-bit QoS field. In addition, CIP Motion packets are given a CIP Priority of Urgent. See IEC 61158-6-2 (Ethernet/IP Mapping Protocol Machine) for the priority specification of real time motion control messages.

7 Motion Axis Object

7.1 General considerations

7.1.1 General

This subclause defines the behaviour of a CIP object called the Motion Axis Object. This object is the main functional component of CIP Motion device profiles defined as part of the CIP Motion extensions to CIP Common, for example, the CIP Motion drive device profile. Instances of this object are required to support motion control when connected to a CIP Motion compliant controller through a CIP network. Applicable CIP networks are DeviceNet, ControlNet, and EtherNet/IP.

7.1.2 Object overview

The Motion Axis Object is one of two major object components required to support motion control under the CIP Motion control architecture. These object components are illustrated in Figure 44.

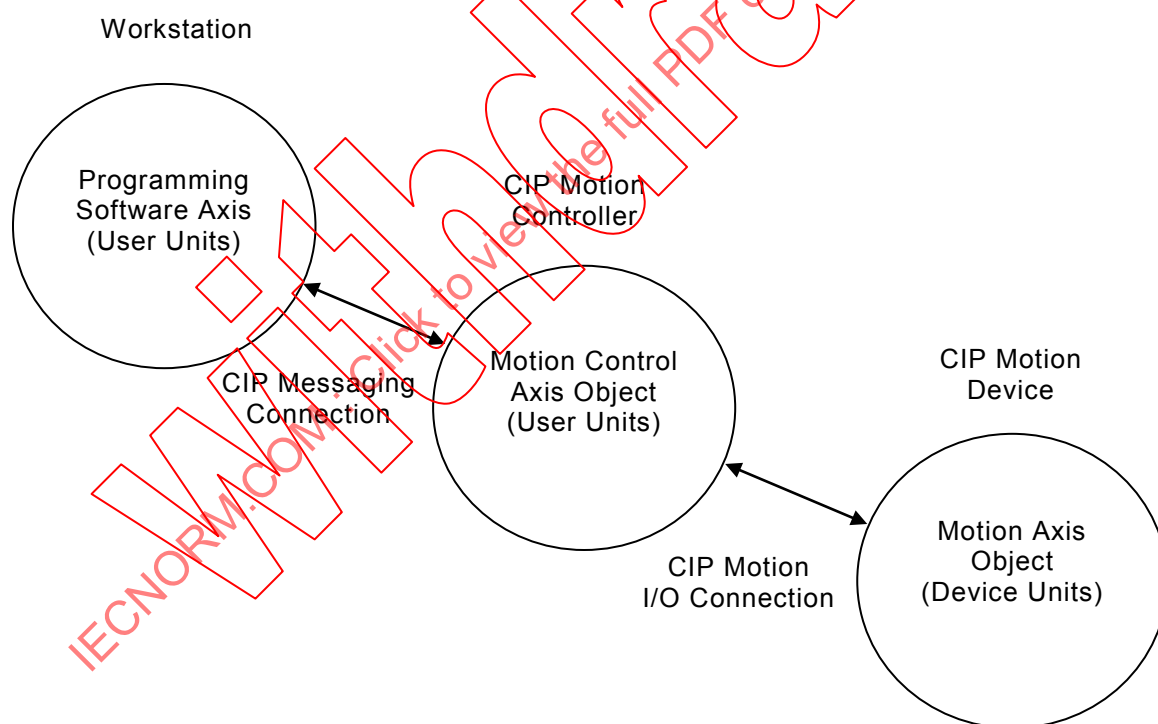


Figure 44 – Object components for CIP motion architecture

7.1.3 Motion Axis Object abstraction

The Motion Axis Object is an abstraction that can be applied to any moving component associated with a motion control system. A moving component can therefore be represented by a Motion Axis Object instance, sometimes simply referred to as an “Axis”. A Motion Axis Object instance generally consists of one or more of the following integral elements:

- 1) Motion actuator, for example rotary and linear motors, hydraulic and pneumatic actuators;
- 2) Motion sensor, for example encoders, resolvers, linear displacement transducers;

- 3) Actuator control, for example vector, frequency, and stepper motor control, hydraulic valve control;
- 4) Amplifier that supplies power to move actuator, for example drive power structure for motors.

Instance attributes are included in the Motion Axis Object for each of these integral elements to define axis behavior. When a Motion Axis Object has an actuator control element, command attributes are included in object to control the dynamics of the actuator. The object also includes attributes that describe the dynamics of the actuator, whether estimated or actual.

While the most common embodiment of a Motion Axis Object is a “drive” device as described by the CIP Motion drive device profile, the Motion Axis Object can be applied to other motion related devices, for example a single channel or multi-channel feedback device. In the case of a multi-channel feedback device, each channel is associated with an axis and therefore assigned an instance of the Motion Axis Object. In the case of a multi-amplifier drive device, each amplifier is considered an axis and assigned an instance of the Motion Axis Object. For drive axis instances, the Motion Axis Object can represent a complex closed loop vector controlled servo drive or a simple open loop Volts/Hertz drive (or VFD). Drive devices are often equipped with multiple feedback channels that can be dedicated to drive amplifier instances or used as independent master feedback sources. In the latter case, the master feedback channel is considered an axis and assigned an instance of the Motion Axis Object.

7.1.4 Motion Control Axis Object

For each device resident Motion Axis Object instance in the motion control system, there is a corresponding controller resident Motion Control Axis Object. Attribute values contained within the Motion Axis Object have matching attributes in the Motion Control Axis Object, the only difference being that the units by which these attributes are expressed. In the Motion Axis Object, attributes are expressed in fixed device units, for example feedback counts or RPM, as defined by the Motion Axis Object. Within the Motion Control Axis Object, attributes are expressed in user defined units, for example inches or metres. Since programming software interacts with the device through the Motion Control Axis Object, all configuration, diagnostic, and motion programming is done using the user's preferred units. Therefore, rather than programming the motion control application in, say, feedback counts or RPM of the motor shaft, the user can program in, for example, centimeters, feet, products per minute, or whatever makes sense for the specific application.

When configuration or command attribute values are transferred to the Motion Axis Object via the CIP Motion I/O Connection, these attribute values are scaled by the Motion Control Axis Object into fixed device unit values required by the Motion Axis Object. Similarly, when signal attributes are transferred to the Motion Control Axis Object via the CIP Motion I/O Connection, these feedback-based unit values are scaled by the Motion Control Axis Object and stored in the object in user units.

7.1.5 Drive control classification

Based on the variations in Control Mode and Control Method listed in 4.2, one can derive some basic Device Control Codes around which the many attributes of the Motion Axis Object can be organized. Device Control Codes can be designated using a single letter identifier that can be used to determine what object attributes are required for implementation of a given CIP Motion device. The list of Device Control Codes is as follows:

Control Modes:

- P** – Position Control
- V** – Velocity Control
- T** – Torque Control
- N** – No Control (Feedback only)

Control Methods:

F – Frequency Controlled Drives (Open Loop V/Hz or VFD)

C – Closed Loop Vector Controlled Drives

Combinations of these letters can be used to designate a specific class of CIP Motion devices for the purposes of identifying applicable attributes. For example, “F” would indicate a variable frequency drive axis, “V” would refer to velocity controlled drive axis, either open loop or closed loop, and “N” would refer to an axis with feedback only functionality.

7.1.6 Required vs. Optional in implementation

In the subclauses that follow, attributes and services are defined as Required or Optional in the implementation of the Motion Axis Object. Required attributes and services shall be supported in the implementation of the object. Optional attributes and services may or may not be supported in the implementation and are left to the discretion of the vendor.

The determination of whether a given attribute or service is Required or Optional often depends on the associated Device Control Code as defined above. If an attribute or service is marked as Required for a given Device Control Code, then a device implementation shall support that attribute or service if it is intended to operate in that mode. For example, an attribute marked as required for Device Control Code “V” shall be supported by any CIP motion device that supports Velocity control mode.

In some cases, an attribute or service may not even be applicable to a Device Control Code. This situation is implied when the attribute is defined as neither Required nor Optional.

Key to Table 18: (See 7.1.5 and 7.1.6 for details.)

(R) – Required bit or enumeration – shall be supported in the implementation.

(O) – Optional bit or enumeration – support is left to vendor’s discretion.

Set* – Indicates the attribute is normally set by the CIP Motion connection data block and not by a Set service.

Table 18 – Instance attribute implementation vs. Device Control Code

Instance attribute			Implementation by Device Control Code					
Attr. ID	Acc. rule	Attribute name	N	F	C-PI Vector Ctrl			Conditional implementation
				V	P	V	T	
80	Set*	Control Mode	-	R	R	R	R	
81	Set	Control Method	-	R	R	R	R	
1310	Get	Motor Catalog Number	-	O	O	O	O	
1311	Get	Motor Serial Number	-	O	O	O	O	
1312	Get	Motor Date Code	-	O	O	O	O	
1313	Set	Motor Data Source	-	R	R	R	R	
1314	Set	Motor Device Code	-	R	R	R	R	
1315	Set	Motor Type	-	R	R	R	R	
1316	Set	Motor Unit	-	O	O	O	O	
1317	Set	Motor Polarity	-	R	R	R	R	
1318	Set	Motor Rated Voltage	-	R	R	R	R	
1319	Set	Motor Rated Cont. Current	-	R	R	R	R	
1320	Set	Motor Rated Peak Current	-	C	C	C	C	R – PM; O - IM
1321	Set	Motor Rated Output Power	-	C	C	C	C	R – PM; O - IM

Instance attribute			Implementation by Device Control Code					
Attr. ID	Acc. rule	Attribute name	N	F	C-PI Vector Ctrl			Conditional implementation
				V	P	V	T	
1322	Set	Motor Overload Limit	-	O	O	O	O	
1323	Set	Motor Int. Thermal Switch	-	O	O	O	O	
1324	Set	Motor Max Winding Temp.	-	O	O	O	O	
1325	Set	Motor Wind-to-Amb. Therm. Cap.	-	O	O	O	O	
1326	Set	Motor Wind-to-Amb. Therm. Res.	-	O	O	O	O	
1327	Set	PM Motor Resistance	-	R	R	R	R	PM Motor only
1328	Set	PM Motor Inductance	-	R	R	R	R	PM Motor only
1329	Set	Rotary Motor Poles	-	R	R	R	R	Rotary Motor only
1330	Set	Rotary Motor Inertia	-	O	O	O	O	Rotary Motor only
1331	Set	Rotary Motor Rated Speed	-	R	R	R	R	Rotary Motor only
1332	Set	Rotary Motor Max Speed	-	O	O	O	O	Rotary Motor only
1333	Set	Rotary Motor Damp. Coef.	-	O	O	O	O	Rotary Motor only
1334	Set	Linear Motor Pole Pitch	-	R	R	R	R	Linear Motor only
1335	Set	Linear Motor Rated Speed	-	R	R	R	R	Linear Motor only
1336	Set	Linear Motor Moving Mass	-	O	O	O	O	Linear Motor only
1337	Set	Linear Motor Max Speed	-	O	O	O	O	Linear Motor only
1338	Set	Linear Motor Damp. Coef.	-	O	O	O	O	Linear Motor only
1339	Set	PM Motor Rated Torque	-	R	R	R	R	Rotary PM Motor only
1340	Set	PM Motor Torque Constant	-	R	R	R	R	Rotary PM Motor only
1341	Set	PM Motor Rotary Voltage Constant	-	R	R	R	R	Rotary PM Motor only
1342	Set	PM Motor Rated Force	-	R	R	R	R	Linear PM Motor only
1343	Set	PM Motor Force Const.	-	R	R	R	R	Linear PM Motor only
1344	Set	PM Motor Linear Voltage Constant	-	R	R	R	R	Linear PM Motor only
1345	Set	Ind. Motor Rated Frequency	-	R	R	R	R	Induction Motor only
1346	Set	Ind. Motor Flux Current	-	R	R	R	R	Induction Motor only
1347	Set	Ind. Motor Stator Resistance	-	R	R	R	R	Induction Motor only
1348	Set	Ind. Motor Stator Leakage X	-	R	R	R	R	Induction Motor only
1349	Set	Ind. Motor Magnetization X	-	R	R	R	R	Induction Motor only
1350	Set	Ind. Motor Rotor Resistance	-	R	R	R	R	Induction Motor only
1351	Set	Ind. Motor Rotor Leakage X	-	R	R	R	R	Induction Motor only
1400 + o	Get	Feedback n Catalog Number	-	O	O	O	O	
1401 + o	Get	Feedback n Serial Number	-	O	O	O	O	
1402 + o	Get	Feedback n Position	R	-	R	R	O	
1403 + o	Get	Feedback n Velocity	R	-	R	R	O	
1404 + o	Get	Feedback n Acceleration	R	-	R	R	O	
90	Set*	Feedback Data Set	R	R	R	R	R	
82	Set*	Feedback Configuration	R	R	R	R	R	
83	Set*	Feedback Master Select	R	-	R	R	R	
84	Set	Feedback 1/2 Count Ratio	-	-	O	-	-	
1411 + o	Set	Feedback n Unit	O	-	O	O	O	
1412 + o	Set	Feedback n Port Select	O	-	O	O	O	
1413 + o	Set	Feedback n Type	R	-	R	R	R	

Instance attribute			Implementation by Device Control Code					
Attr. ID	Acc. rule	Attribute name	N	F	C-PI Vector Ctrl			Conditional implementation
				V	P	V	T	
1414 + o	Set	Feedback n Polarity	O	-	O	O	O	
1415 + o	Set	Feedback n Mode	R	-	R	R	R	
1416 + o	Set	Feedback n Cycle Resolution	R	-	R	R	R	TT, SC, HI, E21, RS
1417 + o	Set	Feedback n Cycle Interpolation	R	-	R	R	R	TT, SC, HI, E21, RS
1420 + o	Set	Feedback n Data Length	R	-	R	R	R	TP, SS, HI, E21, E22
1421 + o	Set	Feedback n Data Code	O	-	O	O	O	TP, SS
1422 + o	Set	Feedback n Resolver Trans. Ratio	O	-	O	O	O	RS
1423 + o	Set	Feedback n Resolver Excit. Voltage	O	-	O	O	O	RS
1424 + o	Set	Feedback n Resolver Excit Freq.	O	-	O	O	O	RS
1425 + o	Set	Feedback n Resolver Cable Balance	O	-	O	O	O	RS
1426 + o	Set	Feedback n LDT Type	R	-	R	R	R	LT
1427 + o	Set	Feedback n LDT Recirculations	R	-	R	R	R	LT
1428 + o	Set	Feedback n Commutation Offset	R	-	R	R	R	HS, TP, HI, E21, E22, SS, RS (PM Only)
1434 + o	Set	Feedback n Velocity Filter BW.	O	-	O	O	O	
1435 + o	Set	Feedback n Accel Filter BW	O	-	O	O	O	
256	Set*	Event Checking Control	O	-	O	-	-	
257	Get	Event Checking Status	O	-	O	-	-	
258	Get	Registration 1 Pos. Edge Position	R	-	R	-	-	
259	Get	Registration 1 Neg. Edge Position	R	-	R	-	-	
260	Get	Registration 2 Pos. Edge Position	O	-	O	-	-	
261	Get	Registration 2 Neg. Edge Position	O	-	O	-	-	
262	Get	Registration 1 Pos. Edge Time	O	-	O	-	-	
263	Get	Registration 1 Neg. Edge Time	R	-	R	-	-	
264	Get	Registration 2 Pos. Edge Time	R	-	R	-	-	
265	Get	Registration 2 Neg. Edge Time	O	-	O	-	-	
266	Get	Home Event Position	R	-	R	-	-	
268	Get	Home Event Time	O	-	O	-	-	
93	Get	Command Target Time	-	-	O	-	-	
286	Set*	Controller Position Command	-	-	R	-	-	
287	Set*	Controller Velocity Command	-	-	O	R	-	
288	Set*	Controller Acceleration Command	-	-	O	O	O	
289	Set*	Controller Torque Command	-	-	O	O	O	
290	Get	Interpolated Command Position	-	-	O	-	-	
291	Get	Interpolated Command Velocity	-	-	O	O	-	
292	Get	Interpolated Command Accel.	-	-	O	O	O	
91	Set*	Command Data Set	-	-	R	R	R	
92	Set*	Interpolation Control	-	-	R	R	-	
302	Set	Velocity Limit - Positive	-	-	O	R	-	
303	Set	Velocity Limit - Negative	-	-	O	R	-	
304	Set	Acceleration Limit	-	-	O	R	-	
305	Set	Deceleration Limit	-	-	O	R	-	

Instance attribute			Implementation by Device Control Code					
Attr. ID	Acc. rule	Attribute name	N	F	C-PI Vector Ctrl			Conditional implementation
				V	P	V	T	
306	Set	Jerk Limit Control	-	-	O	O	-	
307	Set	Flying Start Enable	-	-	O	O	-	
308	Set	Skip Speed 1	-	O	-	-	-	
309	Set	Skip Speed 2	-	O	-	-	-	
310	Set	Skip Speed 3	-	O	-	-	-	
311	Set	Skip Speed Band	-	O	-	-	-	
420	Get	Position Command	-	-	R	-	-	
421	Set*	Position Trim	-	-	R	-	-	
422	Get	Position Reference	-	-	R	-	-	
423	Get	Velocity Feedforward Command	-	-	R	-	-	
424	Get	Position Feedback	R	-	R	R	O	
426	Get	Position Error	-	-	R	-	-	
427	Get	Position Integrator Output	-	-	R	-	-	
428	Get	Position Loop Output	-	-	R	-	-	
430	Set	Kvff	-	-	R	-	-	
431	Set	Kpp	-	-	R	-	-	
432	Set	Kpi	-	-	R	-	-	
433	Set	Position Lock Tolerance	-	-	R	-	-	
434	Set	Position Error Tolerance	-	-	R	-	-	
435	Set	Position Error Tolerance Time	-	-	O	-	-	
439	Set	Position Integrator Control	-	-	R	-	-	
440	Set	Position Integrator Preload	-	-	R	-	-	
450	Get	Velocity Command	-	R	R	R	-	
451	Set*	Velocity Trim	-	R	R	R	-	
452	Get	Accel. Feedforward Command	-	-	R	R	-	
453	Get	Velocity Reference	-	R	R	R	-	
454	Get	Velocity Feedback	R	-	R	R	O	
455	Get	Velocity Error	-	-	R	R	-	
456	Get	Velocity Integrator Output	-	-	R	R	-	
457	Get	Velocity Loop Output	-	-	R	R	-	
460	Set	Kaff	-	-	R	-	-	
461	Set	Kvp	-	-	R	R	-	
462	Set	Kvi	-	-	R	R	-	
464	Set	Kdr	-	R	O	R	-	
465	Set	Velocity Error Tolerance	-	-	O	O	-	
466	Set	Velocity Error Tolerance Time	-	-	O	O	-	
467	Set	Velocity Integrator Control	-	-	R	R	-	
468	Set	Velocity Integrator Preload	-	-	R	R	-	
469	Set	Velocity Low Pass Filter BW.	-	-	O	O	-	
470	Set	Velocity Threshold	-	O	R	R	-	
471	Set	Velocity Lock Tolerance	-	O	R	R	-	
472	Set	Velocity Standstill Window	-	R	R	R	-	

Instance attribute			Implementation by Device Control Code					
Attr. ID	Acc. rule	Attribute name	N	F	C-PI Vector Ctrl			Conditional implementation
				V	P	V	T	
480	Get	Acceleration Command	-	-	O	O	O	
481	Set*	Acceleration Trim	-	-	O	O	O	
482	Get	Acceleration Reference	-	-	O	O	O	
483	Get	Acceleration Feedback	R	-	R	R	O	
490	Get	Torque Command	-	-	R	R	R	
491	Set*	Torque Trim	-	-	R	R	R	
492	Get	Torque Reference	-	-	R	R	R	
493	Get	Filtered Torque Reference	-	-	R	R	R	
494	Get	Limited Torque Reference	-	-	R	R	R	
496	Set	Kj	-	-	R	R	O	
498	Set	Friction Compensation	-	-	O	O	O	
502	Set	Torque LP Filter Bandwidth	-	-	O	O	O	
503	Set	Torque Cmd Notch Freq.	-	-	O	O	O	
504	Set	Torque Limit - Positive	-	-	R	R	R	
505	Set	Torque Limit - Negative	-	-	R	R	R	
506	Set	Torque Rate Limit	-	-	O	O	O	
507	Set	Torque Threshold	-	-	O	O	O	
508	Set	Overtorque Limit	-	-	O	O	O	
509	Set	Overtorque Limit Time	-	-	O	O	O	
510	Set	Undertorque Limit	-	-	O	O	O	
511	Set	Undertorque Limit Time	-	-	O	O	O	
520	Get	Iq Current Command	-	-	R	R	R	
521	Get	Operative Current Limit	-	-	O	O	O	
522	Get	Current Limit Source	-	-	O	O	O	
523	Get	Motor Electrical Angle	-	-	R	R	R	
524	Get	Iq Current Reference	-	-	O	O	O	
525	Get	Id Current Reference	-	-	O	O	O	
527	Get	Iq Current Error	-	-	O	O	O	
528	Get	Id Current Error	-	-	O	O	O	
529	Get	Iq Current Feedback	-	-	O	O	O	
530	Get	Id Current Feedback	-	-	O	O	O	
531	Get	Iq Decoupling	-	-	O	O	O	
532	Get	Id Decoupling	-	-	O	O	O	
533	Get	Vq Voltage Output	-	-	O	O	O	
534	Get	Vd Voltage Output	-	-	O	O	O	
535	Get	U Voltage Output	-	-	O	O	O	
536	Get	V Voltage Output	-	-	O	O	O	
537	Get	W Voltage Output	-	-	O	O	O	
538	Get	U Current Feedback	-	-	O	O	O	
539	Get	V Current Feedback	-	-	O	O	O	
540	Get	W Current Feedback	-	-	O	O	O	
541	Get	U Current Offset	-	-	O	O	O	

Instance attribute			Implementation by Device Control Code					
Attr. ID	Acc. rule	Attribute name	N	F	C-PI Vector Ctrl			Conditional implementation
				V	P	V	T	
542	Get	V Current Offset	-	-	O	O	O	
543	Get	W Current Offset	-	-	O	O	O	
554	Get	Kqp	-	-	O	O	O	
555	Get	Kqi	-	-	O	O	O	
556	Get	Kdp	-	-	O	O	O	
557	Get	Kdi	-	-	O	O	O	
558	Set	Flux Up Control	-	-	O	O	O	Induction Motor only
559	Set	Flux Up Time	-	-	O	O	O	Induction Motor only
570	Get	Slip Compensation	-	R	-	-	-	
575	Set	Frequency Control Method	-	R	-	-	-	
571	Set	Rated Slip Speed	-	R	-	-	-	
572	Set	Maximum Voltage	-	R	-	-	-	
573	Set	Maximum Frequency	-	R	-	-	-	
574	Set	Frequency Limit	-	R	-	-	-	
575	Set	Break Voltage	-	R	-	-	-	
576	Set	Break Frequency	-	R	-	-	-	
577	Set	Start Boost	-	R	-	-	-	
578	Set	Run Boost	-	R	-	-	-	
600	Get	Output Frequency	-	R	-	-	-	
601	Get	Output Current	-	R	R	R	R	
602	Get	Output Voltage	-	R	R	R	R	
603	Get	Output Power	-	R	R	R	R	
604	Set	PWM Frequency	-	O	O	O	O	
610	Set	Stopping Mode	-	R	R	R	R	
611	Set	Stopping Current	-	-	R	R	R	
612	Set	Stopping Time Limit	-	-	R	R	R	
613	Set	Resistive Brake Contact Delay	-	O	O	O	O	PM Motors Only
614	Set	Mechanical Brake Control	-	O	O	O	O	
615	Set	Mechanical Brake Release Delay	-	R	R	R	R	
616	Set	Mechanical Brake Engage Delay	-	R	R	R	R	
620	Get	DC Bus Voltage	-	R	R	R	R	
621	Get	DC Bus Voltage - Nominal	-	R	R	R	R	
624	Set	Bus Regulator Action	-	R	R	R	R	
625	Set	Bus Regulator Power Limit	-	O	O	O	O	
627	Set	Power Loss Action	-	O	O	O	O	
628	Set	Power Loss Threshold	-	O	O	O	O	
629	Set	Shutdown Action	-	O	O	O	O	
635	Get	Motor Capacity	-	R	R	R	R	
636	Get	Inverter Capacity	-	R	R	R	R	
637	Get	Converter Capacity	-	O	O	O	O	
638	Get	Bus Regulator Capacity	-	O	O	O	O	
639	Get	Ambient Temperature	-	O	O	O	O	

Instance attribute			Implementation by Device Control Code					
Attr. ID	Acc. rule	Attribute name	N	F	C-PI Vector Ctrl			Conditional implementation
				V	P	V	T	
640	Get	Inverter Heatsink Temperature	-	O	O	O	O	
641	Get	Inverter Temperature	-	O	O	O	O	
642	Get	Motor Temperature	-	O	O	O	O	
643	Get	Feedback 1 Temperature	-	O	O	O	O	
644	Get	Feedback 2 Temperature	-	O	O	O	O	
645	Get	Inverter Overload Limit	-	O	O	O	O	
646	Set	Motor Overload Action	-	O	O	O	O	
647	Set	Inverter Overload Action	-	O	O	O	O	
650	Get	Axis State	R	R	R	R	R	
651	Get	Axis Status	R	R	R	R	R	
652	Get	Axis Status - Mfg	R	R	R	R	R	
653	Get	Axis I/O Status	R	R	R	R	R	
654	Get	Axis I/O Status - Mfg	R	R	R	R	R	
94	Set*	Status Data Set	R	R	R	R	R	
655	Get	Axis Exception Status	R	R	R	R	R	
656	Get	Axis Exception Status - Mfg	R	R	R	R	R	
657	Get	Axis Fault Status	R	R	R	R	R	
658	Get	Axis Fault Status - Mfg	R	R	R	R	R	
659	Get	Axis Alarm Status	R	R	R	R	R	
660	Get	Axis Alarm Status - Mfg	R	R	R	R	R	
661	Get	Axis Fault Code	R	R	R	R	R	
662	Get	Axis Fault Code - Mfg	R	R	R	R	R	
663	Get	Axis Fault Time Stamp	R	R	R	R	R	
664	Get	Axis Alarm Code	R	R	R	R	R	
665	Get	Axis Alarm Code - Mfg	R	R	R	R	R	
666	Get	Fault Sub Code	O	O	O	O	O	
680	Get	Motor Overtemp. Factory Limit	-	O	O	O	O	
681	Get	Motor T. Overld. Factory Limit	-	O	O	O	O	
682	Get	Inverter Overtemp. Factory Limit	-	O	O	O	O	
683	Get	Inverter T. Overld. Factory Limit	-	O	O	O	O	
684	Get	Converter Overtemp. Factory Limit	-	O	O	O	O	
685	Get	Converter T. Overld. Factory Limit	-	O	O	O	O	
686	Get	Bus Reg. Overtemp. Factory Limit	-	O	O	O	O	
687	Get	Bus Reg. T. Overld. Factory Limit	-	O	O	O	O	
688	Get	Bus Overvoltage Factory Limit	-	O	O	O	O	
689	Get	Bus Undervoltage Factory Limit	-	O	O	O	O	
695	Set	Motor Overspeed User Limit	-	O	O	O	O	
696	Set	Motor Overtemperature User Limit	-	O	O	O	O	
697	Set	Motor T. Overld. User Limit	-	O	O	O	O	
698	Set	Inverter Overtemp. User Limit	-	O	O	O	O	
699	Set	Inverter T. Overld User Limit	-	O	O	O	O	
700	Set	Converter Overtemp. User Limit	-	O	O	O	O	

Instance attribute			Implementation by Device Control Code					
Attr. ID	Acc. rule	Attribute name	N	F	C-PI Vector Ctrl			Conditional implementation
				V	P	V	T	
701	Set	Converter T. Overld. User Limit	-	O	O	O	O	
702	Set	Bus Reg. Overtemp. User Limit	-	O	O	O	O	
703	Set	Bus Reg. T. Overld. User Limit	-	O	O	O	O	
704	Get	Bus Overvoltage User Limit	-	O	O	O	O	
705	Get	Bus Undervoltage User Limit	-	O	O	O	O	
667	Set	Exception Action	R	R	R	R	R	
668	Set	Exception Action - Mfg	R	R	R	R	R	
669	Get	Initialization Fault Status	R	R	R	R	R	
670	Get	Initialization Fault Status - Mfg	R	R	R	R	R	
671	Get	Initialization Fault Code	R	R	R	R	R	
672	Get	Initialization Fault Code - Mfg	R	R	R	R	R	
673	Get	Start Inhibit Status	-	R	R	R	R	
674	Get	Start Inhibit Status - Mfg	-	R	R	R	R	
675	Get	Start Inhibit Code	-	R	R	R	R	
676	Get	Mfg Start Inhibit Code	-	R	R	R	R	
710	Get	Control Power-up Time	-	O	O	O	O	
711	Get	Cumulative Run Time	-	O	O	O	O	
712	Get	Cumulative Energy Usage	-	O	O	O	O	
713	Get	Cumulative Motor Revs	-	O	O	O	O	
714	Get	Cumulative Main Power Cycles	-	O	O	O	O	
715	Get	Cumulative Control Power Cycles	-	O	O	O	O	
720	Get	Inverter Rated Output Voltage	-	R	R	R	R	
721	Get	Inverter Rated Output Current	-	R	R	R	R	
722	Get	Inverter Rated Output Power	-	R	R	R	R	
723	Get	Converter Rated Output Current	-	O	O	O	O	
724	Get	Converter Rated Output Power	-	O	O	O	O	
730	Get	Digital Inputs	-	O	O	O	O	
731	Set	Digital Outputs	-	O	O	O	O	
732	Get	Analog Input 1	-	O	O	O	O	
733	Get	Analog Input 2	-	O	O	O	O	
734	Set	Analog Output 1	-	O	O	O	O	
735	Set	Analog Output 2	-	O	O	O	O	
750	Set	Local Control	O	O	O	O	O	

^a Attribute ID Offset, o = (n-1)x50

7.2 Class attributes

7.2.1 General

The table of attributes in Table 19 applies to the Motion Axis Object class, which may be referenced as instance 0. These attributes exist even before any Motion Axis Object instances have been created. Since they are not tied to any particular axis instance of the CIP Motion device, the class attributes are generally used to address parametric behavior that applies to

all axis instances, for example the communications node behaviour. As instances of this class are created, they are given consecutive instance numbers starting at 1.

NOTE 1 No attributes associated with this object require Non-Volatile storage, so a “No” is implied for all attributes under the NV column.

NOTE 2 Vendor specific bits, and enumerations provide space for device vendors to provide additional product features.

Key to Table 19: (See 7.1.5 and 7.1.6 for details.)

(R) – Required bit or enumeration – shall be supported in the implementation.

(O) – Optional bit or enumeration – support is left to vendor's discretion.

Set* – Indicates the attribute is normally set by the CIP Motion connection data block and not by a Set service.

IECNORM.COM: Click to view the full PDF of IEC 61800-7-202:2007

Table 19 – Class attributes for the Motion Axis Object

Attr ID	Need in implem	Access rule	NV	Attribute name	Data type	Description of attribute	Semantics of values
1 to 7	These class attributes are optional and are described in IEC 61158-5-2 and IEC 61158-6-2.						
10	Required	Set*		Controller Consumed Connection Data	Struct	This attribute contains the consumed data structure defined by the CIP Motion C-to-D Connection.	See Semantics in 7.2.2.1.
11	Required	Set*		Controller Produced Connection Data	Struct	This attribute contains the produced data structure defined by the CIP Motion D-to-C Connection.	See Semantics in 7.2.2.2.
14	Required	Set*		Node Control	BYTE	Contains bits used to control the behavior of the associated device communications node.	See Semantics in 7.2.2.3.
15	Required	Get		Node Status	BYTE	Contains bits used to indicate the status of the associated device communications node.	See Semantics in 7.2.2.4.
16	Required	Get		Node Fault	USINT	Contains numeric code of active node fault condition. The Node Fault Code is used to provide diagnostic detail to pin-point the source of the fault condition.	See Semantics in 7.2.2.5.
17	Optional	Get		Node Alarm	USINT	Contains numeric code of the current active alarm condition. The Node Alarm Code is used to provide diagnostic detail to pin-point the source of the alarm condition.	See Semantics in 7.2.2.6.
18	Required	Set*		Controller Update Period	UDINT	Represents the period between updates of the controller that is also the period between CIP Motion C-to-D Connection updates.	Nanoseconds
19	Required	Set*		Controller Time Offset	LINT	This element represents the 64-bit value at the beginning of the Controller Update Period that is associated with the Controller Time Stamp value. The Controller Time Offset is the value that is added to the controller's local clock to produce System Time.	Nanoseconds
20	Required	Set*		Controller Time Stamp	LINT	This element represents the 64-bit System Time value at the beginning of the Controller Update Period when the controller's update timer event occurred. It is calculated by the controller as the sum of the controller's local	Nanoseconds (CIP Sync absolute)

Attr ID	Need in implem	Access rule	NV	Attribute name	Data type	Description of attribute	Semantics of values
						clock value when update timer event occurred and the controller's System Time Offset value given by Controller Time Offset. The Controller Time Stamp is therefore directly associated with the command data contained in the connection. The time stamp format is absolute and follows the CIP Sync standard with 0 corresponding to 1970-01-01.	
21	Optional	Set		Controller Update Delay High Limit	USINT	Represents high limit delay threshold for a Controller-to-Device (C-to-D) Connection update. This delay is specified in units of Controller Update Periods. Exceeding this limit results in a Control Update Fault.	Number of Controller Update Periods
22	Optional	Set		Controller Update Delay Low Limit	USINT	Represents high limit delay threshold allowed for a Controller-to-Device (C-to-D) Connection update. This delay is specified in units of Controller Update Periods. Exceeding this limit results in a Control Update Alarm.	Number of Controller Update Periods
28	Optional	Get		Sync Variance	UDINT	Represents the current statistical variation of the System Time Offset value from the mean System Time Offset value.	Nanoseconds
29	Optional	Set		Sync Threshold	UDINT	Determines the threshold for the Observed Variance of System Time below which the Motion Axis Object is considered synchronized. The Group Sync service uses this as a criterion for a successful response.	Nanoseconds Default: device dependent minimum value.
30	Optional	Get		Sync Update Delay	REAL	This value is the product of the number of networks hops the device is away from the grandmaster, as determined by the Time Sync Object attribute, Steps Removed, and the time synchronization interval, as determined by the Time Sync Object attribute, Sync Interval.	Seconds

Attr ID	Need in implem	Access rule	NV	Attribute name	Data type	Description of attribute	Semantics of values
31	Optional	Set		Time Data Set	BYTE	This bit-mapped byte contains flags that determine the usage and format of the controller and device timing information. This value is derived from Time Configuration element sent by the controller in the Controller-to-Device (C-to-D) Connection and becomes the Time Configuration value sent by the device in the Device-to-Controller (D-to-C) Connection.	Bit Field: 0 = Update Period 1 = Time Stamps 2 = Time Diagnostics 3-7 = (Reserved)

7.2.2 Semantics

7.2.2.1 Attribute No. 10 – Controller Consumed Connection Data

This attribute contains the consumed data structure defined by the CIP Motion C-to-D Connection. The content of this dynamic data structure is updated by the controller at the Controller Update Period. This structure, outlined in Figure 45, is dynamically defined by connection header data. See 6.4.3 for a detailed description of the connection data structure.

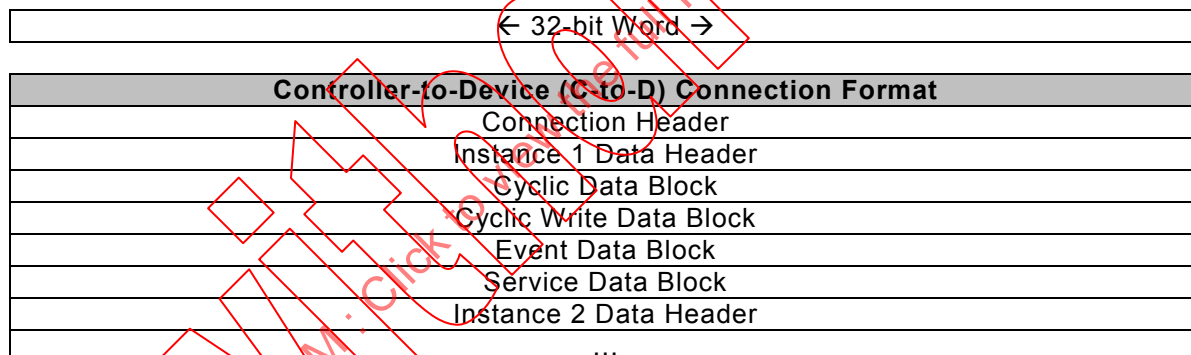


Figure 45 – Controller Consumed Connection Data Format

7.2.2.2 Attribute No. 11 – Controller Produced Connection Data

This attribute contains the produced data structure defined by the CIP Motion D-to-C Connection. This dynamic data structure is updated by the motion device and transmitted to the controller at the Controller Update Period. This structure, outlined in Figure 46, is dynamically defined by connection header data. See 6.4.4 for a detailed description of the connection data structure.

← 32-bit Word →
Device-to-Controller (D-to-C) Connection Format
Connection Header
Instance 1 Data Header
Cyclic Data Block
Cyclic Read Data Block
Event Data Block
Service Data Block
Instance 2 Data Header
...

Figure 46 – Controller Produced Connection Data Format

7.2.2.3 Attribute No. 14 – Node Control

Contains bits used to control the behavior of the associated motion device communications node, as specified in Table 20.

Table 20 – Node Control bit definitions

Bit	Req. Opt.	Name	Description
0	R	Remote Control	This bit is used to request that the device node turn over complete control of the motion device to the controller. If this bit is clear, the CIP Motion device is under the exclusive control of the local interface; the CIP Motion controller cannot affect the behavior of the device in any way other than to switch the device back to Remote Control mode. If the CIP Motion connection is lost while the device is under remote control, the device may then be controlled via the local interface.
1	O	Sync Control	This bit is used by the controller to request synchronous operation of the CIP Motion device. Synchronous operation is defined as having the device node's local timer synchronised with System Time and that the device node start using the Controller Time Stamp and Time Offset to process the connection data and schedule future Device-to-Controller (D-to-C) Connection updates. The Synchronous Control bit implies that the Controller Time Stamp and Time Offset values are valid. The bit shall remain set for the duration of synchronous operation. In asynchronous mode, there is no requirement for the local timer to be synchronized nor is time-stamping necessary or the associated time data even valid.
2	R	Controller Data Valid	This bit shall be set for the device node to process the instance data blocks. During the connection initialization sequence there may be a period of time where the connection data in the instance data blocks is not yet initialized. This condition can be recognized by the device by first checking this bit and finding it clear.
3	R	Node Fault Reset	This bit is used to request that the CIP Motion device perform a reset of the communications node and attempt to clear the Node Fault bit. When the controller detects that the Node Fault bit is clear, the Node Fault Reset bit is cleared. If the communication fault persists, the Node Fault bit may stay set, in which case a power cycle or a complete reconfiguration process shall be performed.
4 to 5			(Reserved)
6 to 7			(Vendor Specific)

7.2.2.4 Attribute No. 15 – Node Status

Contains bits used to indicate the status of the associated CIP Motion device's communications node, as specified in Table 21.

Table 21 – Node Status bit definitions

Bit	Req. Opt.	Name	Description
0	R	Remote Mode	The Remote Mode bit is used to indicate that the device node has turned over total control of the motion device to the controller. If this bit is clear, the device shall not act on any data contained in the Controller-to-Device cyclic or service channels.
1	O	Sync Mode	The Sync Mode bit indicates whether or not the device node is synchronised. Synchronous operation is defined as having the device node's local timer synchronised with System Time and that the device node is using the connection time stamps to process the connection data. The Sync Mode bit being set also implies that the Device Time Stamp is valid. The bit shall remain set for the duration of synchronous operation. If the Sync Mode bit is clear, the device is said to be in Asynchronous mode. In Asynchronous mode, there is no requirement for the local timer to be synchronised nor is time-stamping necessary or even valid.
2	R	Data Valid	The Data Valid bit shall be set for the controller to process the instance data blocks from the device node. During the connection initialization sequence, there may be a period of time where the connection data in the instance data blocks is not yet initialized. This condition can be recognized by the controller by first checking this bit.
3	R	Node Fault	The Node Fault bit is used to indicate that the device has detected one or more fault conditions related to the communications node. Specific fault conditions can be determined by the Node Fault attribute. If this bit is clear, there are no fault conditions present. The Node Fault bit may be cleared by setting the Node Fault Reset bit in the Node Control word. All associated axes are disabled when a Node Fault condition is present.
4 to 5			(Reserved)
6 to 7			(Vendor Specific)

7.2.2.5 Attribute No. 16 – Node Faults

This attribute is an 8-bit fault code used to indicate the presence of specific fault condition associated with the CIP Motion device's communications node, as specified in Table 22.

Table 22 – Node Faults code definitions

Code	Name	Description
0	No Fault	Indicates that there is no fault condition currently present at the device communications node.
1	Control Update Fault	The Control Update Fault bit is used to indicate that updates from the controller have been excessively late.
2	Processor Watchdog Fault	The Processor Watchdog Fault indicates that the processor associated with the device node has experienced an excessive overload condition that has tripped the associated processor watchdog mechanism.
3	Hardware Fault	The Hardware Fault indicates that the critical support hardware (FPGA, ASIC, etc.) associated with the device node has experienced a fault condition.
4	Data Format Error	This fault indicates that an error has occurred in the data format between the controller and the device. A format revision mismatch would be one example which would set this fault.
5 to 127	(Reserved)	
128 to 255	(Vendor Specific)	

7.2.2.6 Attribute No. 17 – Node Alarms

This attribute is an 8-bit alarm code used to indicate specific alarm conditions of the associated CIP Motion device's communications node, as specified in Table 23. Alarms do not result in any direct action.

Table 23 – Node Alarms code definitions

Code	Name	Description
0	No Alarm	Indicates that there is no alarm condition currently present at the device communications node.
1	Control Update Alarm	The Control Update Alarm bit is used to indicate that updates from the controller have been late.
2	Processor Overload Alarm	The Processor Overload Alarm indicates that the processor associated with device is experiencing overload conditions that could eventually lead to a fault.
3	Sync Alarm	Sync Alarm indicates that the Sync Variance has exceeded the Sync Threshold while the device is running in Sync Mode.
4 to 127	(Reserved)	
128 to 255	(Vendor Specific)	

7.3 Instance attributes

7.3.1 General

This subclause lists all the supported attributes of a Motion Axis Object instance. Because of the large number of attributes listed in this subclause, attributes have been organised by functional category. Each functional grouping may be further organised by first listing the object Status and Signal attributes, followed by the object Configuration attributes.

Due to the large number of instance attributes supported by this object, 16-bit Attribute IDs are required. Unless otherwise specified, all optional attributes default to 0.

NOTE No attributes associated with this object require Non-Volatile storage, so a "No" is implied for all attributes under the NV column.

Key to Tables: (See 7.1.5 and 7.1.6 for details.)

(R) – Required bit or enumeration – shall be supported in the implementation

(O) – Optional bit or enumeration – support is left to vendor's discretion

Control Modes:

P – Position Control

V – Velocity Control

T – Torque Control

N – No Control

Control Methods:

F – Frequency Drives (V/Hz or VFD)

C – Closed Loop (Vector Control)

Set* – Indicates the attributes is normally set by the CIP Motion connection header and not by a Set service.

% Device Rated Units – defined as the percentage of the continuous rating of the device with 100 % implying operation at the continuous rated specification for the device. This unit can be applied to attributes related to speed, torque, force, current, voltage, and power. Applicable “Devices” can be Motor, Inverter, Converter, or Bus Regulator. This unit can be used independently of whether the attribute value represents an instantaneous level or a time-averaged level; the appropriate unit for the device rating is implied. As with all attributes that are in units of %, an attribute value of 100 means 100 %.

Feedback Units: Attributes that relate to motion dynamics typically express displacement in terms of the selected feedback device count resolution. Since the CIP Motion device can use different feedback channels for different control loops depending on the feedback configuration, the determination of which feedback device applies depends on the Feedback Configuration.

Table 24 is a cross-reference table to determine the appropriate feedback counts or units to apply to an attribute based on its Dynamic Unit type (position, velocity, or acceleration), and the configured Feedback Configuration. When an axis instance is configured for No Feedback, i.e. sensorless operation (V/Hz, sensorless velocity servo, or future stepper support), feedback counts do not apply, so motion dynamics shall be expressed in terms of motor displacement.

Table 24 – Dynamic Unit vs. Feedback Configuration

Dynamic Units	Feedback Configuration			
	No Feedback	Feedback 1	Feedback 2	Dual Loop Feedback
Position Control Units	Motor Counts	Feedback 1 Counts	Feedback 2 Counts	Feedback 2 Counts
Velocity Control Units	Motor Units	Feedback 1 Units	Feedback 2 Units	Feedback 1 Units
Accel Control Units	Motor Units	Feedback 1 Units	Feedback 2 Units	Feedback 1 Units

7.3.2 Motion Control configuration attributes

7.3.2.1 General

The following attribute table in Table 25 contains basic motion control configuration attributes associated with a Motion Axis Object instance. These attributes govern aspects of the overall behavior of the Motion Axis Object.

Set* ➔ These attributes are generally updated via the cyclic Command Data Set of the CIP Motion Controller-to-Device (C-to-D) Connection. When included as cyclic command data, these attributes should not be updated via a Set service.

Table 25 – Motion Control configuration attributes

Attr ID	Need in implem	Access rule	NV	Attribute name	Data type	Description of attribute	Semantics of values
80	Required - All	Set*		Control Mode	BYTE	Attribute that determines the general dynamic control behavior of the motion axis instance.	See Semantics in 7.3.2.2.1
81	Required - All	Set		Control Method	USINT	Enumerated code that determines the basic motor control algorithm applied by the device to control the dynamic behavior of the motor.	See Semantics in 7.3.2.2.2

7.3.2.2 Semantics

7.3.2.2.1 Attribute No. 80 – Control Mode

The Control Mode attribute consists of a 4-bit enumerated Motor Control field and 4-bits reserved for future expansion, as shown in Figure 47.

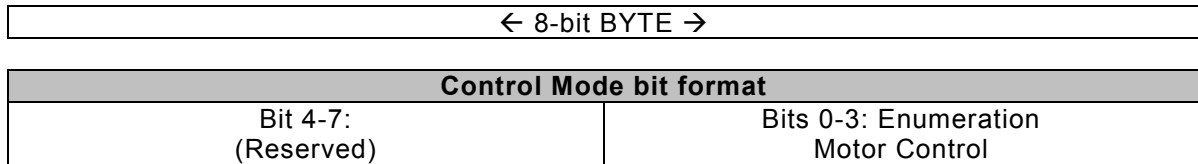


Figure 47 – Control Mode bit field

The Motor Control field enumeration determines the specific dynamic behavior of the motor that the device is to control for this axis instance. The system view of these control modes are described in detail in 4.2. Table 26 shows a brief summary of the Motor Control modes:

Table 26 – Motor Control field enumeration definitions

Enum.	Req. Opt.	Name	Description
0	R/N	No Control	No motor control is provided in this mode but interface to a specific feedback device as a master feedback source is possible via the Feedback Configuration attribute.
1	R/P	Position Control	Device seeks to control the position, or orientation, of the motor.
2	R/PV	Velocity Control	Device seeks to control the velocity of the motor.
3	O	Acceleration Control	Device seeks to control the acceleration of the motor.
4	R/C	Torque Control	Device seeks to control the torque output of the motor.
5	O/C	Current Control	Device seeks to control the torque producing current to the motor.
6 to 15		(Reserved)	-

7.3.2.2.2 Attribute No. 81 – Control Method

The Control Method attribute is an 8-bit enumerated code that determines the basic control algorithm applied by the device to control the dynamic behavior of the motor associated with an axis instance, as specified in Table 27.

Table 27 – Control Method field enumeration definitions

Enum.	Req. Opt.	Name	Description
0	R/N	No Control	No Control is associated with a Control Mode of No Control where there is no explicit motor control provided by the device for this axis instance.
1	R/F	Frequency Control	Frequency Control is an "open loop" control method that applies voltage to the motor, generally in proportion to the commanded frequency or speed. This control method is associated with Variable Frequency Drives (VFDs) or so-called Volts/Hertz drives.
2	R/C	PI Vector Control	PI Vector Control is a "closed loop" control method that uses actual or estimated feedback for closed loop cascaded PI control of motor dynamics, i.e. position, velocity, acceleration, and torque, and always includes independent closed loop PI control of Iq and Id components of the motor current vector.
3 to 255		(Reserved)	-

7.3.3 Motor attributes

7.3.3.1 General

The following attribute tables contain motor configuration attributes associated with a Motion Axis Object instance that apply to various motor technologies. These motor technologies include three-phase motor rotary, linear, permanent magnet and induction motors. Motor attributes are, therefore, organised according to the various motor types. The Need in Implementation category for an attribute is based on the context of the Motor Type and, thus, to the context of the table in which the attribute appears. Where needed, Standard vs. Optional can be further differentiated by abbreviations for PM (Permanent Magnet) and IM (Induction Motors). It is implied that these motor attributes are not required for No Control (Feedback Only) control mode.

The goal of this motor attribute subclause is to define the minimal set of required attributes to support CIP Motion device interchangeability. This guarantees that there is sufficient parametric data provided by the controller for any CIP Motion compliant device, i.e. drive, to effectively control a given motor.

For induction motors, the Motion Axis Object leverages the IEEE recommended phase-neutral equivalent circuit motor model based on “Wye” configuration (see Figure 48). Reactance values, X , are related to their corresponding Inductance values L , by $X = \omega L$, where ω is the rated frequency of the motor. The prime notation, for example X_2' , R_2' , indicates that the actual rotor component values X_2 , and R_2 are referenced to the stator side of the stator-to-rotor winding ratio.

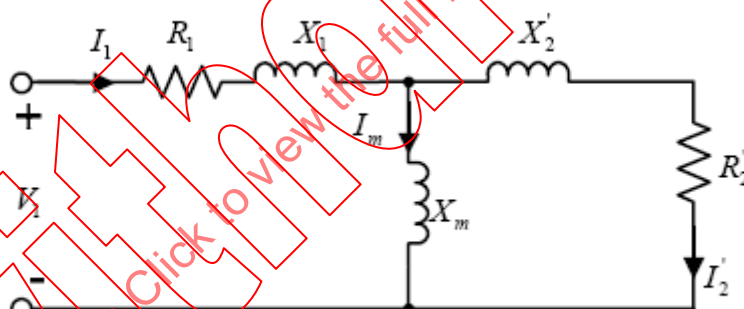


Figure 48 – IEEE per phase motor model

For permanent magnet motors, the Motion Axis Object assumes all motor parameters are defined in the context of a phase-to-phase motor model.

7.3.3.2 General motor attributes

The attribute tables in Table 28 and Table 29 contain general motor attributes that apply to all motor technologies.

Table 28 – General Motor Info Attributes

Attr ID	Need in implem	Access rule	NV	Attribute name	Data type	Description of attribute	Semantics of values
1310	Optional	Get		Motor Catalog Number	SHORT STRING	A 32-character string that specifies the motor catalog number. This attribute is not applicable when the Motor Data Source is Controller NV.	
1311	Optional	Get		Motor Serial Number	SHORT STRING	A 16-character string that specifies the serial number of the motor. This attribute is applicable only when the Motor Data Source is Motor NV.	for example 0012003400560078
1312	Optional	Get		Motor Date Code	SHORT STRING	A 16-character string that specifies the manufacturing date of the motor. This attribute is applicable only when the Motor Data Source is Motor NV.	

Table 29 – General Motor Configuration Attributes

Attr ID	Need in implem	Access rule	NV	Attribute name	Data type	Description of attribute	Semantics of values
1313	Required	Set		Motor Data Source	USINT	An enumeration that specifies the source of motor data for the drive. Controller NV implies that the motor attributes are derived from the controller's non-volatile memory during the drive configuration process. Drive NV implies that the motor attributes are derived directly from the drive's non-volatile memory. In this mode, no motor parameters shall be sent by the controller to the drive. Motor NV implies that the motor attributes are derived from non-volatile memory of a motor-mounted smart feedback device equipped with a serial interface. Again, in this mode, no motor parameters shall be sent by the controller to the drive.	Bits 0-3: Enum 0 = Controller NV (R) 1 = Drive NV (O) 2 = Motor NV (O) 3 to 127 = (reserved) 128 to 255 = (vendor specific)
1314	Required	Set		Motor Device Code	UDINT	A unique number assigned to a motor catalogue number. This value is used to ensure that the motor and motor mounted feedback device configuration data delivered from the controller matches the actual motor and feedback data connected to the drive. This comparison is only valid in the case where the Motor Data Source is Controller NV but the motor is equipped with a smart feedback device. If the codes do not match, a negative acknowledge is given by the drive. Motor Device Codes are assigned by the motor manufacturer.	

Attr ID	Need in implem	Access rule	N V	Attribute name	Data type	Description of attribute	Semantics of values
1315	Required	Set		Motor Type	USINT	An enumeration that specifies the motor type.	Enumeration: 0 = rotary permanent magnet (O) 1 = rotary induction (O) 2 = linear permanent magnet (O) 3 = linear induction (O) 4 to 127 = (reserved) 128 to 255 = (vendor specific)
1316	Required	Set		Motor Unit	UINT	Unit of measure for motor displacement. This is also used for sensorless operation.	Enumeration: 0 = Rev 1 = Metre 2 to 127 = (reserved) 128 to 255 = (vendor specific)
1317	Optional	Set		Motor Polarity	USINT	An enumerated value used to establish the direction of motor motion in response to positive drive output. Normal polarity is defined as the direction of motor travel when the UVW motor leads are hooked up according to the drive's published specifications. Reverse polarity effectively switches the UVW phasing to U'V'W' so that the motor moves in the opposite direction in response to a positive drive output. This attribute can be used to make the direction of travel agree with the user's definition of positive travel and can be used in conjunction with the Feedback Polarity bit to provide negative feedback, when closed loop control is required.	Enumeration: 0 = Normal Polarity 1 = Reverse Polarity
1318	Required	Set		Motor Rated Voltage	REAL	A float that specifies the nameplate AC voltage rating of the motor. This represents the phase-to-phase voltage applied to the motor to reach rated speed at full load.	Volts (RMS)
1319	Required	Set		Motor Rated Continuous Current	REAL	A float that specifies the nameplate AC continuous current rating of the motor. This represents the current applied to the motor under full load conditions at rated speed and voltage.	Amperes (RMS)
1320	Required – PM Optional - IM	Set		Motor Rated Peak Current	REAL	A float that specifies the peak or intermittent current rating of the motor. The peak current rating of the motor is often determined by either the thermal constraints of the stator winding or the saturation limits of PM motor magnetic material.	Amperes (RMS)

Attr ID	Need in implem	Access rule	N V	Attribute name	Data type	Description of attribute	Semantics of values
1321	Required – IM Optional – PM	Set		Motor Rated Output Power	REAL	A float that specifies the nameplate rated output power rating of the motor. This represents the power output of motor under full load conditions at rated current, speed and voltage.	Watts
1322	Required	Set		Motor Overload Limit	REAL	A float that specifies the maximum thermal overload limit for the motor. This value is typically 100 %, corresponding to the power dissipated when operating at the continuous current rating of the motor, but can be significantly higher if cooling options are applied. For induction motors, this attribute is related to the Service Factor of the motor. When the Motor Capacity attribute value associated with the motor thermal model exceeds the Motor Overload Limit, the drive can optionally trigger a predetermined Motor Overload action. The Motor Overload Limit can also be used by the drive to determine the absolute thermal capacity limit of the motor, i.e. the Motor Thermal Overload Factory Limit, that if exceeded, generates a Motor Thermal Overload FL exception.	% Motor Rated
1323	Optional	Set		Motor Integral Thermal Switch	BOOL	A boolean that specifies if the motor has an integral thermal switch.	0 = No 1 = Yes
1324	Optional	Set		Motor Max Winding Temperature	REAL	A float that specifies the maximum winding temperature of the motor.	°C
1325	Optional	Set		Motor Winding-to-Ambient Thermal Capacitance	REAL	A float that specifies the winding-to-ambient thermal capacitance.	J/°C
1326	Optional	Set		Motor Winding-to-Ambient Thermal Resistance	REAL	A float that specifies the winding-to-ambient thermal resistance.	°C/W

7.3.3.3 General PM motor attributes

The attribute table in Table 30 contains motor configuration attributes that apply to Permanent Magnet motor types in general.

Table 30 – General PM Motor Configuration Attributes

Attr ID	Need in implem	Access rule	NV	Attribute name	Data type	Description of attribute	Semantics of values
1327	Required	Set		PM Motor Resistance	REAL	A float that specifies the phase-to-phase, resistance of a permanent magnet motor.	Ohms
1328	Required	Set		PM Motor Inductance	REAL	A float that specifies the phase-to-phase, inductance of a permanent magnet motor.	Henries

7.3.3.4 General rotary motor attributes

The attribute table in Table 31 contains motor configuration attributes that apply specifically to rotary motor types.

Table 31 – General Rotary Motor Configuration Attributes

Attr ID	Need in implem	Access rule	NV	Attribute name	Data type	Description of attribute	Semantics of values
1329	Required	Set		Rotary Motor Poles	UINT	An integer that specifies the number of poles per revolution for rotary motors. This value is always an even number, as poles always exist in pairs.	
1330	Optional	Set		Rotary Motor Inertia	REAL	A float that specifies the unloaded inertia of a rotary motor.	kg.m ²
1331	Required	Set		Rotary Motor Rated Speed	REAL	A float that specifies the nameplate rated speed of a rotary motor at rated voltage and rated current. For induction motors, this value is often referred to as base speed. The value also generally represents the speed at which the output power of the motor is maximized.	RPM
1332	Optional	Set		Rotary Motor Max Speed	REAL	A float that specifies the absolute maximum speed of a rotary motor in units of RPM. This speed may be determined by the limitations of the motor or by limitations of the mechanical system. Specifically, this value can represent the maximum safe operating speed, maximum continuous "no-load" speed, maximum continuous encoder speed, or maximum continuous bearing speed of the motor. This value can be used to determine the Motor Overspeed exception.	RPM
1333	Optional	Set		Rotary Motor Damping Coefficient	REAL	A float that specifies the damping, or viscous friction, associated with a rotary motor.	N.m/(rad/s)

7.3.3.5 General linear motor attributes

The attribute table in Table 32 contains motor configuration attributes that apply specifically to linear motor types.

Table 32 – General Linear Motor Configuration Attributes

Attr ID	Need in implem	Access rule	N V	Attribute name	Data type	Description of attribute	Semantics of values
1334	Required	Set		Linear Motor Pole Pitch	REAL	A float that specifies the pole pitch of a linear motor in units of metres, and is equivalent to the electrical cycle length.	metres
1335	Required	Set		Linear Motor Rated Speed	REAL	A float that specifies the nameplate rated speed of a linear motor at rated voltage and rated current. For induction motors, this value is often referred to as base speed. The value also generally represents the speed at which the rated output power of the motor is measured.	m/s
1336	Optional	Set		Linear Motor Moving Mass	REAL	A float that specifies the unloaded moving mass of a linear motor.	kg
1337	Optional	Set		Linear Motor Max Speed	REAL	A float that specifies the absolute maximum speed of a linear motor in units of m/s. This speed may be determined by the limitations of the motor or by limitations of the mechanical system. Specifically, this value can represent the maximum safe operating speed, maximum continuous "no-load" speed, maximum continuous encoder speed, or maximum continuous bearing speed of the motor. This value can be used to determine the Motor Overspeed exception.	m/s
1338	Optional	Set		Linear Motor Damping Coefficient	REAL	A float that specifies the damping, or viscous friction, associated with a linear motor in units.	N/(m/s)

7.3.3.6 Rotary PM motor attributes

The attribute table in Table 33 contains motor configuration attributes that apply specifically to rotary motor types.

Table 33 – Rotary PM Motor Configuration Attributes

Attr ID	Need in implem	Access rule	N V	Attribute name	Data type	Description of attribute	Semantics of values
1339	Required	Set		PM Motor Rated Torque	REAL	A float that specifies the nameplate continuous torque rating of a rotary permanent magnet motor.	N.m
1340	Required	Set		PM Motor Torque Constant	REAL	A float that specifies the torque constant of a rotary permanent magnet motor in newton metres per RMS ampere.	N.m/ A (RMS)
1341	Required	Set		PM Motor Rotary Voltage Constant	REAL	A float that specifies the voltage, or back-EMF, constant of a rotary permanent magnet motor in phase-to-phase RMS Volts per kRPM.	Volts (RMS)/kRPM

7.3.3.7 Linear PM motor attributes

The attribute table in Table 34 contains motor configuration attributes that apply specifically to linear PM motor types.

Table 34 – Linear PM Motor Configuration Attributes

Attr ID	Need in implem	Access rule	N V	Attribute name	Data type	Description of attribute	Semantics of values
1342	Required	Set		Motor Rated Force	REAL	A float that specifies the nameplate continuous force rating of a linear permanent magnet motor.	N
1343	Required	Set		Motor Force Constant	REAL	A float that specifies the force constant of a linear permanent magnet motor in Newtons per RMS ampere.	N/A (RMS)
1344	Required	Set		Motor Linear Voltage Constant	REAL	A float that specifies the voltage, or back-EMF, constant of a linear permanent magnet motor in phase-to-phase RMS Volts per m/s.	Volts (RMS) / (m/s)

7.3.3.8 Induction motor attributes

The attribute table in Table 35 contains motor configuration attributes that apply specifically to induction motor types.

Table 35 – Induction Motor Configuration Attributes

Attr ID	Need in implem	Access rule	N V	Attribute name	Data type	Description of attribute	Semantics of values
1345	Required	Set		Induction Motor Rated Frequency	REAL	A float that specifies the nameplate frequency rating of an induction motor.	Hertz
1346	Required	Set		Induction Motor Flux Current	REAL	Id Current Reference that required to generate full motor flux. This value is closely approximated by the No Load Motor Rated Current commonly found in Induction Motor data sheets.	Amperes (RMS)
1347	Required	Set		Induction Motor Stator Resistance	REAL	A float that specifies the Y circuit, phase-neutral, winding resistance of the stator as shown as R1 in the IEEE motor model.	Ohms
1348	Required	Set		Induction Motor Stator Leakage Reactance	REAL	A float that specifies the Y circuit, phase-neutral, leakage reactance of the stator winding, at rated frequency, as shown as X ₁ in the IEEE motor model.	Ohms
1349	Required ^a	Set		Induction Motor Magnetization Reactance	REAL	A float that specifies the Y circuit, phase-neutral, magnetizing reactance of the motor, at rated frequency, as shown as X _m in the IEEE motor model.	Ohms
1350	Required ^a	Set		Induction Motor Rotor Resistance	REAL	A float that specifies the phase-neutral equivalent stator-referenced winding resistance of the rotor as shown as R ₂ ' in the IEEE motor model.	Ohms
1351	Required	Set		Induction Motor Rotor Leakage Reactance	REAL	A float that specifies the Y circuit, phase-neutral, equivalent stator-referenced leakage inductance of the rotor winding, at rated frequency, as shown as X ₂ ' in the IEEE motor model.	Ohms

^a These parameters have a strong motor temperature component that some drive devices circumvent through various adaption or compensation techniques. Nevertheless, these parameters are classified as required for the purposes of device interchangeability.

7.3.4 Feedback attributes

7.3.4.1 General

The attribute tables in Table 36 and Table 37 contain all position feedback related attributes associated with a Motion Axis Object instance that apply to various feedback device and feedback interface technologies.

NOTE These feedback interface technologies include Digital AqB (digital A quad B signals), Sine/Cosine (analog A quad B signals), Digital Parallel (parallel digital bit interface), SSI (Synchronous Serial Interface), LDT (Linear Displacement Transducer) and Resolver. Other modern feedback interfaces supported are Hiperface® (by Stegmann) and EnDat 2.1® & EnDat 2.2® (by Heidenhain).

The “Need in Implementation” category for an attribute is based on the context of the Feedback Type abbreviations for TT (Digital AqB), TP (Digital Parallel), SC (Sine/Cosine), HI (Hiperface®), E21 (EnDat 2.1®), E22 (EnDat 2.2®), SS (SSI), LT (LDT – Linear Displacement Transducer, and RS (Resolver). The abbreviation HS applies to all devices that support Hall Sensors.

The goal of this feedback attribute subclause is to define the minimal set of required attributes to support CIP Motion device interchangeability. This guarantees that there is sufficient parametric data provided by the controller for any CIP Motion compliant drive to effectively interface to a wide range of feedback device types.

Multiple feedback device interfaces are currently defined by the Motion Axis Object for control loop feedback, namely Feedback 1 and Feedback 2. In dual feedback control configurations, Feedback 1 is generally associated with the motor mounted feedback device while Feedback 2 is associated with the load-side or machine mounted feedback device. For all other control modes, Feedback 1 shall be used for the singular feedback interface and is always required for PM Motor commutation.

When No Control is selected for a Motion Axis Object instance, any uncommitted feedback channel can be used as a master feedback source. Since Feedback 1 is typically assigned to the control function of the primary Motion Axis Object instance, Feedback 2, 3, 4, etc., are likely candidates as master feedback sources. For this version of the Motion Axis Object, two additional feedback channels, Feedback 3 and Feedback 4, are supported for this purpose.

To minimize the length of the feedback attribute tables below, the letter “n” in the generic “Feedback n” attribute name is used to specify the associated feedback channel number. Valid channel numbers for open standard feedback attributes of the Motion Axis Object are 1, 2, 3, and 4. Attribute IDs are assigned based on the channel number. Support for feedback interface channels, 2, 3, 4 is optional in the device implementation. However, if hardware support for any of these feedback channels is available in a given device, these optional attributes are clearly required in the implementation for the channels to be available for use.

Table 36 – General Feedback Info Attributes

Attr ID	Need in implem	Access rule	N V	Attribute name	Data type	Description of attribute	Semantics of values
1400 + (n-1)x50	Optional - All	Get		Feedback n Catalog Number	SHORT STRING	A 32-character string that specifies the catalog number of the device associated with Feedback n.	
1401 + (n-1)x50	Optional - All	Get		Feedback n Serial Number	SHORT STRING	A 16-character string that specifies the serial number of the device associated with Feedback n.	for example 001200340 0560078

Attr ID	Need in implem	Access rule	N V	Attribute name	Data type	Description of attribute	Semantics of values
1402 + (n-1)x50	Required - NPVC Optional - T	Get		Feedback n Position	DINT	Actual position of the axis based on Feedback n.	Feedback n Counts
1403 + (n-1)x50	Required - NPVC Optional - T	Get		Feedback n Velocity	REAL	Actual filtered velocity of the axis based on Feedback n.	Feedback n Units/s
1404 + (n-1)x50	Required - NPVC Optional - T	Get		Feedback n Acceleration	REAL	Actual filtered acceleration of the axis based on Feedback n.	Feedback n Units/s ²

Table 37 – Feedback Configuration Attributes

Attr ID	Need in implem	Access rule	N V	Attribute name	Data type	Description of attribute	Semantics of values
90	Required - PVT	Set*		Feedback Data Set	BYTE	Bit mapped field that determines what actual feedback data values are being transmitted to the controller in the Device-to-Controller (D-to-C) Connection.	Bit Field: 0 = Position Feedback (R/C) 1 = Velocity Feedback (R/C) 2 = Accel Feedback (O) 3 = Torque Reference (R/T) 4 = Iq Current Reference (O) 5 = Voltage Output (O) 6 = Frequency Output (O) 7 = (Reserved)
82	Required - All	Set*		Feedback Configuration	BYTE	This attribute determines how the various available feedback channels are used to implement the selected Control Mode.	See Semantics in 7.3.4.2
83	Required - All	Set*		Feedback Master Select	USINT	This attribute determines what Logical channel is assigned to this axis instance when the Feedback Configuration is set to Master Feedback.	Enumeration: 0 = (Reserved) 1 = Feedback 1 2 = Feedback 2 3 = Feedback 3 4 = Feedback 4
84	Optional - PC	Set		Feedback 1/2 Count Ratio	REAL	Number of Feedback 1 Counts per Feedback 2 Counts. This value is used to convert between feedback 2 counts to feedback 1 counts when configured for dual position loop operation.	Feedback 1 Counts per Feedback 2 Count
1411 + (n-1)x50	Optional - NC	Set		Feedback n Unit	USINT	Unit of measure for the designated feedback device. The Feedback Unit for Feedback 1 shall be scalable to the configured Motor Unit; if the Motor Unit is set to Rev, Feedback 1 Unit shall be set to Rev; if Motor Unit is set to Metre, Feedback 1 Unit shall be set to either Metre or Inch and the device shall handle conversion between the linear motor and feedback displacement	Enumeration: 0 = Rev 1 = Metre 2 = Inch 3 to 255 = (reserved)

Attr ID	Need in implem	Access rule	N V	Attribute name	Data type	Description of attribute	Semantics of values
						units.	
1412 + (n-1)x50	Optional	Set		Feedback n Port Select	REAL	Maps the logical Feedback Channel “n” to a physical Feedback Port “m”. Default Feedback n Port Select = n.	Enumeration: 0 = (Reserved) 1 = Port 1 2 = Port 2 3 = Port 3 4 = Port 4 5 to 255 = Reserved
1413 + (n-1)x50	Required	Set		Feedback n Type	USINT	Identifies the type of feedback device connected to the associated Feedback interface. Device support for any individual feedback types is left to the discretion of the device manufacturer. However if a specific feedback type is supported, attributes associated with that type are generally required in the implementation.	Enumeration: 0 = No Feedback Present 1 = Digital AqB (w/o Hall) 2 = Digital AqB (w/ Hall) 3 = Digital Parallel (Absolute) 4 = Sine/Cosine (w/o Hall) 5 = Sine/Cosine (w/ Hall) 6 = Hiperface ® 7 = EnDat 2.1 ® 8 = EnDat 2.2 ® 9 = Resolver 10 = SSI 11 = LDT 12 to 127 = (reserved) 128 to 255 = (vendor specific)
1414 + (n-1)x50	Optional	Set		Feedback n Polarity	USINT	An enumerated value used to establish the direction of change in the feedback counter in response to positive motion of the associated feedback device. Normal polarity is defined as that which results in increasing feedback counts when the feedback device is hooked up and moved in the positive direction according to the devices published specifications. Reverse polarity internally switches the polarity of the feedback accumulator so that the feedback counts decrease when the feedback device moves in the positive direction. This attribute can be used to make the direction of travel agree with the user's definition of positive travel and can be used in conjunction with the Motor Polarity bit to provide negative feedback, when this feedback channel is used for closed loop control.	Enumeration: 0 = Normal Polarity 1 = Reverse Polarity
1415 + (n-1)x50	Required	Set		Feedback n Mode	USINT	Determine how the device applies the feedback count value. When configured for Incremental mode, the device zeroes the feedback count accumulator at power-	Enumeration: 0 = Incremental 1 = Absolute

Attr ID	Need in implem	Access rule	N V	Attribute name	Data type	Description of attribute	Semantics of values
						up. When configured for Absolute mode, the device initializes the feedback count accumulator at power-up to the absolute feedback position value read from the feedback device.	
1416 + (n-1)x50	Required - TT, SC, HI, E21, RS	Set		Feedback n Cycle Resolution	UDINT	Number of feedback cycles per Feedback Unit. Cycles for a Digital AqB device is the "line" resolution of the encoder, while it represents the sinusoidal "cycle" resolution of a Sin/Cos feedback device and the "pole" count of Resolver feedback device. For digital serial (e.g. SSI) or parallel absolute feedback device interfaces, this value would represent the "step" resolution of the device.	Feedback Cycles/Feedback Unit
1 417 + (n-1)x50	Required - TT, SC, HI, E21, RS	Set		Feedback n Cycle Interpolation	UDINT	Number of interpolated Feedback Counts per Feedback Cycle. For a Digital AqB device the device's feedback interface hardware can generally support interpolation values of 1, 2, or 4. For a Sin/Cos, Hiperface, Endat 2.1, or Resolver feedback device, the number is generally much larger and determined by the interpolation capability of the device feedback interface hardware. A value of 1 024 is typical in this case. For digital serial (e.g. SSI) or parallel absolute feedback device interfaces, this value is always 1 since there is no opportunity of device-based interpolation. The effective resolution of the feedback device in Feedback Counts per Feedback Unit is the product of the Feedback Cycle Resolution and the Feedback Cycle Interpolation.	Feedback Counts/Feedback Cycle
1420 + (n-1)x50	Required - TP, SS, HI, E21, E22	Set		Feedback n Data Length	USINT	Number of feedback data bits transferred over the digital serial or parallel data interface channel of a feedback	Number of Bits

Attr ID	Need in implem	Access rule	N V	Attribute name	Data type	Description of attribute	Semantics of values
						device.	
1421 + (n-1)x50	Optional – TP, SS	Set		Feedback n Data Code	USINT	Type of feedback data bit encoding used by designated serial or parallel data interface channel of a feedback device.	Enumeration: 0 = Binary 1 = Gray
1422 + (n-1)x50	Optional - RS	Set		Feedback n Resolver Transformer Ratio	REAL	Transformer Ratio specification of the designated resolver feedback device.	
1423 + (n-1)x50	Optional - RS	Set		Feedback n Resolver Excitation Voltage	REAL	Sets the sinusoidal excitation voltage applied to the rotor of the designated resolver feedback device.	Volts (RMS)
1424 + (n-1)x50	Optional - RS	Set		Feedback n Resolver Excitation Frequency	REAL	Frequency of sinusoidal excitation signal applied to the designated resolver feedback device. Valid frequency range or values for this attribute depends on the specific device hardware interface.	Hertz
1425 + (n-1)x50	Optional - RS	Set		Feedback n Resolver Cable Balance	REAL	Adjusts the relative amplitude of the Sine and Cosine signals from the resolver to compensate for impact of resolver cable.	%
1426 + (n-1)x50	Required - LT	Set		Feedback n LDT Type	USINT	Determines the LDT type. Options are Start/Stop and PWM. Start/Stop transducers accept an input (interrogate) signal to start the measurement cycle and respond with two pulses on the Return line. Timing can be based on either the Rising or Falling edge. The time between the pulses is proportional to the position. PWM transducers respond to the interrogate signal with a single long pulse on the Return line. The pulse width is proportional to the position.	Enumeration: 0 = PWM 1 = Start/Stop Rising 2 = Start/Stop Falling
1427 + (n-1)x50	Required - LT	Set		Feedback n LDT Recirculations	USINT	Determines the number of recirculations associated with a PWM type LDT transducer. Multiple recirculations can be used to increase the resolution of the LDT at the expense of increasing the sample period.	Number Recirculations
1428 + (n-1)x50	Required – HS, TP, HI, E21, E22, SS,	Set		Feedback n Commutation Offset	REAL	A value that specifies the commutation offset of the PM motor mounted feedback device in units of	Electrical Degrees

Attr ID	Need in implem	Access rule	N V	Attribute name	Data type	Description of attribute	Semantics of values
	RS (PM Motors Only)					electrical degrees. This attribute specifies the offset from a commutation reference position defined by applying DC current into the U terminal and out of the shorted V and W terminals of the motor and allowing the rotor to move to its magnetic null position relative to the stator. On an absolute encoder or resolver, the offset is the difference from the device's zero absolute position and the commutation reference position. On an incremental encoder with Hall sensors, the offset is the difference between the position corresponding to a transition of the Hall S3 channel (with the S1 channel high and the S2 channel low) and the commutation reference position. The commutation offset is only applicable to the motor mounted Feedback 1 device.	
1434 + (n-1)x50	Optional	Set		Feedback n Velocity Filter Bandwidth	REAL	Controls the bandwidth of the Low Pass Filter applied to the raw velocity signal from Feedback n. A value of 0 for this attribute disables this feature.	rad/s
1435 + (n-1)x50	Optional	Set		Feedback n Accel Filter Bandwidth	REAL	Controls the bandwidth of the Low Pass Filter applied to the raw acceleration signal from Feedback n. A value of 0 for this attribute disables this feature.	rad/s

7.3.4.2 Semantics: Attribute No. 82 – Feedback Configuration

This attribute contains a 4-bit enumerated Feedback Selection field that determines how the various logical feedback channels are used to implement the selected Control Mode for this axis instance, as shown in Figure 49.

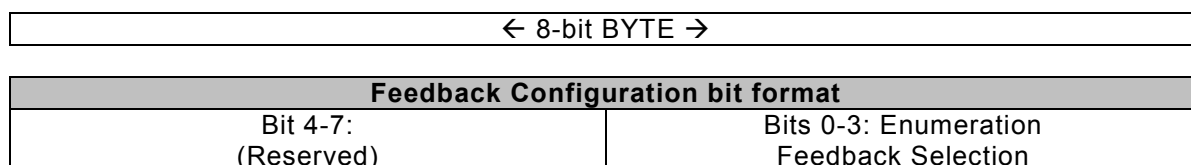


Figure 49 – Feedback Configuration bit field

Feedback Selection enumerations (see Table 38) provide support for multi-feedback control functionality for the various active Control Modes, i.e. where the device is actively controlling

the motor based on feedback. In these active Control Modes, it is assumed that logical channel, Feedback 1, is attached directly to the motor while Feedback 2 is attached to the load side of the mechanical transmission. Commutation signals for a PM motor are always derived from the Feedback 1.

Table 38 – Feedback Selection field enumeration definitions

Enum.	Req. Opt.	Name	Description
0	R/NF O/C	No Feedback	No Feedback is selected when sensorless open loop or closed loop control is desired. When performing open loop control, no feedback signal is required. In closed loop control, the required feedback signal is estimated by a sensorless control algorithm based on motor phase voltage and current signals.
1	R/N	Master Feedback	Master Feedback assigns an uncommitted feedback channel, as specified by the Feedback Master Select attribute, to this motion axis instance to serve as a "master feedback" source when the device is configured for No Control mode.
2	R/C	Motor Feedback	When Motor Feedback is selected, then commutation, acceleration, velocity, and position feedback signals are all derived from motor mounted Feedback 1.
3	O/C	Load Feedback	When Load Feedback is selected, then motor-mounted Feedback 1 is only used for PM motor commutation while load-side Feedback 2 is used for position, velocity, and acceleration.
4	O/P	Dual Feedback	When Dual Feedback is selected, then motor mounted Feedback 1 is used for commutation, acceleration, and velocity, and load-side Feedback 2 is used strictly for position.
5 to 255	-	(Reserved)	-

7.3.5 Event Capture attributes

7.3.5.1 General

The attribute table in Table 39 contains all event related attributes associated with a Motion Axis Object instance. These include registration, marker, and homing events. The Event Capture attributes are designed to support the possibility of up to 16 active events per controller update period. The format of all Time Stamp attributes is absolute System Time and follows the CIP Sync specification with 0 corresponding to 1970-01-01.

The Motion Axis Object currently supports two independent registration input channels per axis instance that can be triggered on either the rising or falling edges of the signal. If the device hardware implementation allows, event time and position data can be captured for all four event conditions simultaneously. The Event Capture attributes also support Auto-rearm for registration events. This allows for controller implementation of important features like Windowed Registration and Registration Pattern Recognition.

The Motion Axis Object also supports Home Switch, Marker and Switch-Marker events for homing functionality on a per axis basis. The Marker events are typically generated by the configured position feedback device for the associated axis instance.

Set* ➔ These attributes are generally updated via the cyclic Command Data Set of the CIP Motion Controller-to-Device (C-to-D) connection. When included as cyclic command data, these attributes should not be updated via a Set service.

Table 39 – Event Attributes

Attr ID	Need in implem	Access rule	N V	Attribute name	Data type	Description of attribute	Semantics of values
256	Optional – NP	Set*		Event Checking Control	DWORD	This attribute is passed to the device by the controller as part of the Device-to-Controller (D-to-C) Connection for the purpose of arming various device inputs, for example marker, home switch, and registration inputs, to generate events to the controller. When these enabled events occur, the device captures both the time and exact axis position when the event occurred.	See Semantics in 7.3.5.2.1
257	Optional – NP	Get		Event Checking Status	DWORD	This attribute is passed to the device by the controller as part of the Device-to-Controller (D-to-C) Connection to indicate if the device is currently checking for events based on various device inputs, for example marker, home, and registration inputs. Event checking is initiated when the corresponding Event Checking Control bit is set in the Controller-to-Device (C-to-D) Connection.	See Semantics in 7.3.5.2.2
258	Required – NP	Get		Registration 1 Positive Edge Position	DINT	Feedback position latched on the rising edge of the Registration Input 1.	Position Control Units
259	Required – NP	Get		Registration 1 Negative Edge Position	DINT	Feedback Position latched on the falling edge of the Registration Input 1.	Position Control Units
260	Optional – NP	Get		Registration 2 Positive Edge Position	DINT	Feedback position latched on the rising edge of the Registration Input 2.	Position Control Units
261	Optional – NP	Get		Registration 2 Negative Edge Position	DINT	Feedback Position latched on the falling edge of the Registration Input 2.	Position Control Units
262	Optional – NP	Get		Registration 1 Positive Edge Time	LINT	System Time stamp on the rising edge of the Registration Input 1.	Nanoseconds (CIP Sync absolute)
263	Required – NP	Get		Registration 1 Negative Edge Time	LINT	System Time stamp on the falling edge of the Registration Input 1.	Nanoseconds (CIP Sync absolute)
264	Required – NP	Get		Registration 2 Positive Edge Time	LINT	System Time stamp on the rising edge of the Registration Input 2.	Nanoseconds (CIP Sync absolute)
265	Optional – NP	Get		Registration 2 Negative Edge Time	LINT	System Time stamp on the falling edge of the Registration Input 2.	Nanoseconds (CIP Sync absolute)
266	Required – NP	Get		Home Event Position	DINT	Feedback Position latched on the specified home event.	Position Control Units
268	Optional – NP	Get		Home Event Time	LINT	System Time stamp latched on the specified home event.	Nanoseconds (CIP Sync absolute)

7.3.5.2 Semantics

7.3.5.2.1 Attribute No. 256 – Event Checking Control

The first 28-bits of this 32-bit attribute (see Figure 50) are used to enable various device inputs, for example marker, home switch, and registration inputs, to generate events to the controller. When these enabled events occur, the device captures both the time and exact position of the associated motion axis instance. The most significant 4-bits represent the number of Event Acknowledgement messages in the Event Data Block in the last Controller-to-Device (C-to-D) Connection update. For a detailed description of the event handling protocol between the drive and the controller, refer to the Connection section in the appropriate CIP Motion device profile (see 6.4.3.4).

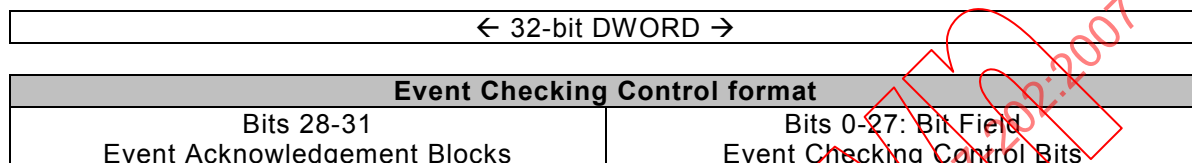


Figure 50 – Event Checking Control word field

For the Home Switch-Marker events specified in Table 40, the device first looks for level of home switch input to transition according to the first + or – symbol and then immediately look for the transition of the marker input according to the second + or – symbol.

Table 40 – Event Checking Control bit definitions

Bit	Req. Opt.	Name	Description
0	R/P	Reg 1 Pos Edge	Enable checking for positive edge transition of Registration 1 Input
1	R/P	Reg 1 Neg Edge	Enable checking for negative edge transition of Registration 1 Input
2	O/P	Reg 2 Pos Edge	Enable checking for positive edge transition of Registration 2 Input
3	O/P	Reg 2 Neg Edge	Enable checking for negative edge transition of Registration 2 Input
4-7		(Reserved)	-
8	O/P	Reg 1 Pos Auto-rearm	Enable auto-rearm mechanism after positive edge transition of Registration 1 Input
9	O/P	Reg 1 Neg Auto-rearm	Enable auto-rearm mechanism after negative edge transition of Registration 1 Input
10	O/P	Reg 2 Pos Auto-rearm	Enable auto-rearm mechanism after positive edge transition of Registration 2 Input
11	O/P	Reg 2 Neg Auto-rearm	Enable auto-rearm mechanism after negative edge transition of Registration 2 Input
12 to 15		(Reserved)	
16	R/P	Marker Pos Edge	Enable checking for positive edge of Marker input of associated position feedback channel.
17	R/P	Marker Neg Edge	Enable checking for negative edge of Marker input of associated position feedback channel.
18	R/P	Home Switch Pos Edge	Enable checking for positive edge of Home input associated with this axis instance.
19	R/P	Home Switch Neg Edge	Enable checking for positive edge of Home input associated with this axis instance.
20	R/P	Home Switch-Marker ++	Enable checking for event sequence specified as a positive edge transition of the Home Switch input followed by a positive edge of the Marker input.
21	R/P	Home Switch-Marker +-	Enable checking for event sequence specified as a positive edge transition of the Home Switch input

Bit	Req. Opt.	Name	Description
			followed by a negative edge of the Marker input.
22	R/P	Home Switch-Marker -+	Enable checking for event sequence specified as a negative edge transition of the Home Switch input followed by a positive edge of the Marker input.
23	R/P	Home Switch-Marker --	Enable checking for event sequence specified as a negative edge transition of the Home Switch input followed by a negative edge of the Marker input.
24 to 27		(Reserved)	

7.3.5.2.2 Attribute No. 257 – Event Checking Status

The first 28-bits of this 32-bit attribute (see Figure 51) are used to indicate if the device is currently checking for events based on various device inputs, for example marker and registration inputs (see Table 41). Event checking is initiated when the corresponding event checking control bit is set in the Controller-to-Device (C-to-D) Connection. When an event occurs, the device captures both the time and exact axis position and passes this information to the controller in event notification data blocks. But for the controller to process the event data, the corresponding event checking status bit shall be set. The most significant 4-bits of this attribute represent the number of Event Notification messages, are in the Event Data Block in the last Device-to-Controller (D-to-C) Connection update. For a detailed description of the event handling protocol between the device and the controller, refer to the Connection section (see 6.4.4.4).

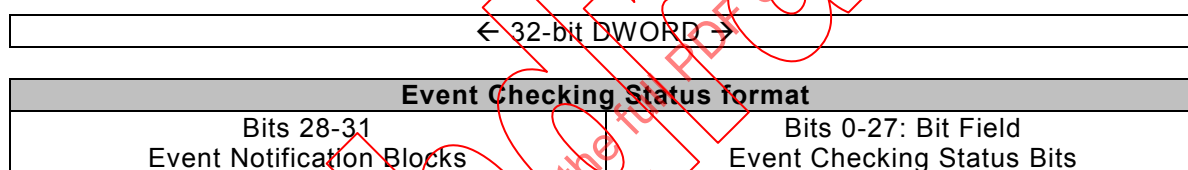


Figure 51 – Event Checking Status word field

Table 41 – Event Checking Status bit definitions

Bit	Req. Opt.	Name	Description
0	R/P	Reg 1 Pos Edge	Checking for positive edge transition of Registration 1 Input?
1	R/P	Reg 1 Neg Edge	Checking for negative edge transition of Registration 1 Input?
2	O/P	Reg 2 Pos Edge	Checking for positive edge transition of Registration 2 Input?
3	O/P	Reg 2 Neg Edge	Checking for negative edge transition of Registration 2 Input?
4 to 7		(Reserved)	-
8	O/P	Reg 1 Pos Auto-rearm	Auto-rearm mechanism after positive edge transition of Registration 1 Input active?
9	O/P	Reg 1 Neg Auto-rearm	Auto-rearm mechanism after negative edge transition of Registration 1 Input active?
10	O/P	Reg 2 Pos Auto-rearm	Auto-rearm mechanism after positive edge transition of Registration 2 Input active?
11	O/P	Reg 2 Neg Auto-rearm	Auto-rearm mechanism after negative edge transition of Registration 2 Input active?
12 to 15		(Reserved)	
16	R/P	Marker Pos Edge	Checking for positive edge of Marker input of associated position feedback channel?
17	R/P	Marker Neg Edge	Checking for negative edge of Marker input of associated position feedback channel?

Bit	Req. Opt.	Name	Description
18	R/P	Home Switch Pos Edge	Checking for positive edge of Home input associated with this axis instance?
19	R/P	Home Switch Neg Edge	Checking for positive edge of Home input associated with this axis instance?
20	R/P	Home Switch-Marker ++	Checking for event sequence specified as a positive edge transition of the Home Switch input followed by a positive edge of the Marker input?
21	R/P	Home Switch-Marker +-	Checking for event sequence specified as a positive edge transition of the Home Switch input followed by a negative edge of the Marker input?
22	R/P	Home Switch-Marker -+	Checking for event sequence specified as a negative edge transition of the Home Switch input followed by a positive edge of the Marker input?
23	R/P	Home Switch-Marker --	Checking for event sequence specified as a negative edge transition of the Home Switch input followed by a negative edge of the Marker input?
24 to 27		(Reserved)	

7.3.6 Command reference generation attributes

7.3.6.1 General

The lists of attributes in Table 42 and Table 43 apply to the command reference generation functionality of the device that converts command position, velocity, acceleration, and torque data output from a controller-based planner into corresponding command references signals to the device's motor control structures. The command reference generator functionality includes fine interpolators, signal selector switches, dynamic limiters.

Set* → These attributes are generally updated via the cyclic Command Data Set of the CIP Motion Controller-to-Device (C-to-D) connection. When included as cyclic command data, these attributes should not be updated via a Set service.

Table 42 – Command Generator Signal Attributes

Attr ID	Need in implem	Access rule	N V	Attribute name	Data type	Description of attribute	Semantics of values
93	Required – P Optional – VT	Get		Command Target Time	LINT	Time stamp associated with command data from the motion planner and used by the command fine interpolators. The Command Target Time is the sum of the Controller Time Stamp value and the Command Target Update interval associated with this axis instance. The time stamp format is absolute and follows the CIP Sync standard with 0 corresponding to 1970-01-01.	Nano-seconds (CIP Sync absolute)
286	Required – P	Set*		Controller Position Command	DINT	Command position data value from controller, updated at the Controller Update Period, and feeding into a fine interpolator.	Position Control Units
287	Required – V Optional – P	Set*		Controller Velocity Command	REAL	Command velocity data value from controller, updated at the Controller Update Period, and feeding into a fine interpolator.	Velocity Control Units/s
288	Optional – PVT	Set*		Controller Acceleration Command	REAL	Command acceleration data value from controller, updated at the Controller Update Period, and feeding into a fine interpolator.	Accel Control Units /s ²

Attr ID	Need in implem	Access rule	N V	Attribute name	Data type	Description of attribute	Semantics of values
289	Optional - PVT	Set*		Controller Torque Command	REAL	Commanded torque data value from controller, updated at the Controller Update Period, and feeding into a fine interpolator.	% Motor Rated
290	Optional - P	Get		Interpolated Command Position	DINT	Output value from the Command Position fine interpolator.	Position Control Units
291	Optional - PV	Get		Interpolated Command Velocity	REAL	Output value from the Command Velocity fine interpolator. When no Command Velocity signal is present when performing position control, this signal can be derived by scaling the Differential Position output value of the Command Position fine interpolator.	Velocity Control Units/s
292	Optional - PVT	Get		Interpolated Command Acceleration	REAL	Output value from the Command Acceleration fine interpolator. When no Command Acceleration signal is present when performing position or velocity control, this signal can be derived by scaling the Differential Velocity output value of the Command Velocity fine interpolator. If no Command Velocity signal is present, the Interpolated Command Acceleration signal can be derived by scaling the 2nd Differential Position output value of the Command Position fine interpolator.	Accel Control Units/s ²

Table 43 – Command Generator Configuration Attributes

Attr ID	Need in implem	Access rule	N V	Attribute name	Data type	Description of attribute	Semantics of values
91	Required – PVT	Set*		Command Data Set	BYTE	Bit mapped field that determines what command data values are being updated by the controller.	Bit Field: 0 = Position Command (R/P) 1 = Velocity Command (R/V) 2 = Accel Command (O) 3 = Torque Command (R/T) 4 = Position Trim (O) 5 = Acceleration Trim (O) 6 = Velocity Trim (O) 7 = Torque Trim (O)
92	Required – PV	Set*		Interpolation Control	BYTE	Controls the interpolation/extrapolation method applied to position, velocity, and acceleration command data from the motion planner based on the associated time stamp.	See Semantics in 7.3.6.2
302	Required – V Optional - P	Set		Velocity Limit - Positive	REAL	This value defines the maximum allowable positive velocity command into the velocity summing junction. If this velocity limit value is exceeded, the device responds by clamping the velocity command to this limit and setting the Velocity Limit	Velocity Control Units/s

Attr ID	Need in implem	Access rule	N V	Attribute name	Data type	Description of attribute	Semantics of values
						status bit.	
303	Required – V Optional – P	Set		Velocity Limit - Negative	REAL	This value defines the maximum allowable negative velocity command into the velocity summing junction. If this velocity limit value is exceeded, the device responds by clamping the velocity command to this limit and setting the Velocity Limit status bit.	Velocity Control Units/s
304	Required – V Optional – P	Set		Acceleration Limit	REAL	This value defines the maximum allowable positive rate of change in the magnitude of the velocity command (increasing speed). If this acceleration limit value is exceeded, the device responds by clamping the rate of change of the velocity command to this limit and setting the Acceleration Limit status bit.	Velocity Control Units/s ²
305	Required – V Optional – P	Set		Deceleration Limit	REAL	This value defines the maximum allowable negative rate of change in the magnitude of the velocity command (decreasing speed). If this deceleration limit value is exceeded, the device responds by clamping the rate of change of the velocity command to this limit and setting the Deceleration Limit status bit.	Velocity Control Units/s ²
306	Optional – PV	Set		Jerk Limit Control	REAL	When accel or decel limited, this value sets the percentage of accel or decel time that is applied to the speed ramp as jerk limited S Curve. The S Curve time is added half at the beginning and half at the end of the ramp. A value of 0 results in no S-Curve, i.e. a linear acceleration ramp. A value of 100 % results in a triangular acceleration profile with the peak being the acceleration/deceleration limit. As the Jerk Limit Control value increases, the derived Jerk limit value decreases.	%
307	Optional – PV	Set		Flying Start Enable	BOOL	This Boolean value is used to enable or disable the Flying Start feature of the device. When Flying Start Enabled is true and the motion axis is enabled, the device determines the current velocity of the motor as part of the Starting State initialization activities. Just prior to transitioning to the Running state, the device presets the output of the Accel Limiter to the current velocity. In this way, the motor seamlessly ramps from its current velocity to the commanded velocity profile generated by the motion planner. When Flying Start Enabled is false, the motor velocity is not determined and a preset of 0 is applied to the Accel Limiter output.	False = Flying Start Disabled True = Flying Start Enabled

Attr ID	Need in implem	Access rule	N V	Attribute name	Data type	Description of attribute	Semantics of values
308	Optional - F	Set		Skip Speed 1	REAL	Sets the central speed of a skip speed band within which the device does not operate.	Velocity Control Units/s
309	Optional - F	Set		Skip Speed 2	REAL	Sets the central speed of a skip speed band within which the device does not operate.	Velocity Control Units/s
310	Optional - F	Set		Skip Speed 3	REAL	Sets the central speed of a skip speed band within which the device does not operate.	Velocity Control Units/s
311	Optional - F	Set		Skip Speed Band	REAL	Determines the speed window around a skip speed that cannot be commanded. Any command set-point within this window is adjusted by the Skip Speed block to fall at either the upper or lower Skip Speed Band boundary value. The axis can smoothly accelerate or decelerate through the skip speed band based on the acceleration limiter block but may not operate at a set speed within the band. The Skip Speed Band is distributed half above and half below the skip speed. This Skip Speed Band attribute applies to all skip speeds supported in the device. A value of 0 for this attribute disables this feature.	Velocity Control Units/s

7.3.6.2 Semantics: Attribute No. 92 – Interpolation Control

The Interpolation Control attribute governs the interpolation/extrapolation method applied to position, velocity, and acceleration command data from the motion planner based on the associated time stamp (see Figure 52). The first two bits of this 8-bit attribute represent the Command Target Update that determines the number of Update Periods (UP) added to the command data Time Stamp to determine the absolute System Time that the command data value is targeted for, i.e. the Command Target Time (see Table 44). For more details on interpolation and extrapolation control, please refer to 7.6.6.2 (Command Fine Interpolation).

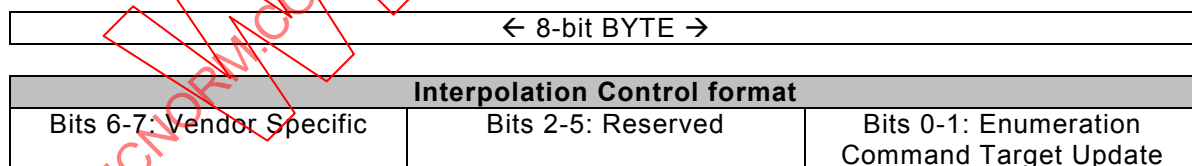


Figure 52 – Interpolation Control word field

Table 44 – Command Target Update enumeration definition

Enum.	Req. Opt.	Name	Description
0	R/P	Immediate	A Command Target Update of 0 implies that the command data shall be applied to the control structure immediately.
1	R/P	Extrapolate (+1 UP)	A Command Target Update of 1 implies that extrapolation shall be used to apply the command data to the control structure based on adding 1 Update Period to the command Time Stamp.
2	R/P	Interpolate (+2 UP)	A Command Target Update of 2 implies that fine interpolation shall be used to apply the command data based on adding 2 Update Periods to the command Time Stamp.
3		(Reserved)	-

7.3.7 Control mode attributes

7.3.7.1 Position Loop attributes

The attributes tables in Table 45 and Table 46 contain all position control related attributes associated with a Motion Axis Object instance.

Set* → this attribute is generally updated via the cyclic Command Data Set of the CIP motion Controller-to-Device (C-to-D) Connection. When included as cyclic command data, this attribute should not be updated via a Set service.

Table 45 – Position Loop Signal Attributes

Attr ID	Need in implem	Access rule	N V	Attribute name	Data type	Description of attribute	Semantics of values
420	Required – P	Get		Position Command	DINT	Command position output from the fine interpolator (if active) into position loop when configured for position loop control.	Position Control Units
421	Required – P	Set*		Position Trim	DINT	Additional position command added to the Position Command to generate the Position Reference signal into the position loop summing junction.	Position Control Units
422	Required – P	Get		Position Reference	DINT	Command position reference signal into the position loop summing junction to be compared with a position feedback signal.	Position Control Units
423	Required – PC	Get		Velocity Feedforward Command	REAL	Velocity feedforward reference into velocity loop summing junction.	Velocity Control Units/s
424	Required – NPVC Optional – T	Get		Position Feedback	DINT	Position feedback value channeled into the position control summing junction.	Position Control Units
426	Required – PC	Get		Position Error	REAL	Error between commanded and actual position that is the output of the position loop summing junction.	Position Control Units
427	Required – PC	Get		Position Integrator Output	REAL	Output of position integrator representing the contribution of the position integrator to Position Loop Output.	Velocity Control Units/s
428	Required – PC	Get		Position Loop Output	REAL	Output of the position loop forward path representing the total control effort of the position loop.	Velocity Control Units/s

Table 46 – Position Loop Configuration Attributes

Attr ID	Need in implem	Access rule	N V	Attribute name	Data type	Description of attribute	Semantics of values
430	Required – PC	Set		Kvff	REAL	Velocity Feedforward Gain value that multiplies the Velocity Feedforward Command signal before applying it to the velocity loop summing junction.	%
431	Required – PC	Set		Kpp	REAL	Position Proportional Gain value that multiplies the Position Error signal before applying it to the velocity loop summing junction.	rad/s
432	Required – PC	Set		Kpi	REAL	Position Integral Gain value that multiplies the Position Integrator Error signal before applying it to the velocity loop summing junction.	rad/s ²
433	Required – PC	Set		Position Lock Tolerance	REAL	This value establishes a window around the current command position. When the actual position is within this window the Position Lock status bit is set. When actual position falls outside this window, the Position Lock status bit is cleared.	Position Control Units
434	Required – PC	Set		Position Error Tolerance	REAL	Determines the absolute maximum Position Error value that can be tolerated without causing an Excessive Position Error exception.	Position Control Units
435	Optional – PVC	Set		Position Error Tolerance Time	REAL	Determines the maximum amount of time that the Position Error Tolerance can be exceeded without generating an exception.	Seconds
439	Required – PC	Set		Position Integrator Control	BYTE	This bit field controls the behavior of the position loop integrator while commanding motion via the controller. When the integrator hold enable bit is set, the integrator is held while motion is being commanded with a non-zero velocity. When clear, the integrator runs without qualification. When the auto-preset bit is set, the integrator preload value is automatically loaded with the current velocity command when there is a control mode change between velocity control and position control. If clear, the integrator is loaded with the configured position integrator preload value.	Bit Field: 0: Integrator Hold Enable 1: Auto-preset
440	Required – PC	Set		Position Integrator Preload	REAL	Value assigned to the position integrator when the position control loop is enabled.	Velocity Control Units/s

7.3.7.2 Velocity Loop attributes

The attributes tables in Table 47 and Table 48 contain all velocity control related attributes associated with a Motion Axis Object instance.

Set* → this attribute is generally updated via the cyclic Command Data Set of the CIP motion Controller-to-Device (C-to-D) Connection. When included as cyclic command data, this attribute should not be updated via a Set service.

Table 47 – Velocity Loop Signal Attributes

Attr ID	Need in implem	Access rule	N V	Attribute name	Data type	Description of attribute	Semantics of values
450	Required – PV	Get		Velocity Command	REAL	Command velocity output from fine interpolator (if active) into velocity loop when configured for velocity loop control.	Velocity Control Units/s
451	Required – PV	Set*		Velocity Trim	REAL	Additional velocity command added to the velocity loop summing junction.	Velocity Control Units/s
452	Required – PVC	Get		Acceleration Feedforward Command	REAL	Acceleration feedforward reference into acceleration loop summing junction.	Velocity Control Units/s ²
453	Required – PV	Get		Velocity Reference	REAL	Command velocity reference into velocity loop summing junction.	Velocity Control Units/s
454	Required – NPVC Optional – T	Get		Velocity Feedback	REAL	Actual velocity of the axis applied to the velocity summing junction based on Control Mode selection.	Velocity Control Units/s
455	Required – PVC	Get		Velocity Error	REAL	Error between the velocity reference and velocity feedback value that is the output of the velocity loop summing junction.	Velocity Control Units/s
456	Required – PVC	Get		Velocity Integrator Output	REAL	Output of velocity integrator representing the contribution of the velocity integrator to Velocity Loop Output.	Accel Control Units/s ²
457	Required – PVC	Get		Velocity Loop Output	REAL	Output of velocity forward path representing the total control effort of the velocity loop.	Accel Control Units/s ²

Table 48 – Velocity Loop Configuration Attributes

Attr ID	Need in implem	Access rule	N V	Attribute name	Data type	Description of attribute	Semantics of values
460	Required – PC	Set		Kaff	REAL	Acceleration Feedforward Gain value that multiplies the Acceleration Feedforward Command signal before applying it to the velocity loop summing junction.	%
461	Required – PVC	Set		Kvp	REAL	Velocity Proportional Gain value that multiplies the Velocity Error signal before applying it to the acceleration summing junction.	rad/s
462	Required – PVC	Set		Kvi	REAL	Velocity Integral Gain value that multiplies the Velocity Integrator Error signal before applying it to the acceleration summing junction.	rad/s ²
464	Required – V Optional – P	Set		Kdr	REAL	Velocity Droop value that provides compliance to the velocity integrator by subtracting a portion of the velocity loop effort from the	Velocity Control Units/ % Rated

Attr ID	Need in implem	Access rule	N V	Attribute name	Data type	Description of attribute	Semantics of values
						velocity error input to the velocity integrator. The presence of the Torque/Force scaling gain, K _j , in the droop signal path allows Velocity Droop to be specified in velocity units per % rated torque output. This parameter is also valid for V/Hz drives and its behavior is nearly identical, but instead of % rated being related to torque, % rated is related to current.	
465	Optional – PVC	Set		Velocity Error Tolerance	REAL	Determines the absolute maximum Velocity Error value that can be tolerated without causing an Excessive Velocity Error exception.	Velocity Control Units/s
466	Optional – PVC	Set		Velocity Error Tolerance Time	REAL	Determines the maximum amount of time that the Velocity Error Tolerance can be exceeded without generating an exception.	seconds
467	Required – PVC	Set		Velocity Integrator Control	BYTE	This bit field controls the behavior of the velocity loop integrator while commanding motion via the controller. When the integrator hold enable bit is set, the integrator is held while motion is being commanded with a non-zero velocity. When clear, the integrator runs without qualification. When the auto-preset bit is set, the integrator preload value is automatically loaded with the current torque command when there is a control mode change between torque control and velocity control. If clear, the integrator is loaded with the configured velocity integrator preload value.	Bit Field: 0: Integrator Hold Enable 1: Auto-preset
468	Required – PVC	Set		Velocity Integrator Preload	REAL	Value assigned to the velocity integrator when the velocity control loop is enabled.	Accel Control Units/s ²
469	Optional – PVC	Set		Velocity Low Pass Filter Bandwidth	REAL	Controls the bandwidth of the Low Pass Filter applied to the Velocity Error signal. Recommended implementation is a two pole IIR filter. A value of 0 for this attribute disables this feature.	rad/s
470	Optional – PV	Set		Velocity Threshold	REAL	Defines a minimum absolute velocity, below which, results in the Velocity Threshold status bit being set.	Velocity Control Units/s
471	Optional – PV	Set		Velocity Lock Tolerance	REAL	This value establishes a window around the current command velocity. When the actual velocity is within this window the Velocity Lock status bit is set. When actual position falls outside this window, the Velocity Lock status bit is cleared.	Velocity Control Units/s

Attr ID	Need in implem	Access rule	N V	Attribute name	Data type	Description of attribute	Semantics of values
472	Required – PV	Set		Velocity Standstill Window	REAL	This value establishes a window around zero speed. When the actual velocity is within this window, the Velocity Standstill status bit is set. When actual speed falls outside this window, the Velocity Standstill status bit is cleared.	Velocity Control Units/s

7.3.7.3 Acceleration Control attributes

The attributes table in Table 49 contains all accelerations related attributes associated with a Motion Axis Object instance.

Set* → this attribute is generally updated via the cyclic Command Data Set of the CIP motion Controller-to-Device (C-to-D) Connection. When included as cyclic command data, this attribute should not be updated via a Set service.

Table 49 – Acceleration Signal Attributes

Attr ID	Need in implem	Access rule	N V	Attribute name	Data type	Description of attribute	Semantics of values
480	Optional - C	Get		Acceleration Command	REAL	Command acceleration output from fine interpolator (if active) into acceleration loop when configured for acceleration control.	Accel Control Units/s ²
481	Optional - C	Set*		Acceleration Trim	REAL	Additional acceleration command added to the acceleration loop summing junction.	Accel Control Units/s ²
482	Optional - C	Get		Acceleration Reference	REAL	Command acceleration reference into acceleration loop summing junction.	Accel Control Units/s ²
483	Required – NPVC Optional – T	Get		Acceleration Feedback	REAL	Actual acceleration of the axis based on the selected feedback device.	Accel Control Units/s ²

7.3.7.4 Torque/Force Reference attributes

The attributes tables in Table 50 and Table 51 contain all torque/force related attributes associated with a Motion Axis Object instance.

Set* → this attribute is generally updated via the cyclic Command Data Set of the CIP motion Controller-to-Device (C-to-D) Connection. When included as cyclic command data, this attribute should not be updated via a Set service.

Table 50 – Torque/Force Reference Signal Attributes

Attr ID	Need in implem	Access rule	N V	Attribute name	Data type	Description of attribute	Semantics of values
490	Required – C	Get		Torque Command	REAL	Command torque output from fine interpolator (if active) into torque input summing junction when configured for torque control.	% Motor Rated
491	Required – C	Set*		Torque Trim	REAL	Additional torque command added to the torque input	% Motor Rated

Attr ID	Need in implem	Access rule	N V	Attribute name	Data type	Description of attribute	Semantics of values
						summing junction.	
492	Required – C	Get		Torque Reference	REAL	Commanded torque reference input signal before torque filter section representing the sum of the Torque Command and Torque Trim signal inputs.	% Motor Rated
493	Required – C	Get		Filtered Torque Reference	REAL	Commanded torque reference input signal after torque filter section.	% Motor Rated
494	Required – C	Get		Limited Torque Reference	REAL	Commanded torque reference input signal after torque limiter section.	% Motor Rated

Table 51 – Torque/Force Reference Configuration Attributes

Attr ID	Need in implem	Access rule	N V	Attribute name	Data type	Description of attribute	Semantics of values
496	Required – PVC	Set		Kj	REAL	Torque/Force Scaling Gain value that converts commanded acceleration into equivalent rated torque/force. Properly set, this value represents the total system inertia.	% Motor Rated/(Acc el Units/s ²)
498	Optional – C	Set		Friction Compensation	REAL	Value added to the current/torque command to offset the effects of coulomb friction.	% Motor Rated
502	Optional – C	Set		Torque LP Filter Bandwidth	REAL	Break frequency for the 2nd order low pass filter applied to torque reference signal.	rad/s
503	Optional – C	Set		Torque Cmd Notch Freq	REAL	Center frequency of the notch filter applied to the torque reference signal. A value of 0 for this attribute disables this feature.	rad/s
504	Required – C	Set		Torque Limit - Positive	REAL	Determines the maximum positive torque that can be applied to the motor. If the device attempts to exceed this value, the torque command is clamped to this value.	% Motor Rated
505	Required – C	Set		Torque Limit - Negative	REAL	Determines the maximum negative torque that can be applied to the motor. If the device attempts to exceed this value, the torque command is clamped to this value.	% Motor Rated
506	Optional - C	Set		Torque Rate Limit	REAL	Limits the rate of change of the torque reference signal.	% Motor Rated/s
507	Optional – C	Set		Torque Threshold	REAL	Defines a maximum absolute torque level that, when exceeded, results in the Torque Threshold status bit being set.	% Motor Rated
508	Optional – C	Set		Overtorque Limit	REAL	Maximum limit for the Torque Reference signal that when exceeded for the duration specified by Overtorque Limit Time attribute, results in an Overtorque Limit exception. This feature allows the device to generate an exception if there is a sudden increase in load torque during operation. This condition could occur if a bearing fails or	% Motor Rated

Attr ID	Need in implem	Access rule	N V	Attribute name	Data type	Description of attribute	Semantics of values
						there is some other mechanical failure.	
509	Optional – C	Set		Overtorque Limit Time	REAL	Specifies the amount of time allowed in an Overtorque Limit condition before generating an Overtorque Limit exception. A value of 0 for this attribute disables the Undertorque feature.	seconds
510	Optional – C	Set		Undertorque Limit	REAL	Minimum limit for the Torque Reference signal that when exceeded for the duration specified by Undertorque Limit Time attribute, results in an Undertorque Limit exception. This feature allows the device to generate an exception if there is a sudden decrease in load torque during operation. This condition could occur if a load coupling breaks, tensioned web material breaks, etc.	% Motor Rated
511	Optional – C	Set		Undertorque Limit Time	REAL	Specifies the amount of time allowed in an Undertorque Limit condition before generating an Undertorque Limit exception. A value of 0 for this attribute disables the Undertorque feature.	seconds

7.3.7.5 Current Loop attributes

The attributes tables in Table 52 and Table 53 contain all current loop related attributes associated with a Motion Axis Object instance.

Table 52 – Current Loop Signal Attributes

Attr ID	Need in implem	Access rule	N V	Attribute name	Data type	Description of attribute	Semantics of values
520	Required – C	Get		Iq Current Command	REAL	Represents the instantaneous value of the commanded torque producing current signal, Iq, prior to passing through the current limiter.	% Motor Rated
521	Optional – C	Get		Operative Current Limit	REAL	Represents the operative current limit based on multiple limit sources.	% Motor Rated
522	Optional – C	Get		Current Limit Source	USINT	Represents the operative source of a torque limit should a current limit condition occur.	Enumeration: 0 = Not Limited 1 = Inverter Peak Current Limit 2 = Motor Peak Current Limit 3 = Inverter Thermal Current Limit 4 = Motor Thermal Current Limit 5 = Shunt Regulator Limit 6 to 127 = reserved 128 to 255 = vendor specific
523	Required – C	Get		Motor Electrical	REAL	Calculated electrical angle of the motor based on motor	Degrees

Attr ID	Need in implem	Access rule	N V	Attribute name	Data type	Description of attribute	Semantics of values
				Angle		pole count, commutation offset, and selected feedback drive device.	
524	Optional – C	Get		Iq Current Reference	REAL	Current reference signal, Iq, into the torque current loop summing junction.	% Motor Rated
525	Optional – C	Get		Id Current Reference	REAL	Command current reference, Id, into the flux current loop summing junction.	% Motor Rated
527	Optional – C	Get		Iq Current Error	REAL	Error between commanded and actual current that is the output of the torque producing, q-axis, current loop summing junction.	% Motor Rated
528	Optional – C	Get		Id Current Error	REAL	Error between commanded and actual current that is the output of the flux producing, d-axis, current loop summing junction.	% Motor Rated
529	Optional – C	Get		Iq Current Feedback	REAL	Actual torque current applied to the axis based on current sensor feedback.	% Motor Rated
530	Optional – C	Get		Id Current Feedback	REAL	Actual flux current applied to the axis based on current sensor feedback.	% Motor Rated
531	Optional – C	Get		Iq Decoupling	REAL	Signal added to the Iq control loop output to compensate for the effects of Id.	% Motor Rated
532	Optional – C	Get		Id Decoupling	REAL	Signal added to the Id control loop output to compensate for the effects of Iq.	% Motor Rated
533	Optional – C	Get		Vq Voltage Output	REAL	Instantaneous Torque/Force producing output voltage.	% Motor Rated
534	Optional – C	Get		Vd Voltage Output	REAL	Instantaneous Flux producing output voltage.	% Motor Rated
535	Optional – C	Get		U Voltage Output	REAL	Instantaneous voltage applied to U phase.	Volts
536	Optional – C	Get		V Voltage Output	REAL	Instantaneous voltage applied to V phase.	Volts
537	Optional – C	Get		W Voltage Output	REAL	Instantaneous voltage applied to W phase.	Volts
538	Optional – C	Get		U Current Feedback	REAL	Instantaneous current measured on U phase.	Amperes
539	Optional – C	Get		V Current Feedback	REAL	Instantaneous current measured on V phase.	Amperes
540	Optional – C	Get		W Current Feedback	REAL	Instantaneous current measured on W phase.	Amperes
541	Optional – C	Get		U Current Offset	REAL	Offset for U Phase current transducer.	Amperes
542	Optional – C	Get		V Current Offset	REAL	Offset for V Phase current transducer.	Amperes
543	Optional – C	Get		W Current Offset	REAL	Offset for W Phase current transducer.	Amperes

Table 53 – Current Loop Configuration Attributes

Attr ID	Need in implem	Access rule	N V	Attribute name	Data type	Description of attribute	Semantics of values
554	Optional – C	Get		Kqp	REAL	Iq Proportional Gain value that multiplies the Iq Current Error signal before applying it to the Iq decoupling summing junction.	rad/s
555	Optional – C	Get		Kqi	REAL	Iq Integral Gain value that multiplies the Iq Current Error signal before applying it to the Iq Integrator Error accumulator.	rad/s ²
556	Optional – C	Get		Kdp	REAL	Id Proportional Gain value that multiplies the Id Current Error signal before applying it to the Id decoupling summing junction.	rad/s
557	Optional – C	Get		Kdi	REAL	Id Integral Gain value that multiplies the Id Current Error signal before applying it to the Id Integrator Error accumulator.	rad/ s ²
558	Optional – IM	Set		Flux Up Control	USINT	When the motion axis is enabled, DC current is applied to an induction motor to build stator flux before transitioning to the Running state. This attribute controls how an induction motor is to be fluxed in the Starting state prior to transitioning to the Running state. If No Delay is selected, the axis transitions immediately to the Running state while the motor flux is building. With Manual Delay, the axis remains in the Starting state for the Flux Up Time to allow time for the motor to be fully fluxed. With Automatic Delay, the drive device determines the amount of time to delay to fully flux the motor based on motor configuration attribute data or measurements.	Enumeration: 0 = No Delay 1 = Manual Delay 2 = Automatic Delay
559	Optional – IM	Set		Flux Up Time	REAL	Sets the amount of time the drive device allows to build full motor flux before transitioning to the Running state.	seconds

7.3.7.6 Frequency Control Attributes

The attributes tables in Table 54 and Table 55 contain all related attributes associated with the Frequency Control method of operation of a Motion Axis Object instance.

Table 54 – Frequency Control Signal Attributes

Attr ID	Need in implem	Access rule	N V	Attribute name	Data type	Description of attribute	Semantics of values
570	Required - F	Get		Slip Compensation	REAL	Indicates the actual amount of slip compensation currently being applied.	RPM

Table 55 – Frequency Control Configuration Attributes

Attr ID	Need in implem	Access rule	N V	Attribute name	Data type	Description of attribute	Semantics of values
575	Required - F	Set		Frequency Control Method	BYTE	The Basic Volts/Hertz control method applies voltage to the motor generally in direct proportion to the commanded frequency or speed.	Enumeration 0 = basic volts/hertz (R/F) 1 to 127 = reserved 128 to 255 = vendor specific
571	Required - F	Set		Rated Slip Speed	REAL	Represents the amount of slip at motor rated current (full load) and motor rated frequency.	RPM
572	Required - F	Set		Maximum Voltage	REAL	Sets the highest phase-to-phase voltage the drive device can output.	Volts (RMS)
573	Required - F	Set		Maximum Frequency	REAL	Sets the highest frequency the drive device can output.	Hertz
574	Required - F	Set		Frequency Limit	REAL	Sets the frequency limit which is generally set above the Velocity Limit, translated to frequency, to allow for slip compensation.	Hertz
575	Required - F	Set		Break Voltage	REAL	Sets the phase-to-phase output voltage of the drive device at the Break Frequency where boost ends.	Volts (RMS)
576	Required - F	Set		Break Frequency	REAL	Sets the output frequency of the drive device at the Break Voltage where boost ends.	Hertz
577	Required - F	Set		Start Boost	REAL	Sets phase-to-phase voltage boost level for starting and accelerating.	Volts (RMS)
578	Required - F	Set		Run Boost	REAL	Sets the phase-to-phase voltage boost level for steady-state speed or deceleration.	Volts (RMS)

7.3.7.7 Drive Output attributes

The attributes table in Table 56 contains all inverter output related attributes associated with a Motion Axis Object instance.

Table 56 – Drive Output Attributes

Attr ID	Need in implem	Access rule	N V	Attribute name	Data type	Description of attribute	Semantics of values
600	Required - PVT	Get		Output Frequency	REAL	Time averaged output frequency applied to motor.	Hertz
601	Required - PVT	Get		Output Current	REAL	Total time averaged output current applied to motor.	Amperes (RMS)
602	Required - PVT	Get		Output Voltage	REAL	Total time averaged phase-to-phase output voltage applied to motor.	Volts (RMS)
603	Required - PVT	Get		Output Power	REAL	Total time averaged output power of the motor. This value is based on the product of the Torque Reference signal and the Velocity Feedback.	kW
604	Optional -	Set		PWM	UINT	Sets the carrier frequency for the Pulse Width Modulation output to	Hertz

Attr ID	Need in implem	Access rule	N V	Attribute name	Data type	Description of attribute	Semantics of values
	PVT			Frequency		the motor. Drive derating is required at higher PWM frequencies due to switching losses. Current loop update time is tied directly to the PWM frequency so loop performance generally increases with increasing PWM rate.	

7.3.8 Stopping & Braking attributes

7.3.8.1 General

The attributes table in Table 57 contains all active stopping and braking related attributes associated with a Motion Axis Object instance.

Table 57 – Stopping/Braking Attributes

Attr ID	Need in implem	Access rule	N V	Attribute name	Data type	Description of attribute	Semantics of values
610	Required - PVT	Set		Stopping Mode	USINT	When disabling or aborting an axis, this value determines the stopping method to apply to the motor. The final state after the stopping method is applied is the Stopped state. In the Stopped state, the device's inverter power structure should either be Disabled (Disable selection) and free of torque or actively held (Hold selection) in a static condition via servo action.	See Semantics in 7.3.8.2.1
611	Required - PVT	Set		Stopping Current	REAL	When disabling or aborting an axis, this value determines the maximum amount of torque producing current available to stop the motor when the Stopping Mode is set to Current Decel. If this attribute is not supported, the drive device shall use the configured Positive and Negative Peak Current Limits.	% Motor Rated
612	Required - PVT	Set		Stopping Time Limit	REAL	When disabling or aborting an axis, this parameter determines the maximum amount of time Stopping Current can be applied to bring the motor to a successful stop before the axis is disabled and all current is removed from the motor.	seconds
613	Optional - PVT(PM Motors Only)	Set		Resistive Brake Contact Delay	REAL	When an external resistive brake is used, the Resistive Brake Contact Delay can be set to delay the enabling of the drive power structure until after the resistive brake has had time to connect the motor to the drive.	seconds See Semantics in 7.3.8.2.2 for details.
614	Optional - PVT	Set		Mechanical Brake Control	USINT	This attribute allows direct control of the drive output connected to the mechanical brake mechanism. When set to release or engage the brake, the brake is released or engaged without regard to the current state of the axis. An axis state transition can however override the Mechanical Brake Control setting. For example, if this	Enumeration: 0 = Brake Release 1 = Brake Engage

Attr ID	Need in implem	Access rule	N V	Attribute name	Data type	Description of attribute	Semantics of values
						attribute is a 0 and the brake is released and then the axis transitions the Stopped state to the Running state and back to the Stopped state, the brake automatically engages as part of the transition to the stopped state. The device indicates this condition by changing the Mechanical Brake Control setting to 1, indicating that the brake is engaged.	
615	Required - PVT	Set		Mechanical Brake Release Delay	REAL	When enabling the axis, the Mechanical Brake Release Delay value determines the amount of time the drive shall delay transition from the Starting state to the Running or Testing states. This delay prevents any commanded motion of the motion axis until the external mechanical brake has had enough time to disengage.	seconds See Semantics in 7.3.8.2.3 for details.
616	Required - PVT	Set		Mechanical Brake Engage Delay	REAL	When disabling the axis, the Mechanical Brake Engage Delay value determines the amount of time the inverter power structure shall remain enabled after the axis has decelerated to standstill. This allows time for an external mechanical brake to engage. The configured Stopping Mode determines the type of stopping sequence applied.	seconds See Semantics in 7.3.8.2.4 for details.

7.3.8.2 Semantics

7.3.8.2.1 Attribute No. 610 – Stopping Mode

When disabling or aborting an axis, this attribute determines the stopping method to apply to the motor (see Table 58). The final state after the stopping method is applied is the Stopped state. In the Stopped state the axis' inverter power structure should either be Disabled (Disable selection) and free of torque or actively held (Hold selection) in a static condition via servo action. This attribute has no impact or relationship to the planner generated acceleration and deceleration profiles.

Table 58 – Stopping Mode enumeration definitions

Enum.	Req. Opt.	Name	Description
0	R/CF	Disabled & Coast	Disable & Coast immediately disables the inverter power structure which causes the motor to coast unless some form of external braking is applied. This is equivalent to an IEC 60204-1 Category 0 Stop.
1	R/C	Current Decel & Disable	Current Decel & Disable forces the drive to stop under control of the configured Stopping Current. This is accomplished by forcing the relevant upstream control loops to saturate by setting the appropriate command references to fixed values. In the position mode, the position command is reset to actual position when the axis reaches zero velocity. Once stopped, or the configured Stopping Time limit expires, the device disables the power structure and control loops. This stop mode complies with the IEC 60204-1 Category 1 Stop.
2	R/V	Ramped Decel & Disable	Ramped Decel & Disable applies only to the velocity control mode. This stop mode immediately zeroes the output of the Velocity Command Selector causing the Accel Limiter to ramp the motor speed down to zero. Once stopped, or the configured Stopping Time limit expires, the device disables the power

Enum.	Req. Opt.	Name	Description
			structure and control loops. This stop mode also complies with the IEC 60204-1 Category 1 Stop.
3	O/C	Current Decel & Hold	Current Decel & Hold behaves like Current Decel & Disable, but leaves the power structure and control loops active under local command reference control with holding torque. In this way, there is active holding torque to maintain the stopped condition. This stop mode complies with the IEC 60204-1 Category 2 Stop.
4	O/F	Ramped Decel & Hold	Ramped Decel & Hold behaves like Ramped Decel & Disable, but leaves the power structure and control loops active under local command reference control. In this way, there is active holding torque to maintain the stopped condition. This stop mode also complies with the IEC 60204-1 Category 2 Stop.
5 to 127		(Reserved)	-
128 to 255		(Vendor Specific)	-

7.3.8.2.2 Attribute No. 613 – Resistive Brake Contact Delay

When an external resistive brake is used, an external contactor switches the UVW motor leads from the inverter power structure to an energy dissipating resistor to stop the motor. This switching does not occur instantaneously and so enabling the power structure too early can cause electrical arcing across the contactor. To prevent this condition, the Resistive Brake Contact Delay can be set to the maximum time that it takes to fully close the contactor across the UVW motor lines so when the axis is enabled, the inverter power structure is not enabled until after the Resistive Brake Contact Delay Time has expired. Resistive Brake operation is only applicable to PM Motor types. The following sequence illustrates how the Resistive Brake Contact Delay factors into the overall Enable Sequence that may also include the operation of a Mechanical Brake.

Enable Sequence:

- 1) Switch to Starting state.
- 2) Activate Resistive Brake contactor to connect motor to inverter power structure.
- 3) Wait for “Resistive Brake Contact Delay” while Resistive Brake contacts close.
- 4) Enable inverter power structure.
- 5) Activate Mechanical Brake output to release brake.
- 6) Wait for “Mechanical Brake Release Delay” while brake releases.
- 7) Transition to Running state.

7.3.8.2.3 Attribute No. 615 – Mechanical Brake Release Delay

When enabling the axis, the Mechanical Brake Release Delay value determines the amount of time the axis shall delay transition from the Starting state to the Running or Testing states. This delay prevents any commanded motion of the axis until the external mechanical brake has had enough time to disengage.

Enable Sequence:

- 1) Switch to Starting state.
- 2) Activate Resistive Brake contactor to connect motor to inverter power structure.
- 3) Wait for “Resistive Brake Contact Delay” while Resistive Brake contacts close.

- 4) Enable inverter power structure.
- 5) Activate Mechanical Brake output to release brake.
- 6) Wait for “Mechanical Brake Release Delay” while brake releases.
- 7) Transition to Running (or Testing) state.

7.3.8.2.4 Attribute No. 616 – Mechanical Brake Engage Delay

When disabling the axis, the Mechanical Brake Engage Delay value determines the amount of time the inverter power structure shall remain enabled after the axis has decelerated to standstill. This allows time for an external mechanical brake to engage. The configured Stopping Mode determines the type of stopping sequence applied.

Disable Sequence (Category 0 Stop):

- 1) Disable inverter power structure.
- 2) Transition to Stopped state.
- 3) Deactivate Mechanical Brake output to engage brake.
- 4) Deactivate Resistive Brake contactor to disconnect motor from inverter power structure.

Disable Sequence (Category 1 Stop):

- 1) Switch to Stopping state.
- 2) Apply “Current Decel” or “Ramp Decel” method to stop motor.
- 3) Wait for zero speed or “Stopping Time Limit”, whichever occurs first.
- 4) Deactivate Mechanical Brake output to engage brake.
- 5) Wait for “Mechanical Brake Engage Delay” while brake engages.
- 6) Disable inverter power structure.
- 7) Transition to Stopped state.
- 8) Deactivate Resistive Brake contactor to disconnect motor from inverter power structure.

Disable Sequence (Category 2 Stop): The mechanical brake not used.

7.3.9 DC Bus Control attributes

The attributes table in Table 59 contains Motion Axis Object instance attributes associated with DC Bus control including functionality to address both under-voltage and over-voltage conditions.

Table 59 – DC Bus Control Attributes

Attr ID	Need in implem	Access rule	N V	Attribute name	Data type	Description of attribute	Semantics of values
620	Required - PVT	Get		DC Bus Voltage	REAL	Measured DC Bus Voltage.	Volts
621	Required - PVT	Get		DC Bus Voltage - Nominal	REAL	Normal DC Bus Voltage during operation as determined by averaging the DC Bus Voltage over a device specific time interval. This value is used as the basis for Bus Overvoltage and Undervoltage limits.	Volts

Attr ID	Need in implem	Access rule	N V	Attribute name	Data type	Description of attribute	Semantics of values
						NOTE If the device does not support this bus voltage averaging concept, the alternative is to hard code this value.	
624	Required - PVT	Set		Bus Regulator Action	USINT	Controls the method of operation of the DC Bus Regulator that addresses the regenerative over-voltage conditions that can occur when decelerating a motor. If Disabled, no regulation is applied to the DC Bus level to control regenerative energy sourced by the motor. When Shunt Regulator is selected, the associated shunt regulation hardware is applied to the DC Bus to dissipate regenerative energy via a resistor.	Enumeration: 0 = Disabled (R/PVT) 1 = Shunt Regulator (R/PVT) 2 to 127 = (reserved) 128 to 255 = (vendor specific)
625	Optional - PVT	Set		Bus Regulator Power Limit	REAL	Limits the amount of power allowed to transfer between the motor and the DC Bus during regenerative braking of the motor load. When using an external shunt resistor, set this value to its maximum value.	% Regulator Rated
627	Optional - PVT	Set		Power Loss Action	USINT	Set the reaction to a DC Bus under-voltage condition when the DC Bus drops below a hard-coded threshold in the device or the configured Power Loss Threshold.	Enumeration: 0 = Disabled & Coast 1 = Decel 2 = Continue 3 to 127 = (reserved) 128 to 255 = (vendor specific)
628	Optional - PVT	Set		Power Loss Threshold	REAL	Sets the Level for Power Loss as percent of nominal DC Bus Voltage. If this value is 0, the hard-coded threshold is used.	%
629	Optional - PVT	Set		Shutdown Action	UINT	Selects the action when transitioning to the Shutdown state. Standard action is to immediately disable the axis's power structure. Alternatively, action can be taken to drop the DC Bus voltage as well.	Enumeration: 0 = Disable (R) 1 = Drop DC Bus (O) 2 to 127 = (reserved) 128 to 255 = (vendor specific)

7.3.10 Power and thermal management attributes

The attribute tables in Table 60 and Table 61 contain all power and thermal related attributes associated with a Motion Axis Object instance.

Table 60 – Power and Thermal Management Status Attributes

Attr ID	Need in implem	Access rule	N V	Attribute name	Data type	Description of attribute	Semantics of values
635	Required - PVT	Get		Motor Capacity	REAL	Real-time estimate of the continuous rated motor thermal capacity utilized during operation based on the motor thermal model. A value of 100 % would indicate that the motor is being used at 100 % of rated capacity as determined by the continuous current rating of the motor.	% Motor Rated
636	Required - PVT	Get		Inverter Capacity	REAL	Real-time estimate of the continuous rated inverter thermal capacity utilized during operation based on the inverter thermal model. A value of 100 % would indicate that the inverter is being used at 100 % of rated capacity as determined by the continuous current rating of the inverter.	% Inverter Rated
637	Optional - PVT	Get		Converter Capacity	REAL	Real-time estimate of the continuous rated converter thermal capacity utilized during operation based on the converter thermal model. A value of 100 % would indicate that the converter is being used at 100 % of rated capacity as determined by the continuous current rating of the converter.	% Converter Rated
638	Optional - PVT	Get		Bus Regulator Capacity	REAL	Real-time estimate of the continuous rated bus regulator thermal capacity utilized during operation based on the bus regulator thermal model. A value of 100 % would indicate that the bus regulator is being used at 100 % of rated capacity as determined by the continuous current rating of the bus regulator.	% Regulator Rated
639	Optional - PVT	Get		Ambient Temperature	REAL	Current internal ambient temperature of the device enclosure.	°C
640	Optional - PVT	Get		Inverter Heatsink Temperature	REAL	Current temperature of the inverter heatsink, typically based on an embedded temp sensor.	°C
641	Optional - PVT	Get		Inverter Temperature	REAL	Current temperature of the power block used in the inverter's power structure, sometime referred to as the semiconductor junction temperature.	°C
642	Optional - PVT	Get		Motor Temperature	REAL	Current temperature of the motor stator, typically based on an embedded temp sensor.	°C
643	Optional - PVT	Get		Feedback 1 Temperature	REAL	Current temperature of the Feedback 1 device.	°C
644	Optional - PVT	Get		Feedback 2 Temperature	REAL	Current temperature of the Feedback 2 device.	°C
645	Optional - PVT	Get		Inverter Overload Limit	REAL	Factory set maximum limit for Inverter Capacity that when exceeded triggers the selected Inverter Overload action.	% Inverter Rated

Table 61 – Power and Thermal Management Configuration Attributes

Attr ID	Need in implem	Access rule	N V	Attribute name	Data type	Description of attribute	Semantics of values
646	Optional - PVT	Set		Motor Overload Action	USINT	Selects the device's response to a motor overload condition based on an I2t based motor thermal model. The motor overload condition occurs when the motor thermal model indicates that the Motor Capacity has exceeded the Motor Overload Limit. This motor overload action functionality is independent of the motor overload exception action functionality. No explicit action is taken by the device in the overload condition if None is the selected overload action. Selecting the Current Foldback action, however, results in a reduction of the current command via the Motor Thermal Current Limit attribute value that is reduced in proportion to the percentage difference between Motor Capacity and the Motor Overload Limit.	Enumeration: 0 = None (R) 1 = Current Foldback (O) 2 to 127 = (reserved) 128 to 255 = (vendor specific)
647	Optional - PVT	Set		Inverter Overload Action	USINT	Selects the device's response to an inverter overload alarm condition based on an I2t based inverter thermal model. The inverter overload alarm condition occurs when the inverter thermal model indicates that the Inverter Capacity has exceeded the Inverter Overload Limit. The Inverter Overload Action provides opportunities to mitigate the overload condition without stopping operation. This inverter overload action functionality is independent of the motor overload exception action functionality. No explicit action is taken by the device in the overload condition if None is the selected overload action. Selecting the Current Foldback action, however, results in a reduction of the current command via the Inverter Thermal Current Limit attribute value that is reduced in proportion to the percentage difference between Inverter Capacity and the Inverter Overload Limit.	Enumeration: 0 = None (R) 1 = Current Foldback (O) 2 to 127 = (reserved) 128 to 255 = (vendor specific)

7.3.11 Axis Status Attributes

7.3.11.1 General

The attributes table in Table 62 contains all status attributes associated with a Motion Axis Object instance.

Table 62 – Axis Status Attributes

Attr ID	Need in implem	Access rule	N V	Attribute name	Data type	Description of attribute	Semantics of values
650	Required - All	Get		Axis State	USINT	Enumerated value indicating the state of the axis.	Enumeration: 0 = Initializing 1 = Pre-Charge 2 = Stopped 3 = Starting 4 = Running 5 = Testing 6 = Stopping 7 = Aborting 8 = Major Faulted 9 = Start Inhibited 10 = Shutdown 11 to 255: Reserved
651	Required - All	Get		Axis Status	DWORD	Collection of bits indicating the internal status of the axis.	See Semantics in 7.3.11.2.1
652	Required - All	Get		Axis Status - Mfg	DWORD	Collection of bits indicating the status of the axis.	Bitmap: 0-31: Vendor Specific (Published in Product Manual)
653	Required - All	Get		Axis I/O Status	DWORD	The Axis I/O Status attribute is a 32-bit collection of bits indicating the state of standard digital inputs and outputs associated with the operation of this axis. A value of zero for a given input bit indicates a logical 0 value, while a value of 1 indicates a logical 1 value.	See Semantics in 7.3.11.2.2
654	Required - All	Get		Axis I/O Status - Mfg	DWORD	Collection of bits indicating the state of vendor specific digital inputs associated with the operation of this axis. A value of zero for a given input bit indicates a logical 0 value, while a value of 1 indicates a logical 1 value.	Bitmap: 0 to 31: Vendor Specific (Published in Product Manual)
94	Required - All	Set*		Status Data Set	BYTE	Bit mapped field that determines what status data values are to be transmitted to the controller via the Device-to-Controller (D-to-C) Connection.	See Semantics in 7.3.11.2.3

7.3.11.2 Semantics**7.3.11.2.1 Attribute No. 651 – Axis Status**

The Axis Status attribute is a 32-bit collection of bits indicating various internal status conditions of the axis instance, as specified in Table 63.

Table 63 – Axis Status bit definitions

Bit	Rec.Opt.	Name	Description
0	R	Local Control	This bit is set if axis is taking command reference and services from local interface instead of the remote (CIP Motion) interface.
1	R	Alarm	This bit is set if the axis has detected one or more exception conditions configured to generate an alarm.
2	R	DC Bus Up	This bit is set if DC Bus has charged up to an operational voltage level.
3	R	Power Structure Enabled	This bit is set if the axis amplifier is energized and capable of generating motor flux and torque.
4	IM	Motor Flux Up	This bit is set if motor flux for an induction motor has reached an operational level.
5	R	Tracking Command	This bit is set if the axis control structure is now actively tracking the command reference from motion planner.
6	R/P	Position Lock	This bit is set if the actual position is within Position Lock Tolerance of command position.
7	R/V	Velocity Lock	This bit is set if the actual velocity is within Velocity Lock Tolerance of command velocity.
8	R	Velocity Standstill	This bit is set if the actual velocity is within Velocity Standstill Window of 0.
9	R	Velocity Threshold	This bit is set if the absolute actual velocity is below Velocity Threshold.
10	R/V	Velocity Limit	This bit is set if the command velocity in velocity control mode is currently being limited by the Velocity Limiter.
11	R/V	Acceleration Limit	This bit is set if the rate of change of the command velocity in velocity control mode is currently being limited by the Acceleration Limiter.
12	R/V	Deceleration Limit	This bit is set if the rate of change of command acceleration in velocity control mode is currently being limited by the Deceleration Limiter.
13	R/C	Torque Threshold	This bit is set if the absolute filtered torque reference is below the Torque Threshold.
14	R/C	Current Limit	This bit is set if the command current, Iq, is currently being limited by the Current Vector Limiter.
15	R/C	Thermal Limit	This bit is set if Current Limit condition of the axis is being limited by any of the axis's Thermal Models.
16 to 31		Reserved	-

7.3.11.2.2 Attribute No. 653 – Axis I/O Status

The Axis I/O Status attribute is a 32-bit collection of bits indicating the state of standard digital inputs and outputs associated with the operation of this axis, as specified in Table 64.

A value of zero for a given input bit indicates a logical 0 value, while a value of 1 indicates a logical 1 value.

Table 64 – Axis I/O Status bit definitions

Bit	Rec. Opt.	Name
0	R	Enable Input
1	R/P	Home Input
2	R/P	Registration 1 Input
3	O/P	Registration 2 Input
4	R/P	Positive Overtravel Input

Bit	Rec. Opt.	Name
5	R/P	Negative Overtravel Input
6	O	Feedback 1 Thermostat
7	O	Feedback 1 Hall S1
8	O	Feedback 1 Hall S2
9	O	Feedback 1 Hall S3
10	O	Feedback 1 Ch A
11	O	Feedback 1 Ch B
12	O	Feedback 2 Ch A
13	O	Feedback 2 Ch B
14	O	Feedback 3 Ch A
15	O	Feedback 3 Ch B
16	O	Feedback 4 Ch A
17	O	Feedback 4 Ch B
18	O	Resistive Brake Output
19	O	Mechanical Brake Output
20 to 31		Reserved

7.3.11.2.3 Attribute No. 655 – Status Data Set

The Status Data Set attribute is a 8-bit collection of bits indicating what Status attributes shall be transmitted to the controller over the Device-to-Controller (D-to-C) Connection. Status data appears in the connection data structure in the same order as the bit numbers defined in Table 65, so Axis Fault Status would appear before, for example, Axis Alarm Status data in the Status Data Set structure. Multiple attributes comprising a selected Status Data Element are transmitted in the order listed from top to bottom, so the Axis Fault Code is transmitted before Mfg Axis Fault Code.

Table 65 – Status Data Set bit definitions

Bit	Status Data Element Produced	Data Type
0	Initialization Fault Code Initialization Fault Code - Mfg Axis Fault Code Axis Fault Code - Mfg Axis Alarm Code Axis Alarm Code - Mfg Start Inhibit Code Start Inhibit Code - Mfg Axis Fault Time Stamp	USINT USINT USINT USINT USINT USINT USINT USINT LINT
1	Axis Fault Status Axis Fault Status - Mfg	LWORD LWORD
2	Axis Alarm Status Axis Alarm Status - Mfg	LWORD LWORD
3	Start Inhibit Status Start Inhibit Status - Mfg	WORD WORD
4	Axis Status Axis Status - Mfg Axis I/O Status Axis I/O Status - Mfg	DWORD DWORD DWORD DWORD
5	(Reserved)	(Reserved)
6	(Reserved)	(Reserved)
7	(Vendor Specific)	(Vendor Specific)

7.3.12 Exception, fault, and alarm status attributes

7.3.12.1 General

The attribute table in Table 66 contains all exception, fault, and alarm related attributes associated with a Motion Axis Object instance. Exceptions are conditions that can occur during motion axis operation that have the potential of generating faults or alarms.

Table 66 – Exception, Fault and Alarm Status Attributes

Attr ID	Need in implem	Access rule	N V	Attribute name	Data type	Description of attribute	Semantics of values
655	Required - All	Get		Axis Exception Status	LWORD	A bit map that represents the current state of all standard exception conditions. See the Std. Exception Table bit map definition in 7.3.12.2 for detail on the bit locations. Each exception has a corresponding Exception Action. Exceptions that are configured to be Ignored are only visible in this attribute.	See Semantics in 7.3.12.2.
656	Required - All	Get		Axis Exception Status - Mfg	LWORD	A bit map that represents the current state of all manufacturer specific exception conditions. See the Mfg. Exception Table published in device product manual. Each exception has a corresponding Exception Action. Exceptions that are configured to be Ignored are only visible in this attribute.	See Mfg. Exception Table (Published in Product Manual)
657	Required - All	Get		Axis Fault Status	LWORD	A bit map that represents the state of all standard runtime faults. The bit map is identical to that of the Axis Exception Status attribute. Fault status bits when set are latched until a fault reset occurs. A fault reset clears the runtime fault bits, but the bits set again immediately if the underlying exception condition is still present. Any exceptions whose Exception Action is configured to ignore or report as alarms do not appear in this attribute.	See Semantics in 7.3.12.2.
658	Required - All	Get		Axis Fault Status - Mfg	LWORD	A bit map that represents the state of all manufacturer specific runtime faults. The bit map is identical to that of the Mfg. Axis Exception Status attribute. Fault status bits when set are latched until a fault reset occurs. A fault reset clears the runtime fault bits, but the bits set again immediately if the underlying exception condition is still present. Any exceptions whose Exception Action is configured to ignore or report as alarms do not appear in this attribute.	See Mfg. Exception Table (Published in Product Manual)
659	Required - All	Get		Axis Alarm Status	LWORD	A bit map that represents the current state of all standard alarm conditions. The bit map is identical to that of the Std. Axis Exception Status attribute. Only exception conditions whose Exception Action is configured to report as an alarm appear in this attribute, and will not be reported in the Axis Fault Status attribute. Alarm status bits when set	See Semantics in 7.3.12.2.

Attr ID	Need in implem	Access rule	N V	Attribute name	Data type	Description of attribute	Semantics of values
						are not latched and will clear as soon as the underlying exception condition is corrected.	
660	Required - All	Get		Axis Alarm Status - Mfg	LWORD	A bit map that represents the current state of all manufacturer specific alarm conditions. The bit map is identical to that of the Mfg. Axis Exception Status attribute. Only exception conditions whose Exception Action is configured to report as an alarm appear in this attribute, and will not be reported in the Axis Fault Status attribute. Alarm status bits when set are not latched and will clear as soon as the underlying exception condition is corrected.	See Mfg. Exception Table. (Published in Product Manual)
661	Required - All	Get		Axis Fault Code	USINT	An 8-bit enumeration that specifies the standard exception that led to the current faulted state. The value corresponds to the bit position of the exception condition, and a value of 0 indicates that no standard runtime faults currently exist. This attribute only reports the first runtime fault that transitioned the axis from an unfaulted state to the faulted state, and then only if that was not a Mfg. Axis Fault. Subsequent runtime faults that occur when the Axis Fault Code attribute or the Mfg. Axis Fault Code attribute is already nonzero are not reported in this attribute. NOTE 1 No runtime faults exist if both the Axis Fault Code and the Mfg. Axis Fault Code are zero. The Axis Fault Code and the Mfg. Axis Fault Code can never both be non-zero at the same time.	See Semantics in 7.3.12.2.
662	Required - All	Get		Axis Fault Code - Mfg	USINT	An 8-bit enumeration that specifies the manufacturer specific exception that led to the current faulted state. The value corresponds to the bit position of the exception condition, and a value of 0 indicates that no manufacturer runtime faults currently exist. This attribute only reports the first runtime fault that transitioned the axis from an unfaulted state to the faulted state, and then only if that was a Mfg. Axis Fault. Subsequent runtime faults that occur when either the Axis Fault Code attribute or the Mfg. Axis Fault Code attribute is already nonzero are not reported in this attribute. NOTE 2 No runtime faults exist if both the Axis Fault Code and the Mfg. Axis Fault Code are zero. The Axis Fault Code and the Mfg. Axis Fault Code can never both be non-zero at the same time.	See Mfg. Exception Table (Published in Product Manual)
663	Required - All	Get		Axis Fault Time Stamp	LINT	A 64-bit Time Stamps of the exception that led to the current non-zero Axis Fault Code or Mfg Axis Fault Code. The time stamp format is absolute follows the CIP Sync standard with 0 corresponding	Nanoseconds

Attr ID	Need in implem	Access rule	N V	Attribute name	Data type	Description of attribute	Semantics of values
						to 1970-01-01.	
664	Required - All	Get		Axis Alarm Code	USINT	An 8-bit enumeration that specifies the exception that led to the current standard alarm condition. The value corresponds to the bit position of the exception condition that led to the alarm, and a value of 0 indicates that no standard alarms currently are active. If the standard alarm condition is removed, this attribute will automatically be cleared or else be set to another standard alarm condition if one is present. If multiple standard alarm conditions are present, the Axis Alarm Code is set to the lowest active alarm bit number. The Axis Alarm Code and the Mfg. Axis Alarm Code can both be non-zero at the same time.	See Semantics in 7.3.12.2.
665	Required - All	Get		Axis Alarm Code - Mfg	USINT	An 8-bit enumeration that specifies the manufacturer specific exception that led to the current alarm condition. The value corresponds to the bit position of the manufacturer exception condition that led to the alarm, and a value of 0 indicates that no manufacturer alarms currently exist. If the manufacturer alarm condition is removed, this attribute will automatically be cleared or else set to another manufacturer alarm condition if one is present. If multiple manufacturer alarm conditions are present, the Mfg. Axis Alarm Code is set to the lowest active alarm bit number. The Axis Alarm Code and the Mfg. Axis Alarm Code can both be non-zero at the same time.	See Mfg. Exception Table (Published in Product Manual)
666	Optional - All	Get		Fault Sub Code	USINT	A 16-bit enumeration that specifies the fault sub code for the fault condition currently displayed on the front panel of the device (and reported in the Axis Fault Code). This attribute allows additional detail to be provided in the case of some faults, and has a different interpretation for each fault. Non-zero values indicate that additional detail is present.	Enumeration specific to fault condition.

7.3.12.2 Semantics: Standard Exception Table

Table 67 defines a list of standard exceptions associated with the Axis Exception Status attributes but also applicable to Axis Fault Status, Axis Fault Code, Axis Alarm Status, and Axis Alarm Code attributes.

Table 67 – Standard Exception Table

Bit	Exception	Description
0	-- Reserved --	This bit cannot be used since the Alarm Codes and Fault Code are defined by the associated exception bit number and an Alarm Code or Fault Code of 0 means no alarm or fault condition is present.
1	Motor Stall	Induction motor stall condition detected.
2	Motor Commutation	Permanent magnet motor commutation problem detected. Example would be an illegal state "111" or "000" for Hall Sensor.
4	Motor Overspeed FL	Motor speed has exceeded its maximum limit given by either Rotary Motor Max Speed or Linear Motor Max Speed attributes depending on the type of motor.
5	Motor Overspeed UL	Motor speed has exceeded the user defined Motor Overspeed User Limit margin above the Positive Velocity Limit or below the Negative Velocity Limit.
6	Motor Overtemperature FL	Motor temperature has exceeded its factory set temperature limit given by Motor Overtemperature Factory Limit, or the integral motor thermal switch has tripped.
7	Motor Overtemperature UL	Motor temperature has exceeded the user defined temperature limit given by Motor Overtemperature User Limit.
8	Motor Thermal Overload FL	Motor thermal model has exceeded its factory set thermal capacity limit given by Motor Thermal Overload Factory Limit.
9	Motor Thermal Overload UL	Motor thermal model has exceeded its user defined thermal capacity given by Motor Thermal Overload User Limit.
10	-- Reserved --	
11	-- Reserved --	
12	Inverter Overcurrent	Inverter current has exceeded the factory set peak or instantaneous current limit.
13	Inverter Overtemperature FL	Inverter temperature has exceeded its factory set temperature limit given by Inverter Overtemperature Factory Limit.
14	Inverter Overtemperature UL	Inverter temperature has exceeded the user defined temperature limit given by Inverter Overtemperature User Limit.
15	Inverter Thermal Overload FL	Inverter thermal model has exceeded its factory set thermal capacity limit given by Inverter Thermal Overload Factory Limit.
16	Inverter Thermal Overload UL	Inverter thermal model has exceeded its user defined thermal capacity given by Inverter Thermal Overload User Limit.
17	-- Reserved --	
18	-- Reserved --	
19	Converter Overtemperature FL	Converter temperature has exceeded its factory set temperature limit given by Converter Overtemperature Factory Limit.
20	Converter Overtemperature UL	Converter temperature has exceeded the user defined temperature limit given by Converter Overtemperature User Limit.
21	Converter Thermal Overload FL	Converter thermal model has exceeded its factory set thermal capacity limit given by Converter Thermal Overload Factory Limit.
22	Converter Thermal Overload UL	Converter thermal model has exceeded its user defined thermal capacity given by Converter Thermal Overload User Limit.
23	Converter AC Phase Loss	One or more phases have been lost on the AC line to the converter.
24	Converter AC Ground Current	Excessive Ground Current was detected by the converter.
25	-- Reserved --	
26	-- Reserved --	
27	Bus Regulator Overtemperature FL	Bus Regulator temperature has exceeded its factory set temperature limit given by Bus Regulator Overtemperature Factory Limit.
28	Bus Regulator Overtemperature UL	Bus Regulator temperature has exceeded the user defined temperature limit given by Bus Regulator Overtemperature User Limit.
29	Bus Regulator Thermal Overload FL	Bus Regulator thermal model has exceeded its factory set thermal capacity limit given by Bus Regulator Thermal Overload Factory Limit.
30	Bus Regulator Thermal Overload UL	Bus Regulator thermal model has exceeded its user defined thermal capacity given by Bus Regulator Thermal Overload User Limit.
31	Bus Undervoltage FL	DC Bus voltage level is below the factory set limit given by Bus Undervoltage Factory Limit.

Bit	Exception	Description
32	Bus Undervoltage UL	DC Bus voltage level is below user defined limit given by Bus Undervoltage User Limit.
33	Bus Overvoltage FL	DC Bus voltage level is above the factory set limit given by Bus Overvoltage Factory Limit.
34	Bus Overvoltage UL	DC Bus voltage level is above user defined limit given by Bus Overvoltage User Limit.
35	-- Reserved --	
36	-- Reserved --	
37	Feedback 1 Signal Noise	Noise induced A/B channel state changes were detected on the Feedback 1 signal connection. Specifically, the rate at which these noise events have occurred has exceeded a factory set rate limit.
38	Feedback 1 Signal Loss	One or more A/B channel Feedback 1 device signal connections are open, shorted, missing, or severely attenuated. Specifically, the detected voltage levels of the signals are below a factory set limit.
39	Feedback 1 Serial Communication	The number of consecutive missed or corrupted serial data packets over the serial data channel associated with Feedback 1 has exceeded a factory set limit.
40	Feedback 1 Power	A problem has been found related to the power supply, or battery, associated with the Feedback 1 device.
41	-- Reserved --	
43	-- Reserved --	
44	Feedback 2 Signal Noise	Noise induced A/B channel state changes were detected on the Feedback 2 signal connection. Specifically, the rate at which these noise events have occurred has exceeded a factory set rate limit.
45	Feedback 2 Signal Loss	One or more A/B channel Feedback 2 device signal connections are open, shorted, missing, or severely attenuated. Specifically, the detected voltage levels of the signals are below a factory set limit.
46	Feedback 2 Serial Communication	The number of consecutive missed or corrupted serial data packets over the serial data channel associated with Feedback 2 has exceeded a factory set limit.
47	Feedback 2 Power	A problem has been found related to the power supply, or battery, associated with the Feedback 2 device.
48	-- Reserved --	
49	-- Reserved --	
50	Hardware Overtravel Positive	Axis moved beyond the physical travel limits in the positive direction and activated the Positive Overtravel limit switch.
51	Hardware Overtravel Negative	Axis moved beyond the physical travel limits in the negative direction and activated the Negative Overtravel limit switch.
52	-- Reserved --	
53	-- Reserved --	
54	Excessive Position Error	The Position Error value of the position control loop has exceeded the configured value for Position Error Tolerance.
55	Excessive Velocity Error	The Velocity Error value of the velocity control loop has exceeded the configured value for Velocity Error Tolerance.
56	Overtorque Limit	Motor torque has risen above user defined maximum torque level given by Overtorque Limit.
57	Undertorque Limit	Motor torque has dropped below user defined minimum torque level given by Undertorque Limit.
58	-- Reserved --	
59	-- Reserved --	
60	Illegal Mode Change	Controller has specified an unsupported Control Mode or Feedback Configuration.
61	Enable Input Deactivated	Enable has been deactivated while the axis is in Running state.
62	Controller Initiated Exception	Exception generated specifically by controller.
63	External Exception Input	Exception generated by external input to device.

7.3.13 Exception limit attributes

The attribute tables in Table 68 and Table 69 contain all exception limit related attributes associated with a Motion Axis Object instance. Exception Limit attributes define the conditions under which a corresponding exception is generated during motion axis operation that has the potential of generating either a fault or alarm. They are typically associated with temperature, current, and voltage conditions of the device that are continuous in nature. Factory Limits (FL) for exceptions are usually hard coded in the device and typically result in a major fault condition. User Limits (UL) for exceptions are configurable and typically used to generate a minor fault, or alarm condition. For this reason, the User Limits are generally set inside the corresponding Factory Limits. The triggering of a User Limit exception does not preclude triggering of the corresponding Factory Limit exception; the two exception trigger conditions are totally independent of one another.

Table 68 – Exception Factory Limit Info Attributes

Attr ID	Need in implem	Access rule	N V	Attribute name	Data type	Description of attribute	Semantics of values
680	Optional - PVT	Get		Motor Overtemperature Factory Limit	REAL	Returns the Factory Limit for the Motor Overtemperature FL exception based on a factory set value related to the Motor Max Winding Temperature of the motor.	°C
681	Optional - PVT	Get		Motor Thermal Overload Factory Limit	REAL	Returns the Factory Limit for the Motor Thermal Overload FL exception based on a factory set value related to the Motor Thermal Overload rating of the motor.	% Motor Rated
682	Optional - PVT	Get		Inverter Overtemperature Factory Limit	REAL	Returns the Factory Limit for the Inverter Overtemperature FL exception.	°C
683	Optional - PVT	Get		Inverter Thermal Overload Factory Limit	REAL	Returns the Factory Limit for the Inverter Thermal Overload FL exception.	% Inverter Rated
684	Optional - PVT	Get		Converter Overtemperature Factory Limit	REAL	Returns the Factory Limit for the Converter Overtemperature FL exception.	°C
685	Optional - PVT	Get		Converter Thermal Overload Factory Limit	REAL	Returns the Factory Limit for the Converter Thermal Overload FL exception.	% Converter Rated
686	Optional - PVT	Get		Bus Regulator Overtemperature Factory Limit	REAL	Returns the Factory Limit for the Bus Regulator Overtemperature FL exception.	°C
687	Optional PVT	Get		Bus Regulator Thermal Overload Factory Limit	REAL	Returns the Factory Limit for the Bus Regulator Thermal Overload FL exception.	% Regulator Rated
688	Optional - PVT	Get		Bus Overvoltage Factory Limit	REAL	Returns the Factory Limit for the Bus Overvoltage FL exception.	Volts
689	Optional - PVT	Get		Bus Undervoltage Factory Limit	REAL	Returns the Factory Limit for the Bus Undervoltage FL exception.	Volts

Table 69 – Exception User Limit Configuration Attributes

Attr ID	Need in implem	Access rule	N V	Attribute name	Data type	Description of attribute	Semantics of values
695	Optional - PVT	Set		Motor Overspeed User Limit	REAL	Sets an incremental speed above the Positive Velocity Limit and below the Negative Velocity Limit that is allowable before throwing a Motor Overspeed UL exception.	Velocity Control Units/s
696	Optional - PVT	Set		Motor Overtemperature User Limit	REAL	Sets User Limit for the Motor Overtemperature UL exception.	°C
697	Optional - PVT	Set		Motor Thermal Overload User Limit	REAL	Sets User Limit for the Motor Thermal Overload UL exception.	% Motor Rated
698	Optional - PVT	Set		Inverter Overtemperature User Limit	REAL	Sets User Limit for the Inverter Overtemperature UL exception.	°C
699	Optional - PVT	Set		Inverter Thermal Overload User Limit	REAL	Sets User Limit for the Inverter Thermal Overload UL exception.	% Inverter Rated
700	Optional - PVT	Set		Converter Overtemperature User Limit	REAL	Sets User Limit for the Converter Overtemperature UL exception.	°C
701	Optional - PVT	Set		Converter Thermal Overload User Limit	REAL	Sets User Limit for the Converter Thermal Overload UL exception.	% Converter Rated
702	Optional - PVT	Set		Bus Regulator Overtemperature User Limit	REAL	Sets User Limit for the Bus Regulator Overtemperature UL exception.	°C
703	Optional - PVT	Set		Bus Regulator Thermal Overload User Limit	REAL	Sets User Limit for the Bus Regulator Thermal UL exception.	% Regulator Rated
704	Optional - PVT	Get		Bus Overvoltage User Limit	REAL	Sets User Limit for the Bus Overvoltage UL exception. Unlike the corresponding Factory Limit, which is specified in Volts, the User Limit is based on percent of Nominal Bus Voltage during operation.	% Nominal Bus Voltage
705	Optional - PVT	Get		Bus Undervoltage User Limit	REAL	Sets User Limit for the Bus Undervoltage UL exception. Unlike the corresponding Factory Limit, which is specified in Volts, the User Limit is based on percent of Nominal Bus Voltage during operation.	% Nominal Bus Voltage

7.3.14 Exception action configuration attribute

7.3.14.1 General

The configuration attribute in Table 70 controls the action performed by the Motion Axis Object as a result of an exception condition. A unique exception action is defined for each supported exception condition.

Table 70 – Exception Action Configuration Attribute

Attr ID	Need in implem	Access rule	N V	Attribute name	Data type	Description of attribute	Semantics of values
667	Required - All	Set		Exception Action	USINT [64]	The Exception Action attribute is 64-element array of enumerated bytes that specifies the action for the associated standard exception. See Semantics in 7.3.14.2 for details.	Enumeration: 0 = Ignore (O) 1 = Alarm (O) 2 = Minor Fault (O) 3 = Major Fault (R) 4 to 255 = reserved
668	Required - All	Set		Exception Action - Mfg	USINT [64]	The Exception Action attribute is 64-element array of enumerated bytes that specifies the action for the associated manufacturer specific exception. See Semantics in 7.3.14.2 for details	Enumeration: 0 = Ignore (O) 1 = Alarm (O) 2 = Minor Fault (O) 3 = Major Fault (R) 4 to 255 = reserved

7.3.14.2 Semantics: Attribute No. 667-668 – Exception Action

The Exception Action and Exception Action–Mfg attributes are 64-element array of enumerated bytes that specifies the action for the associated standard or manufacturer specific exception, respectively (see Table 71). For a given exception, certain actions may not be supported. Each device product shall specify the available actions for each exception that is supported.

Table 71 – Exception Action Bit Definitions

Enum.	Name	Description
0	Ignore	Ignore instructs the Motion Axis Object to completely ignore the exception condition. For some exceptions that are fundamental to the operation of the device, it may not be possible to ignore the condition.
1	Alarm	Alarm action simply sets the associated bit in the Alarm Status word but does not otherwise affect axis behavior. For some exceptions that are fundamental to the operation of the axis, it may not be possible to simply indicate the condition as an Alarm.
2	Minor Fault	Minor Fault action reports the exception by setting the associated bit in the Fault Status word but does not affect axis behavior. Minor Faults allow the controller to take application specific action in response to the exception condition. For some exceptions that are fundamental to the operation of the axis, it may not be possible to simply indicate the condition as a Minor Fault.

Enum.	Name	Description
3	Major Fault	Major Fault action results in both setting the associated bit in the Fault Status word and bringing the axis to a stop based on the factory set “best” available stopping method. This “best” stopping method includes both the method of decelerating the motor to a stop and the final state of the axis given the expected level of control still available. The level of control available depends on the specific exception condition and on the configured control mode. The available deceleration methods are defined by the Stopping Mode attribute. Standard stopping modes, listed in decreasing levels of deceleration control, are as follows: 1) Ramp Decel 2) Current Decel 3) Coast In general, the “best” stopping mode is the most controlled deceleration method still available given the exception condition. The final state of the axis in response to the Major Fault exception action can be any one of the following states that are listed in decreasing levels of control functionality: 1) Hold (Stopped with Holding Torque) 2) Disable (Power Structure Disabled) 3) Shutdown (DC Bus Power Disabled) The “best” final state of the axis is the state with the most control functionality still available given the exception condition. If the application requires exception action that is a more severe stopping action than the factory set “best” method, the controller can initiate that action programmatically. If the application requires exception action that is less severe than the factory set “best” method, the controller shall configure the axis for a Minor Fault exception action and handle the fault directly. This may put device and motor components at risk and should only be allowed by the device when there is an opportunity, albeit temporal, for the device to remain operational. This is important in applications where the value of the product is higher than the value of the motor or device.
4 to 255	Reserved	-

7.3.15 Initialisation fault status attributes

7.3.15.1 General

The attribute table in Table 72 contains all initialization fault related attributes associated with a Motion Axis Object instance. Initialization Faults are conditions that can occur during the device initialization process that prevent normal operation of the device.

Table 72 – Initialisation Fault Status Attributes

Attr ID	Need in implem	Access rule	N V	Attribute name	Data type	Description of attribute	Semantics of values
669	Required All	Get		Initialization Fault Status	DWORD	A bit map that represents the state of all standard initialization faults. These faults prevent any motion, and do not have configurable fault actions. Examples of initialization faults are corrupted memory data, calibration errors, firmware startup problems, or an invalid configuration attribute value. Initialization faults cannot be cleared with a Fault Reset service, although a power-cycle provides a new attempt at initialization.	See Semantics in 7.3.15.2

Attr ID	Need in implem	Access rule	N V	Attribute name	Data type	Description of attribute	Semantics of values
670	Required - All	Get		Initialization Fault Status - Mfg	DWORD	A bit map that represents the state of all manufacturer specific initialization faults. These faults prevent any motion, and do not have configurable fault actions. Examples of initialization faults are corrupted memory data, calibration errors, firmware startup problems, or an invalid configuration attribute value. Initialization faults cannot be cleared with a Fault Reset service, although a power-cycle provides a new attempt at initialization.	See Mfg. Initialization Fault Table (Published in Product Manual)
671	Required - All	Get		Initialization Fault Code	USINT	An 8-bit enumeration that specifies the initialization fault that led to the current faulted state. The value corresponds to the bit position of the standard initialization fault condition, and a value of 0 indicates that no standard initialization faults currently exist. This attribute only reports the first initialization fault that transitioned the axis from an unfaulted state to a faulted state, and then only if that was a standard Initialization Fault. NOTE 1 No initialization faults exist if both the Initialization Fault Code and the Mfg. Initialization Fault Code are zero. The Initialization Fault Code and the Mfg. Initialization Fault Code can never both be non-zero at the same time.	See Semantics in 7.3.15.2
672	Required - All	Get		Initialization Fault Code - Mfg	USINT	An 8-bit enumeration that specifies the manufacturer specific initialization fault that led to the current faulted state. The value corresponds to the bit position of the initialization fault condition, and a value of 0 indicates that no manufacturer specific initialization faults currently exist. This attribute only reports the first initialization fault that transitioned the axis from an unfaulted state to a faulted state, and then only if that was a Mfg. Initialization Fault. NOTE 2 No initialization faults exist if both the Initialization Fault Code and the Mfg. Initialization Fault Code are zero. The Initialization Fault Code and the Mfg. Initialization Fault Code can never both be non-zero at the same time.	See Mfg. Initialization Fault Table (Published in Product Manual)

7.3.15.2 Semantics: Standard Initialization Fault Table

Table 73 defines a list of standard faults associated with the Initialization Fault Status attribute and also applicable to the Initialization Fault Code attribute.

Table 73 – Standard Initialisation Fault Table

Bit	Fault	Bit	Fault
000	-- Reserved --	016	(Reserved)
001	Boot Block Checksum Fault	017	(Reserved)
002	Main Block Checksum Fault	018	(Reserved)
003	Nonvolatile Memory Checksum Fault	019	(Reserved)
004	(Reserved)	020	(Reserved)
005	(Reserved)	021	(Reserved)
006	(Reserved)	022	(Reserved)
007	(Reserved)	023	(Reserved)
008	(Reserved)	024	(Reserved)
009	(Reserved)	025	(Reserved)
010	(Reserved)	026	(Reserved)
011	(Reserved)	027	(Reserved)
012	(Reserved)	028	(Reserved)
013	(Reserved)	029	(Reserved)
014	(Reserved)	030	(Reserved)
015	(Reserved)	031	(Reserved)

7.3.16 Start inhibit status attributes

7.3.16.1 General

The attribute table in Table 74 contains all Start Inhibit related attributes associated with a Motion Axis Object instance. Start Inhibits are conditions that prevent transition of the motion axis from the Stopped State into any of the operational states.

Table 74 – Start Inhibit Status Attributes

Attr ID	Need in implem	Access rule	N V	Attribute name	Data type	Description of attribute	Semantics of values
673	Required - PVT	Get		Start Inhibit Status	WORD	A bit map that specifies the current state of all standard conditions that can inhibit starting of the axis.	See Semantics in 7.3.16.2
674	Required - PVT	Get		Start Inhibit Status - Mfg	WORD	A bit map that specifies the current state of all manufacturer specific conditions that can inhibit starting of the axis.	See Mfg. Start Inhibit Table (Published in Product Manual)
675	Required - PVT	Get		Start Inhibit Code	USINT	The Start Inhibit Condition attribute is an enumeration that indicates a condition that presently is causing the axis to be in the start inhibited state. If the inhibiting condition is removed, this attribute will automatically be cleared or else set to another standard inhibiting condition if one is present. If multiple standard start inhibit conditions are present, the Start Inhibit Code is set to the lowest active start inhibit bit number. The Start Inhibit Code and the Mfg. Start Inhibit Code can both be non-zero at the same time.	See Semantics in 7.3.16.2

Attr ID	Need in implem	Access rule	N V	Attribute name	Data type	Description of attribute	Semantics of values
676	Required - PVT	Get		Start Inhibit Code - Mfg	USINT	The Mfg. Start Inhibit Condition attribute is a manufacturer specific enumeration that indicates a condition that presently is causing the axis to be in the start inhibited state. If the inhibiting condition is removed, this attribute will automatically be cleared or else set to another manufacturer specific inhibiting condition if one is present. If multiple manufacturer specific start inhibit conditions are present, the Mfg. Start Inhibit Code is set to the lowest active start inhibit bit number. The Start Inhibit Code and the Mfg. Start Inhibit Code can both be non-zero at the same time.	See Mfg. Start Inhibit Tables (Published in Product Manual)

7.3.16.2 Semantics: Standard Start Inhibit Table

Table 75 defines a list of standard start inhibits associated with the Start Inhibit Status attribute and also applicable to the Start Inhibit Status Code attribute.

Table 75 – Standard Start Inhibit Table

Bit	Start Inhibit	Bit	Start Inhibit
000	-- Reserved --	008	(Reserved)
001	Enable Input Inactive	009	(Reserved)
002	(Reserved)	010	(Reserved)
003	(Reserved)	011	(Reserved)
004	(Reserved)	012	(Reserved)
005	(Reserved)	013	(Reserved)
006	(Reserved)	014	(Reserved)
007	(Reserved)	015	(Reserved)

7.3.17 Axis statistical attributes

Table 76 includes attributes that provide useful statistics on motion axis operation.

Table 76 –Statistical Attributes

Attr ID	Need in implem	Access rule	N V	Attribute name	Data type	Description of attribute	Semantics of values
710	Optional - PVT	Get		Control Power-up Time	REAL	Elapsed time since axis control power was last applied.	Seconds
711	Optional - PVT	Get		Cumulative Run Time	REAL	Accumulated time that the axis has been powering the Running state.	Hours
712	Optional - PVT	Get		Cumulative Energy Usage	REAL	Accumulated output energy of the axis.	kW·h
713	Optional – PVT	Get		Cumulative Motor Revs	LINT	Cumulative number of times motor shaft has turned. (Rotary Motors Only)	

Attr ID	Need in implem	Access rule	N V	Attribute name	Data type	Description of attribute	Semantics of values
714	Optional - PVT	Get		Cumulative Main Power Cycles	DINT	Cumulative number of times AC Mains has been cycled.	
715	Optional - PVT	Get		Cumulative Control Power Cycles	DINT	Cumulative number of times Control Power has been cycled.	

7.3.18 Axis info attributes

Table 77 includes attributes that provide information about the associated hardware capabilities of Motion Axis Object instance.

Table 77 – Axis Info Attributes

Attr ID	Need in implem	Access rule	N V	Attribute name	Data type	Description of attribute	Semantics of values
720	Required – PVT	Get		Inverter Rated Output Voltage	REAL	Drive inverter output voltage rating. This value is hard coded in the device.	Volts (RMS)
721	Required – PVT	Get		Inverter Rated Output Current	REAL	Drive inverter output current rating. This value is hard coded in the device.	Amperes (RMS)
722	Required – PVT	Get		Inverter Rated Output Power	REAL	Drive inverter output power rating. This value is hard coded in the device.	kW
723	Optional - PVT	Get		Converter Rated Output Current	REAL	Drive converter output current rating. This value is determined by the motion axis from the associated converter.	Amperes
724	Optional - PVT	Get		Converter Rated Output Power	REAL	Drive converter output power rating. This value is determined by the motion axis from the associated converter.	kW

7.3.19 General purpose I/O attributes

Table 78 includes attributes that provide to general purpose analog and digital I/O associated with the Motion Axis Object instance.

Table 78 – General Purpose I/O Attributes

Attr ID	Need in implem	Access rule	N V	Attribute name	Data type	Description of attribute	Semantics of values
730	Optional – PVT	Get		Digital Inputs	DWORD	A 32-bit word with whose bits can be assigned by the vendor to general purpose digital inputs.	Vendor Specific Bit Map
731	Optional – PVT	Set		Digital Outputs	DWORD	A 32-bit word with whose bits can be assigned by the vendor to general purpose digital outputs.	Vendor Specific Bit Map
732	Optional – PVT	Get		Analog Input 1	REAL	General purpose analogue input 1 level.	% Full Scale

Attr ID	Need in implem	Access rule	N V	Attribute name	Data type	Description of attribute	Semantics of values
733	Optional - PVT	Get		Analog Input 2	REAL	General purpose analogue input 2 level.	% Full Scale
734	Optional - PVT	Set		Analog Output 1	REAL	General purpose analogue output 2 level.	% Full Scale
735	Optional - PVT	Set		Analog Output 2	REAL	General purpose analogue output 2 level.	% Full Scale

7.3.20 Local Mode Attributes

Table 79 contains all local mode attributes associated with a Motion Axis Object instance. These attributes govern the behavior of the Motion Axis Object instance when a local (i.e. non-CIP Motion) device interface requests control of axis behavior.

Table 79 – Local Mode Configuration Attributes

Attr ID	Need in implem	Access rule	N V	Attribute name	Data type	Description of attribute	Semantics of values
750	Optional	Set		Local Control	USINT	Mechanism for controller to allow a local interface to request local control. Control implies, in this case, the ability to change the state or behavior of the axis. Local Control Not Allowed configures the device to prevent a local interface from taking control of the axis. Local Control Conditionally Allowed configures the device to prevent a local interface from taking control of the axis while the axis is in an Operational State, such as Running or Testing. Local Control Allowed configures the device to allow a local interface to take control of the axis, even in an Operational State. Some devices may not support this state.	Enumeration: 0 = Local Control Not Allowed 1 = Local Control Conditionally Allowed 2 = Local Control Allowed (O)

7.4 Common services

7.4.1 Supported services

The Motion Axis Object provides the CIP Common Services specified in Table 80.

Table 80 – Motion Axis Object – Common Services

Service Code (Hex)	Need in Implementation		Service Name	Description of Service
	Class	Instance		
03 _{hex}	Required	Required	Get_Attributes_List	Returns the contents of the selected gettable attributes for given instance.
04 _{hex}	Required	Required	Set_Attributes_List	Sets the content of the selected settable attributes for given instance.
0E _{hex}	Required	Required	Get_Attributes_Single	Returns the contents of the selected gettable attribute for given instance.

Service Code (Hex)	Need in Implementation		Service Name	Description of Service
	Class	Instance		
10 _{hex}	Required	Required	Set_Attributes_Single	Sets the content of the selected settable attribute for given instance.
TBD	Optional	N/A	Group_Sync	Passes controller's System Time Offset to device and reports if this device can be synchronised with the controller as part of a group of associated devices. If the device can be synchronised, the device also returns the device's System Time Offset to the controller in the response.

7.4.2 Service specific data

7.4.2.1 General

Common Services supported by the Motion Axis Object have associated Request Data and Response Data. The format and syntax of the Service Specific Data depends on the specified Service Code. Descriptions of the Service Specific Data for the Set/Get_Attribute_List and Set/Get_Attribute_Single services are already defined in CIP (see IEC 61158-5-2 and IEC 61158-6-2). A description of the Service Specific Data associated with the Group_Sync service (part of the CIP Sync extensions to CIP specifications) is detailed in the following subclause.

7.4.2.2 Group_Sync

The Group_Sync service is used to synchronise the Motion Axis Object associated with this device node to the controller issuing the service request and, ultimately, to all other devices comprising the controller's Motion Group. The service request passes the controller's System Time Offset and replies with a 1 (true) if the device is presently group synchronised to the IEC 61588:2004 time master or a 0 (false) if it is not group synchronised. The format and syntax of Group_Sync Request Data is specified in Table 81.

Table 81 – Group_Sync Request Data Structure

Name	Type	Description of Request Parameter	Semantics of Values
System Time Offset	LINT	Specifies the current System Time Offset of the controller.	Nanoseconds (CIP Sync Absolute)

The System Time Offset value is used to initialise the last Controller Time Offset value used as part of the System Time Offset Check and Compensation algorithm. The format and syntax of Group_Sync Response Data is specified in Table 82.

Table 82 – Group_Sync Response Data Structure

Name	Type	Description of Response Parameter	Semantics of Values
IsSynchronized	BOOL	Boolean value that indicates whether or not the device is presently group synchronized.	0 = device not synchronised 1 = device synchronised
System Time Offset	LINT	Specifies the current System Time Offset of the device.	Nanoseconds (CIP Sync Absolute)

There are two class attributes used by the Motion Axis Object to determine if the device is group synchronised:

- 1) Sync Update Delay;

2) Sync Threshold.

The Sync Update Delay attribute determines the minimum time to wait after the last System Time Offset change greater than Sync Threshold before the device is considered to be group synchronized. This value is the product of the number of network hops the device is away from the grandmaster, as determined by the Time Sync Object attribute, Steps Removed, and the time synchronization interval, as determined by the Time Sync Object attribute, Sync Interval. The Sync Update Delay attribute is typically several seconds.

The Sync Threshold is the value used to determine if the device's local clock is frequency synchronized to the master System Time clock. System Time Offset changes at the motion device that exceed the Sync Threshold are considered adjustments to the System Time reference system rather than minor adjustment due to local clock drift. This kind of adjustment is indicative of a motion device whose local clock is not yet fully synchronized to the master clock. Though application dependent, the minimum Sync Threshold value depends on the motion device's time sync implementation, but is typically in the order of microseconds for a hardware assisted clock implementation.

Once the controller and all the devices in the Motion Group are group synchronized, the controller can then set the Synchronous Control bit in the Controller-to-Device (C-to-D) Connection header of the next cyclic packet sent to the motion devices indicating that the device can schedule its next Device-to-Controller (D-to-C) Connection update based on the Controller Time Stamp and Controller Update Period contained in the connection data. The motion device indicates that it is now sending scheduled connection data based on the Controller Update Period by setting the Sync Mode bit in the Device-to-Controller (D-to-C) Connection header of the next cyclic packet sent to the controller. Data exchange between the controller and the device proceed thereafter according the CIP Motion Timing Model.

7.5 Object specific services

7.5.1 Supported services

The Motion Axis Object provides the Object Specific Services specified in Table 83.

Table 83 – Motion Axis Object – Object Specific Services

Service Code (Hex)	Need in Implementation		Service Name	Description of Service
	Class	Instance		
4B _{hex}		Required	Get_Axis_Attributes_List	Returns the contents of the selected gettable Motion Axis Object attributes.
4C _{hex}		Required	Set_Axis_Attributes_List	Sets the content of the selected settable Motion Axis Object attributes.
4D _{hex}		Required	Set_Cyclic_Write_List	Sets the attribute contents of the Cyclic Write Data block.
4E _{hex}		Required	Set_Cyclic_Read_List	Sets the attribute contents of the Cyclic Read Data block.
4F _{hex}		Optional - FC	Run_Motor_Test	Initiates the selected test on the motor to measure various motor parameters.
50 _{hex}		Optional - FC	Get_Motor_Test_Data	Returns the results of the preceding Run Motor Test service.
51 _{hex}		Required - C	Run_Inertia_Test	Initiates the selected test on the motor to measure the inertia.
52 _{hex}		Required - C	Get_Inertia_Test_Data	Returns the results of the preceding Run Inertia Test service.

Service Code (Hex)	Need in Implementation		Service Name	Description of Service
	Class	Instance		
53 _{hex}		Required	Run_Hookup_Test	Initiates the selected test to determine the condition of the motor and feedback device connections.
54 _{hex}		Required	Get_Hookup_Test_Data	Returns the results of the preceding Run Hookup Test service.
55-63 _{hex}			Reserved	

7.5.2 Service specific data

7.5.2.1 General

Object Specific Services supported by the Motion Axis Object have associated Request Data and Response Data. The format and syntax of the Service Specific Data depends on the specified Service Code. All parameters contained in the Service Specific Data are word aligned. This is true regardless of whether the service specific request data is passed in the Controller-to-Device (C-to-D) Connection or as part of an Explicit messaging connection. A description of the Service Specific Data associated with each service is shown in the following subclauses.

7.5.2.2 Get Axis Attribute List

The Get Axis Attribute List service provides a mechanism to read the current value of one or more Motion Axis Object attributes, including attributes having a multi-dimensional array data type. The buffer/array transfer mechanism can be used to transfer large data log arrays from the device to facilitate features like high speed trending.

The format of the Request Data Block for this service is shown in Figure 53.

← 32-bit Word →			
Get Axis Attribute List Request Format			
Number of Attributes		-	
Attr ID 1		Attr 1 Dimension	Attr 1 Element Size
Attr Start Index 1 (array only)		Attr Data Elements 1 (array only)	
...		...	
Attr ID 2		Attr 2 Dimension	Attr 2 Element Size
Attr Start Index 2 (array only)		Attr Data Elements 2 (array only)	
...		...	
Attr ID n		Attr n Dimension	Attr n Element Size
Attr Start Index n (array only)		Attr Data Elements n (array only)	
...		...	

Figure 53 – Get Axis Attribute List Request Format

Definitions of the individual parameters in this data structure are as follows:

- Number of Attributes – represents the number of attributes contained in the Get Axis Attribute service request.
- Attr ID – identifies the targeted Motion Axis Object attribute to get.
- Attr Dimension – determines the dimension of the attribute array. A dimension of zero means the attribute is a singular data element and, therefore, not really an array at all. Multidimensional arrays (dimension > 1) are supported by adding additional Attr Start Index and Addr Data Elements values. If there is an error associated with a specific requested get attribute operation, the device shall indicate this by setting the Attr Dimension to 0xFF, in which case the Element Size field contains the specific error code.

When an error code is present, neither the array index parameters nor the attribute data fields are returned.

- **Attr Element Size** – determines the size, in bytes, of the attribute data element. Data elements shall be word aligned; 32-bit words are 32-bit aligned and 16-bit words are 16-bit aligned. Padding may be added to maintain word alignment. If there is an error associated with a specific requested get attribute operation, the Element Size field contains the CIP Common General Status error code. When an error code is present neither the array index parameters nor the attribute data fields are returned.
- **Attr Start Index** – identifies the starting index for the array of attribute values in the Attr Data section. This field is only present when the attribute data type is an array (i.e., dimension > 0).
- **Attr Data Elements** – determines the number of data element values in the Attr Data section for the associated index. This field is only present when the attribute data type is an array (i.e., dimension > 0).

The structure of Response Data Block for this service is as shown in Figure 54.

← 32-bit Word →		
Get Axis Attribute List Response Format		
Number of Attributes	-	
Attr ID 1	Attr 1 Dimension	Attr 1 Element Size
Attr Start Index 1 (array only)	Attr Data Elements 1 (array only)	
...	...	
Attr Data 1		
...		
Attr ID 2	Attr 2 Dimension	Attr 2 Element Size
Attr Start Index 2 (array only)	Attr Data Elements 2 (array only)	
...	...	
Attr Data 2		
...		
Attr ID n	Attr n Dimension	Attr n Element Size
Attr Start Index n (array only)	Attr Data Elements n (array only)	
...	...	
Attr Data n		
...		

Figure 54 – Get Axis Attribute List Response Format

Definitions of the individual parameters in this data structure are identical to the request data structure with the addition of the Attribute Data element, which is defined as follows:

- **Attr Data** – contains the current value(s) of the targeted Motion Axis Object attribute indicated by the Attr ID. If the attribute is an array, the value(s) are listed according to the Attr Start Index and the number of Attr Data Elements. For multidimensional arrays (dimension > 1), the sequence of data move sequentially through the indices from right to left. Examples are provided below.

EXAMPLE 1

Figure 55 illustrates a Get Axis Attribute List Request for a simple 4-byte DINT scalar attribute. Since the Attr Dim is 0, the Attr Start Index and Attr Data Element fields are omitted.

Get Axis Attribute List Response – 0 Dimensional Data Example		
Number of Attributes = 1	-	
Attr ID = 25	Attr Dim = 0	Attr Elem Size = 4
Attr Data = ?		

Figure 55 – Get Axis Attribute List Response – Example 1

EXAMPLE 2

Figure 56 illustrates a Get Axis Attribute List Request for the first three elements of a one dimensional array attribute that is a UINT. Since the Attr Elem Size for this attribute is a 2-byte value, a Pad byte is added to maintain word alignment for any connection data to follow.

Get Axis Attribute List Response – 1 Dimensional Array Example		
Number of Attributes = 1	-	
Attr ID = 43	Attr Dim = 1	Attr Elem Size = 2
Attr Start Index 1 = 0	Attr Data Elements 1 = 3	
Attr Data (0) = ?	Attr Data (1) = ?	
Attr Data (2) = ?	(Pad)	

Figure 56 – Get Axis Attribute List Response – Example 2

EXAMPLE 3

Figure 57 illustrates a Get Axis Attribute List Response for six elements of a two dimensional array of REALs. The Attr Data sequences through the left-most array index first beginning with the Attr Start Index 1 of 0.

Get Axis Attribute List Response – 2 Dimensional Array Example		
Number of Attributes = 1		-
Attr ID = 27	Attr Dim = 2	Attr Elem Size = 4
Attr Start Index 1 = 0	Attr Data Elements 1 = 2	
Attr Start Index 2 = 2	Attr Data Elements 2 = 3	
Attr Data (0, 2) = ?		
Attr Data (0, 3) = ?		
Attr Data (0, 4) = ?		
Attr Data (1, 2) = ?		
Attr Data (1, 3) = ?		
Attr Data (1, 4) = ?		

Figure 57 – Get Axis Attribute List Response – Example 3

If there is an error associated with a specific requested get attribute operation, the device shall indicate this by setting the Attr Dimension to 0xFF, in which case the Element Size field may contain the specific error code. When an error code is present, neither the array index parameters nor the attribute data fields are returned.

EXAMPLE 4

Figure 58 illustrates such a case, where Attribute 29 is not supported by the targeted drive axis.

Get Axis Attribute List Response – Attribute Error Example		
Number of Attributes = 3	-	
Attr ID = 25	Attr Dim = 0	Attr Elem Size = 4
Attr Data = 43,5		
Attr ID = 29	Attr Dim = 0xFF	Error Code = 0x14
Attr ID = 43	Attr Dim = 1	Attr Elem Size = 2
Attr Start Index 1 = 0	Attr Data Elements 1 = 3	
Attr Data [0] = 55	Attr Data [1] = 45	
Attr Data [2] = 35	(Pad)	

Figure 58 – Get Axis Attribute List Response – Example 4

7.5.2.3 Set Axis Attribute List

The Set Axis Attribute List service provides a mechanism to write a value to one or more settable Motion Axis Object attributes, including attributes having a multi-dimensional array data type. The buffer/array transfer mechanism can be used to build parameter tables in the device for indexing, or camming applications.

The format of the Request Data Block for this service is shown in Figure 59.

← 32-bit Word →		
Set Axis Attribute List Request Format		
Number of Attributes	-	
Attr ID 1	Attr 1 Dimension	Attr 1 Element Size
Attr Start Index 1 (array only)	Attr Data Elements 1 (array only)	
...	...	
Attr Data 1		
...		
Attr ID 2	Attr 2 Dimension	Attr 2 Element Size
Attr Start Index 2 (array only)	Attr Data Elements 2 (array only)	
...	...	
Attr Data 2		
...		
Attr ID 3	Attr n Dimension	Attr n Element Size
Attr Start Index n (array only)	Attr Data Elements n (array only)	
...	...	
Attr Data 3		

Figure 59 – Set Axis Attribute List Request Format

Definitions of the individual parameters in this data structure are as follows:

- **Number of Attributes** – represents the number of attributes contained in the Set Axis Attribute service request.
- **Attr ID** – identifies the targeted Motion Axis Object configuration attribute to set.
- **Attr Dimension** – determines the dimension of the attribute array. A dimension of zero means the attribute is a singular data element and, therefore, not really an array at all. Multidimensional arrays (dimension > 1) are supported by adding additional Attr Start Index and Attr Data Elements values prior to the Attr Data sequence.
- **Attr Element Size** – determines the size, in bytes, of the attribute data element. Data elements shall be word aligned; 32-bit words are 32-bit aligned and 16-bit words are 16-bit aligned. Padding may be added to maintain word alignment.
- **Attr Start Index** – identifies the starting index for the array of attribute values in the Attr Data section. This field is only present when the attribute data type is an array (i.e., dimension > 0).
- **Attr Data Elements** – determines the number of data element values based on the start index that included in the Attr Data section. This field is only present when the attribute data type is an array (i.e., dimension > 0).
- **Attr Data** – contains the new value(s) that are to be applied to the targeted device configuration attribute indicated by the Attr ID. If the attribute is an array, the new value(s) are applied according to the Attr Start Index and the number of Attr Data Elements. For multidimensional arrays (dimension > 1), the list of Attr Data elements moves sequentially through the indices from left to right. The data type of the Attr Data shall match the data type of the attribute specified by the Attr ID.

EXAMPLE 1

Figure 60 illustrates a Set Axis Attribute List Request for a simple 4-byte scalar attribute. Since the Attr Dim is 0, the Attr Start Index and Attr Data Element fields are omitted.

Set Axis Attribute List Request – 0 Dimensional Data Example		
Number of Attributes = 1	-	
Write Attr ID = 25	Attr Dim = 0	Attr Elem Size = 4
Attr Data = ?		

Figure 60 – Set Axis Attribute List Request – Example 1

EXAMPLE 2

Figure 61 illustrates a Set Axis Attribute List Request for the first three elements of a one dimensional array of UINTs. Since the Attr Elem Size is 2-bytes, a Pad byte is added to maintain word alignment for any connection data to follow.

Set Axis Attribute List Request – 1 Dimensional Array Example		
Number of Attributes = 1	1	
Write Attr ID = 43	Attr Dim = 1	Attr Elem Size = 2
Attr Start Index 1 = 0	Attr Data Elements 1 = 3	
Attr Data (0) = ?	Attr Data (1) = ?	
Attr Data (2) = ?	(Pad)	

Figure 61 – Set Axis Attribute List Request – Example 2

EXAMPLE 3

Figure 62 illustrates a Set Axis Attribute List Request for six elements of a two dimensional array of REALs. The Attr Data sequences through the left-most array index first beginning with the Attr Start Index 1 of 0.

Set Axis Attribute List Request – 2 Dimensional Array Example		
Number of Attributes = 1	-	
Write Attr ID = 27	Attr Dim = 2	Attr Elem Size = 4
Attr Start Index 1 = 0	Attr Data Elements 1 = 2	
Attr Start Index 2 = 2	Attr Data Elements 2 = 3	
	Attr Data (0, 2) = ?	
	Attr Data (0, 3) = ?	
	Attr Data (0, 4) = ?	
	Attr Data (1, 2) = ?	
	Attr Data (1, 3) = ?	
	Attr Data (1, 4) = ?	

Figure 62 – Set Axis Attribute List Request – Example 3

The structure of Response Data Block for this service is as shown in Figure 63.

← 32-bit Word →		
Set Axis Attribute List Response Format		
Number of Attributes	-	
Attr ID 1	Attr Status 1	-
Attr ID 2	Attr Status 2	-
...		
Attr ID n	Attr Status n	-

Figure 63 – Set Axis Attribute List Response Format

- Attr ID – identifies the targeted Motion Axis Object configuration attribute of the set list request.

- **Attr Status** – indicates whether the set list action for the targeted attribute was successful. An Attribute Status value is zero if the targeted attribute was successfully written. A non-zero Attr Status value indicates that an error occurred that prevented the attribute from being updated. The error codes follow the CIP Common General Status Codes.

7.5.2.4 Set Cyclic Write List

The Set Cyclic Write List service provides a mechanism to determine the list of attributes to be passed as part of the Cyclic Write Data Block of the Controller-to-Device (C-to-D) Connection. The format of the Request Data Block for this service is shown in Figure 64.

← 32-bit Word →	
Set Cyclic Write List Request Format	
Number of Attributes	-
Attr ID 1	Attr ID 2
...	

Figure 64 – Set Cyclic Write List Request Format

- **Number of Attributes** – represents the number of attributes contained in the Cyclic Write List.
- **Attr ID** – identifies the specific axis configuration attribute that is to be updated via the Cyclic Write Data Block of the Controller-to-Device (C-to-D) Connection. The Attr ID determines the data type and implied semantics of the data based on the specifications for the associated attribute.

The ordering of the attribute data in the Cyclic Write Data Block is determined by the ordering of the Attr IDs in the Set Cyclic Write List request. Attribute data elements shall be word aligned; 32-bit words are 32-bit aligned and 16-bit words are 16-bit aligned. Padding may be added to maintain word alignment.

The structure of Response Data Block for this service is as shown in Figure 65.

← 32-bit Word →		
Set Cyclic Write List Response Format		
Number of Attributes	Cyclic Write Block ID	
Attr ID 1	Attr Status 1	-
Attr ID 2	Attr Status 2	-
...		
Attr ID n	Attr Status n	-

Figure 65 – Set Cyclic Write List Response Format

- **Number of Attributes** – represents the number of attributes contained in the requested Cyclic Write List.
- **Cyclic Write Block ID** – If all attributes in the requested list can be supported in the Set Cyclic Write Data Block, the Set Cyclic Write List response provides a new Cyclic Write Block ID that is simply the increment of the current Cyclic Write Block ID. This new Cyclic Write Block ID can be used in the next Controller-to-Device (C-to-D) Connection update. If the Set Cyclic Write List is not successful, the Cyclic Write Block ID remains the current value.
- **Attr ID** – identifies the specific axis configuration attribute that was requested to be updated via the Cyclic Write Data Block of the Controller-to-Device (C-to-D) Connection.
- **Attr Status** – The Attr Status value is zero if the associated attribute indicated by the Attr ID can be supported in the Set Cyclic Write Data Block. A non-zero Attr Status code indicates the specific reason why the associated attribute cannot be supported. These codes follow the CIP Common General Status Codes. If the Attr Status of one or more Attribute IDs in the list indicates an error, the Set Cyclic Write List request is considered unsuccessful.

7.5.2.5 Set Cyclic Read List

The Set Cyclic Read List service provides a mechanism to determine the list of attributes to be passed as part of the Cyclic Read Data Block of the Device-to-Controller (D-to-C) Connection.

The format of the Request Data Block for this service is shown in Figure 66.

← 32-bit Word →	
Set Cyclic Read List Request Format	
Number of Attributes	-
Attr ID 1	Attr ID 2
...	

Figure 66 – Set Cyclic Read List Request Format

- Number of Attributes – represents the number of attributes contained in the Cyclic Read List.
- Attr ID – identifies the specific axis status or signal attribute that is to be updated via the Cyclic Read Data Block of the Device-to-Controller (D-to-C) Connection. The Attr ID determines the data type and implied semantics of the data based on the specifications for the associated attribute.

The ordering of the attribute data in the Cyclic Read Data Block is determined by the ordering of the Attr IDs in the Set Cyclic Read List request. Attribute data elements shall be word aligned; 32-bit words are 32-bit aligned and 16-bit words are 16-bit aligned. Padding may be added to maintain word alignment.

The structure of Response Data Block for this service is as shown in Figure 67.

← 32-bit Word →		
Set Cyclic Read List Response Format		
Number of Attributes	Cyclic Read Block ID	
Attr ID 1	Attr Status 1	-
Attr ID 2	Attr Status 2	-
...		
Attr ID n	Attr Status n	-

Figure 67 – Set Cyclic Read List Response Format

- Number of Attributes – represents the number of attributes contained in the requested Cyclic Read List.
- Cyclic Read Block ID – If all attributes in the requested list can be supported in the Set Cyclic Read Data Block, the Set Cyclic Read List response provides a new Cyclic Read Block ID that is simply the increment of the current Cyclic Read Block ID. This new Cyclic Read Block ID can be used in the next Device-to-Controller (D-to-C) Connection update. If the Set Cyclic read List is not successful, the Cyclic Read Block ID remains the current value.
- Attr ID – identifies the specific device configuration attribute that was requested to be updated via the Cyclic Read Data Block of the Device-to-Controller (D-to-C) Connection.
- Attr Status – The Attr Status value is zero if the associated attribute indicated by the Attr ID can be supported in the Set Cyclic Read Data Block. A non-zero Attr Status code indicates the specific reason why the associated attribute cannot be supported. These codes follow the CIP Common General Status Codes. If the Attr Status of one or more Attribute IDs in the list indicates an error, the Set Cyclic Read List request is considered unsuccessful.

7.5.2.6 Run Motor Test

The Run Motor Test service initiates various test operations motor to determine important motor parameters. The Static Test applies a directionally static flux to the motor, while the Dynamic Test performs dynamic tests on the motor that produce motion (see Table 84). The parameters measured by these tests are retrieved by the controller via the Get Motor Test Data command.

Table 84 – Run Motor Test Request Data Structure

Name	Type	Description of Request Parameter	Semantics of Values
Test Direction	USINT	Enumeration that selects test direction:	Enumeration: 0 = forward 1 = reverse 2 = forward bi-directional 3 = reverse bi-direction
Test Type	USINT	Enumeration that specifies the type of test to perform:	Enumeration: 0 = static test 1 = dynamic test

There are no specific response parameters required for this service.

7.5.2.7 Get Motor Test Data

The Get Motor Test Data service provides access to the motor parameter measurements generated by the last Motor Test command (see Table 85). The controller uses this data to update the device with the best estimates of critical motor parameters.

There are no specific request parameters required for this service.

Table 85 – Get Motor Test Response Data Structure

Name	Type	Description of Response Parameter	Semantics of Values
Test Status	USINT	Indicates the result of the last Motor Test command.	Enumeration: 0 = test process successful 1 = test in progress 2 = test process aborted 3 = test process timed-out 4 = test process faulted
Stator Resistance	REAL	Measured phase-to-phase stator resistance of the motor.	Ohms
Leakage Inductance	REAL	Measured phase-to-phase leakage inductance of the motor.	Henries
Flux Current	REAL	Measured current to achieve full induction motor flux. (IM only)	Amperes
Slip Frequency	REAL	Measured slip at rated induction motor current. (IM only)	Hertz

7.5.2.8 Run Inertia Test

The Run Inertia Test service performs an acceleration and deceleration ramp on the axis and makes timing measurements in the process (see Table 86). The timing measurements are accessed by the controller via the Get Inertia Test Data command and ultimately used to calculate an accurate inertia value for the motor and load.

Table 86 – Run Inertia Test Request Data Structure

Name	Type	Description of Request Parameter	Semantics of Values
Test Direction	USINT	Enumeration that selects test direction.	Enumeration: 0 = forward 1 = reverse
Test Velocity	REAL	Determines the maximum velocity that will be reached during the test profile.	Velocity Feedback Counts/s
Test Torque	REAL	Determines the maximum torque that is applied during the test profile.	% Motor Rated
Test Travel Limit	REAL	Establishes the maximum excursion of the axis in Feedback Counts allowed in the test direction.	Feedback Counts

There are no specific response parameters required for this service.

7.5.2.9 Get Inertia Test Data

The Get Inertia Test Data service provides access to the timing measurements generated by the last Run Inertia Test service (see Table 87). The controller uses this data to ultimately calculate an accurate inertia value for the motor and load.

There are no specific request parameters required for this service.

Table 87 – Get Inertia Test Response Data Structure

Name	Type	Description of Response Parameter	Semantics of Values
Test Status	USINT	Enumeration that indicates the result of the last Run Inertia Test service:	Enumeration: 0 = test process successful 1 = test in progress 2 = test process aborted 3 = test process timed-out 4 = test process fault 5 = test reached limit 6 = test polarity incorrect 7 = test speed set too low
Accel Time	REAL	Measured acceleration time to reach test velocity.	Seconds
Decel Time	REAL	Measured deceleration time to reach zero speed from test velocity.	Seconds

7.5.2.10 Run Hookup Test

The Run Hookup Test service performs a number of different tests to check for proper interface to the motor and/or feedback device. Three tests are currently supported (see Table 88). The Motor/Feedback test is an active test that attempts to move the axis a specified distance given by the Test Increment. The Feedback test monitors the axis while an external agent moves it and indicates success if the axis feedback count exceeds the distance specified by the Test Increment. Finally, the Commutation Test, which applies only to PM motors, applies current to the motor to align the rotor, check for proper phasing of a Hall sensor if applicable, and measure the Commutation Offset. The resultant data generated by these tests can be accessed by the Get Hookup Test Data service and used by the controller to automatically set the proper polarities of the feedback interface and the motor interface.

Table 88 – Run Hookup Test Request Data Structure

Name	Type	Description of Request Parameter	Semantics of Values
Test Increment	REAL	Establishes the distance that the axis needs to travel, in Feedback Counts to indicate a successful test.	Feedback Counts
Hookup Test	USINT	Determines the specific test to run:	Enumeration: 0 = motor/feedback test 1 = feedback test 2 = commutation test

There are no specific response parameters required for this service request.

7.5.2.11 Get Hookup Test Data

The Get Hookup Test Data service provides access to the hookup test results generated by the last Run Hookup Test service (see Table 89). The controller uses this data to flag a wiring problem to the user and to calculate polarity configuration bit parameters for the axis.

There are no specific request parameters required for this service.

Table 89 – Get Hookup Test Response Data Structure

Name	Type	Description of Response Parameter	Semantics of Values
Test Status	USINT	Enumeration that indicates the result of the last Hookup Test command:	Enumeration: 0 = test process successful 1 = test in progress 2 = test process aborted 3 = test process timed-out 4 = test process faulted
Test Direction Forward	USINT	Enumeration that indicates the direction of axis motion associated with the last Hookup Test command.	Enumeration: 0 = forward 1 = reverse
Hall Commutation Map	USINT	When performing a Commutation Test on a PM motor feedback device, the test checks if the UVW phases of the Hall Sensor match the UVW phases of the Motor. If it is determined that the UVW phases for the motor and hall device do not match, typically due to incorrect wiring, this parameter reports the necessary commutation map can be used to compensate the mismatch. The enumerated choices listed assume that the motor is wired correctly as UVW and describes different mappings of the UVW Hall sensor inputs, which if applied to the device, would correct the phasing.	Enumeration: 0 = UVW (normal) 1 = UWV 2 = WVU 3 = WUV 4 = VUW 5 = VWU
Commutation Offset	REAL	Offset measured during the Commutation Test that shall be applied to the motor position accumulator in order to align the Electrical Angle signal with motor stator's orientation.	Electrical Degrees

7.6 Behaviour

7.6.1 State model

7.6.1.1 General

The Motion Axis Object State Model is based on elements of the S88 and Pack/ML standard state models. The current state of the Motion Axis Object instance is indicated by the Axis

State attribute. State transitions can be initiated either directly via the Axis Control request mechanism or by conditions that occur during device operation. Figure 68 shows the basic operating states of the Motion Axis Object.

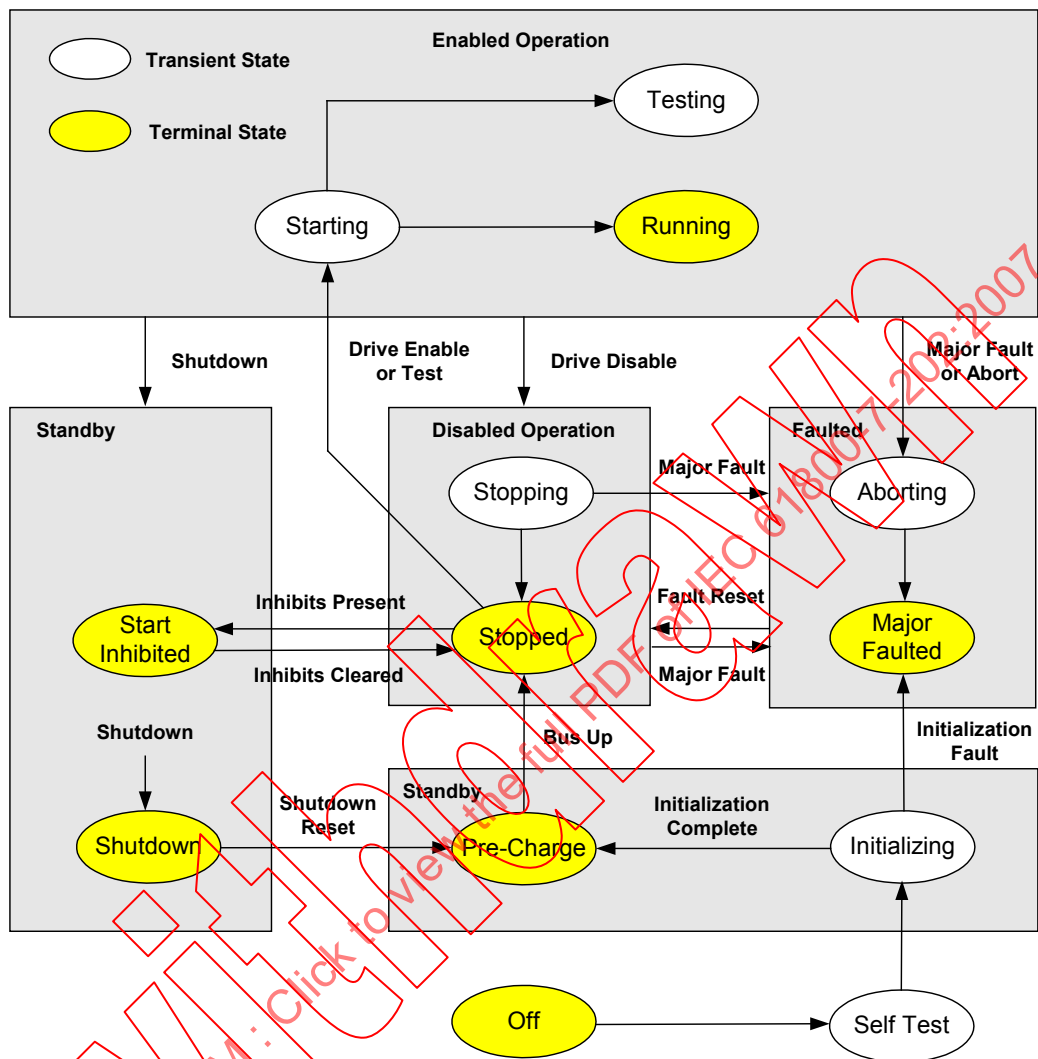


Figure 68 – Motion Axis Object State Model

7.6.1.2 State Control

The primary method for changing the state of a Motion Axis Object instance is via the Axis Control/Axis Response mechanism that is built into the CIP Motion I/O Connection header. Changing the state of the motion axis is simply performed by placing the appropriate Request Code in the Axis Control element of the Controller-to-Device (C-to-D) Connection header. The Control Request codes currently defined in this object are as specified in Table 90.

Table 90 – Axis Control Request Code

Request Code	Requested Operation
0	No Request
1	Enable Request
2	Disable Request
3	Shutdown Request
4	Shutdown Reset Request
5	Abort Request
6	Fault Reset Request

Request Code	Requested Operation
7 to 127	(Reserved)
128 to 255	(Vendor Specific)

When a state transition is requested via Axis Control element of the Controller-to-Device (C-to-D) Connection, the device initiates the state transition and then acknowledges the transition request via the Axis Response attribute when the requested state transition completes or is determined unsuccessful. The Acknowledge Codes for the Axis Response element are the same as the Request Codes for the Axis Control element (see Table 91).

Table 91 – Axis Response Acknowledge Codes

Acknowledge Code	Axis Response
0	No Acknowledge
1	Enable Acknowledge
2	Disable Acknowledge
3	Shutdown Acknowledge
4	Shutdown Reset Acknowledge
5	Abort Acknowledge
6	Fault Reset Acknowledge
7 to 127	(Reserved)
128 to 255	(Vendor Specific)

The criteria for successful completion of the requested operation depend on the specific Request Code as shown in Table 92.

Table 92 – Completion Criteria for Requested Operation

Request Code	Completion Criteria
1	Axis successfully transitions to Starting state.
2	Axis successfully transitions to Stopped state.
3	Axis successfully transitions to Shutdown state.
4	Axis successfully transitions to Stopped or Inhibited state.
5	Axis successfully transitions to Faulted state.
6	Axis successfully transitions to Stopped, Inhibited, or Shutdown state.
7 to 127	(Reserved)
128 to 255	(Vendor Specific)

When the controller receives the Acknowledge Code, it then zeroes the matching Request Code. When the device sees the Request Code zeroed, it clears the Acknowledge Code and the associated Response Status.

If the axis cannot transition to the requested state, the specific error condition is indicated by a non-zero Response Status attribute value that is passed along with the Axis Response at part of the Device-to-Controller (D-to-C) Connection.

In the case of a Fault Reset, if any fault generating exception conditions still exist, an Error Code is generated to indicate that the axis cannot transition out of the Faulted State until the associated exception conditions have been cleared.

This behavior is illustrated in Table 93 and Table 94.

Table 93 – Successful Axis Control Request Cycle

Controller Action	Connection Data	Motion Device Action
Controller sets Request Code in the Axis Control element to request a state change.	→ Axis Control = x	Motion Device sees non-zero Request Code, x, in Axis Control element and initiates the requested state change.
	← Axis Response = x ← Response Status = 0	If state change operation is successful, the device acknowledges the state change by setting the Acknowledge Code in the Axis Response element to the originating Request Code, x. The Response Status element is set to 0 indicating success.
Controller sees the non-zero Acknowledge Code in Axis Response element and zeroes the originating Request Code in the Axis Control element. The requested state change was successful as evidenced by the zero Response Status.	→ Axis Control = 0	
The state change request transaction is now complete. Another Axis Control request can be processed at this time.	← Axis Response = 0 ← Response Status = 0	Motion Device sees a zero Request Code in the Axis Control element and zeroes the associated Acknowledge Code in the Axis Response element.

Table 94 – Unsuccessful Axis Control Request Cycle

Controller Action	Connection	Motion Device Action
Controller sets Request Code in the Axis Control element to request a state change.	→ Axis Control = x	Motion Device sees non-zero Request Code, x, in Axis Control element and initiates the requested state change.
	← Axis Response = x ← Response Status = Error Code	If state change operation is unsuccessful, the device indicates the problem via a non-zero Response Status Error Code along with the Acknowledge Code matching the originating Request Code, x.
Controller sees the non-zero Response Status associated with the Request Code, x, in the Axis Response, handles the error condition based on the Error Code, and clears the originating Request Code in the Axis Control element.	→ Axis Control = 0	
The state change request transaction is now complete. Another Axis Control request can be processed at this time.	← Axis Response = 0 ← Response Status = 0	Motion Device sees a zero Request Code in the Axis Control element and zeroes the associated Acknowledge Code in the Axis Response element.

Object state transitions can also be initiated by service requests such as the Test service requests. In general, these state change requests use services because they require one or more parameters to be passed to the device in order to initiate the state change.

7.6.2 State behaviour

7.6.2.1 General

The following subclause offers a detailed description of each of the states and state transitions of the Motion Axis Object state model.

7.6.2.2 Off state

This is the state of the Motion Axis Object with power off.

7.6.2.3 Self Test State

When power is applied to the device, or the device is reset, the device typically goes through a series of self-test diagnostics and internal device parameters are set to their power-up default values. Once completed successfully, the device and all its associated axis instances transition to the Initializing state and are ready for initialization by the associated controller. If unsuccessful, the device and all its associated axis instances transition immediately to the Major Faulted state by declaring an Initialization Fault that is classified according to CIP Common as Unrecoverable. Clearing this fault can only be accomplished through a power cycle and is most likely the result of a device hardware problem.

7.6.2.4 Initialising state

During the Initializing state, network connections to the device are established by the controller. Once connections are established, the controller sends services to the device to set the Motion Axis Object configuration attributes to values stored in the controller. If the device supports synchronous operation, the controller then synchronizes with the device using the Group Sync service. Once this entire process has been completed successfully, the device and all its configured axis instances transition to the Pre-charge state. If a problem is found during this initialization process, an Initialization Fault is generated. An Initialization Fault is viewed as an unrecoverable fault so clearing the fault can only be accomplished through a power cycle or a device reset service to the associated Identity Object.

7.6.2.5 Pre-Charge state

In the Pre-Charge state, the device is waiting for the DC Bus to fully charge. Once the DC Bus reaches an operational voltage level, as indicated by the DC Bus Up bit in the Axis Status attribute, the device and all its configured axis instances transition to the Stopped state. Any attempt for the controller to enable an axis via the Axis Control mechanism while it is in the Pre-charge state is reported back to the controller as an error in the Response Status and the axis remains in the Pre-charge state.

7.6.2.6 Stopped state

In the Stopped state, the device's inverter power structure should either be disabled and free of torque or held in a static condition via an active control loop. No motion can be initiated by the device in the Stopped State nor can the device respond to a planner generated command references. In general, the axis should be at rest, but if an external force or torque is applied to the load, a brake may be needed to maintain the rest condition. In the Stopped state, main power is applied to the device and the DC Bus is at an operational voltage level. If there are any Start Inhibit conditions detected while in this state, the axis transitions to the Start Inhibited state. If an Enable request or one of the Run Test service requests is applied to an axis in the Stopped state, the axis transitions to the Starting state.

7.6.2.7 Starting state

When an Enable request is given to an axis in the Stopped state, the axis immediately transitions to the Starting state. In this state, the device checks various conditions before transitioning the axis to the Running state. These conditions can include Brake Release delay time and Induction Motor flux level. The control and power structures are activated during the Starting state but the command reference is set to a local static value and does not track the command reference derived from the motion planner. If all the starting conditions are met, the axis state transitions to either the Running state or the Testing state.

7.6.2.8 Running state

The Running state is where the work gets done. In this state, the axis's power structure is active and the selected Control Mode is enabled and actively tracking command data from the controller based motion planner output to affect axis motion. The motion axis remains in the Running state until either a fault occurs or it is explicitly commanded to stop via a Axis Control request.

7.6.2.9 Testing state

When any one of the Run Test request services is sent to the motion axis while in the Stopped state, the axis immediately transitions to the Starting State, and then once the Starting conditions are met, the axis transitions to the Testing state. This Testing state is like the Running state in that the axis's power structure is active, but in the Testing state one of the axis's built-in test algorithms is controlling the motor, not a motion planner. In the Testing state, the device excites the motor in various ways while performing measurements to determine characteristics of the motor and load. The motion axis remains in this state for the duration of the requested test procedure and then returns to the Stopped state. The motion axis can also exit the Testing state by either a fault or an explicit Axis Control request.

7.6.2.10 Start Inhibited state

The Start Inhibited state is the same as the Stopped state with the exception that the axis has one or more "start inhibit" conditions that prevent it from successfully transitioning to the Starting state. These conditions can be found in the Start Inhibit Status attribute. Once corrected, the axis state automatically transitions back to the Stopped state.

7.6.2.11 Stopping state

When a Disable request is issued to the Motion Axis Object in the Running or the Testing state, the axis immediately transitions to the Stopping state. In this state, the axis is in the process of stopping. There are a number of different Stopping Modes supported by the Motion Axis Object. Once the selected stop mode procedure has completed, the axis transitions to the Stopped state.

7.6.2.12 Aborting state

When a Major Fault occurs while the axis is in either the Running or Testing states, the motion axis immediately transitions to the Aborting state. The Aborting state executes the appropriate stopping action as specified by the device vendor. Once the stopping procedure is complete, the axis transitions to the Major Faulted state.

7.6.2.13 Major Faulted state

The Major Faulted state is identical to the Stopped state (or, if a Shutdown fault action was initiated, the Shutdown state) with the exception that there are one or more Major Faults active. In other words, a Major Faulted axis is a Stopped (or Shutdown) axis with a Major Fault condition present. Since faults are latched conditions, a Fault Reset request from the controller is required to clear the fault and, assuming the original fault condition has been removed, the axis transitions to the Stopped (or Shutdown) state.

7.6.2.14 Shutdown state

When a Shutdown request is issued to the device or a Shutdown fault action is executed by the device, the targeted axis transitions to the Shutdown state. In the case of a Shutdown request, the axis immediately transitions from whatever state it is currently in to the Shutdown state. The Shutdown state has the same basic characteristics of the Stopped state except that it can be configured to drop the DC Bus power to the device's power structure. Regardless of whether or not DC Bus power is disconnected, this state requires an explicit Shutdown Reset request from the controller to transition to the Pre-Charge state. If the axis is configured to keep the DC Bus power active while in the Shutdown state then the motion axis transitions

through the Pre-Charge state to the Stopped state. The Shutdown state offers an extra level of safety against unexpected motion.

7.6.3 Fault and alarm behaviour

7.6.3.1 General

The Motion Axis Object's Fault and Alarm handling functionality addresses both the need for a large and ever-expanding number of specific faults and alarms, the need for programmable actions, and the need for timely reporting of those faults and alarms to the controller. Additionally, no compromises are made to restrict the resolution of the reported faults and alarms, so that the controller always has access to the unique axis condition and a meaningful diagnosis. Numerous Fault and Alarm related attributes can be included in the fixed portion of the cyclic Device-to-Controller (D-to-C) Connection so the controller can monitor the condition of the motion axis in real-time, without cumbersome polling.

The Axis Status attribute contains bits to indicate whether an alarm condition is present. The Axis State enumeration indicates when the axis has a major fault, which could be either a regular runtime Axis Fault or an Initialization Fault. Fault Codes are also provided to report the specific fault condition that led to the current Faulted state and covers both major and minor Faults. But before going into detail on this, the terms used to describe the Fault and Alarm functionality of the Motion Axis Object need to be carefully defined.

7.6.3.2 Exceptions

Exceptions are runtime conditions that the device continually checks that might indicate improper behavior of the motion axis or operation outside of an allowable range. An exception can result in an alarm, a minor fault, or a major fault, depending on how the associated Exception Action has been configured – an exception can even be configured to be ignored. Exceptions are automatically cleared by the device when the underlying exception condition is no longer present.

7.6.3.3 Exception actions

For each exception, the motion axis can be programmed a variety of actions via the Exception Action attribute. Exception Actions range from generating a major fault that results in the stopping of the motion axis all the way to taking no action at all. The Axis Fault Status attribute allows the controller to have immediate access to any exceptions that have been configured to generate a major or minor fault. The Axis Alarm Status attribute allows the controller to have immediate access to any exceptions that have been configured to be reported as alarms.

7.6.3.4 Alarms

Alarms are runtime exception conditions for which the device is to take no action other than to report as an alarm. Alarms and warnings, therefore, are basically synonymous. On a given device product, some exception conditions may not be able to simply be reported as an alarm without any associated action; for example an IPM fault in which the power module automatically shuts off without software intervention. Alarm conditions are automatically cleared when the underlying exception condition is no longer present.

7.6.3.5 Major faults

Major Faults are either initialization faults or runtime exception conditions that the device has been configured to regard as a major fault. If such a runtime fault occurs during an operational state, for example Running or Testing, it results in the device stopping (or aborting) all axis motion. Major Faults ultimately transition the axis state to the Major Faulted state. A Major Fault that results from an exception condition is latched, and does not clear when the exception condition clears. A fault can only be cleared with a Fault Reset service request from the controller. If the fault condition is classified as an "unrecoverable fault", only a power cycle or a device reset can clear the fault condition.

7.6.3.6 Minor faults

Minor Faults are exception conditions that the device has been configured to report to the controller as a fault but not take any direct action. This provides the controller with an opportunity to perform an application specific fault action that may not be supported by the device as one of the defined exception actions. Like alarms, Minor Faults do not initiate a state change nor does a Minor Faulted state even exist; a Minor Fault allows the motion axis to continue operation in the state that it is presently in. But unlike alarms, a Minor Fault is latched, i.e. the fault does not clear when the exception condition clears. Both Major and Minor Faults can only be cleared with a Fault Reset service request from the controller. Since the motion axis cannot transition from the Stopped state to any operational state until the Minor Fault is cleared, a Minor Fault is considered a Start Inhibit and is indicated by the Minor Fault bit being set in the Start Inhibit Status attribute.

7.6.3.7 Initialisation faults

Initialization Faults are faults that are generated during the power-up or device reset procedure when the device detects a problem that prevents normal device operation. This could be a hardware or firmware problem detected as part of its self diagnostic tests or a problem with the attribute configuration process. These faults are not sourced by exception conditions and, therefore, they do not have configurable actions. Examples of initialization faults are corrupted memory data, calibration errors, or firmware startup problems. Initialization Faults that result in the Faulted state cannot be cleared with a Fault Reset service, so any kind of motion is impossible in this state; only a power-cycle or a Device Reset has a chance of clearing this kind of fault.

7.6.3.8 Fault codes

The Fault Code attribute is an enumeration that indicates the exception condition that generated the fault. This attribute does not change unless a Fault Reset service is initiated. The Initialization Fault Code attribute is an enumeration that indicates the initialization fault that led to the current Faulted state.

7.6.4 Start Inhibit behaviour

A Start Inhibit is a condition that inhibits the axis from starting, i.e. transitioning to the Starting state for enabled axis operation. This condition does not generate an exception if a start attempt is made. If the circumstances that led to the Start Inhibit are no longer present, the start inhibit condition is automatically cleared by the device, returning the axis to the Stopped State.

If the motion axis is in the Start Inhibit state, it indicates that one or more conditions are present that prevent the axis from transitioning to enabled operation. The Start Inhibit Status attribute reports the specific condition that is inhibiting the axis.

The Start Inhibit Code attribute is an enumeration that indicates a condition that is presently causing the axis to be in the Start Inhibit state. If the original inhibiting condition is removed, this attribute automatically cleared or else set to another inhibiting condition if one is present.

7.6.5 Visualization behaviour

7.6.5.1 General

Motion Axis Object state behavior has a direct impact on motion device visualization components. These components range from bicolor LEDs to multi-character alphanumeric displays. This section defines how the Motion Axis Object states affect the behavior of these visualization components.

7.6.5.2 Module Status LED

Motion Axis Object states have a relationship to the state behavior of the Identity Object of the CIP Motion device and to its associated Module Status LED. Table 95 maps the states of the primary Motion Axis Object instance to the appropriate states of the Identity Object. According to CIP Common practices, all CIP Motion compliant devices are required to support a Module Status LED.

For more information regarding the Identity Object state model, refer to the Identity Object specification (see IEC 61158–5-2).

7.6.5.3 Axis Status LED

To further augment the visual information provided by the standard Module Status LED, the CIP Motion device profile also defines the behavior of a second LED. This, so-called Axis Status LED provides visual indication of, for example, whether or not the DC Bus is energized, whether the axis is enabled or disabled, and even provides indication of active alarms or minor fault conditions (see Table 95). This LED is also required by CIP Motion compliant devices unless the device is equipped with a multi-character alphanumeric display.

The Axis Status LED uses three colors that can be generated by a standard bicolor Red/Green LED, namely Red, Green, and Amber. Amber (or Yellow) is the color produced when both the Red and Green junctions of the bicolor LED are on. The general meaning of these three colors are as follows:

Green – indicates a normal power-up or operational state;

Amber – indicates the presence of an alarm or start inhibiting condition;

Red – indicates presence of some form of fault condition.

When both minor fault and alarm conditions are present, the minor fault indication takes precedence.

The normal power up or device reset, both the Module Status and Axis Status LEDs shall start in the Red state (under hardware control) while the device processor is booting and then switch to the Green state for approximately 1 s once the device begins executing its self test. Ideally, both LEDs should be Red for approximately 1 s and then Green for approximately 1 s prior to transitioning to a Standby state indication. This Red-Green sequence confirms proper operation of the LED indicators.

Table 95 – Axis State Mapping to Identity Object with LED Behavior

Identity Object State	Module Status LED	Motion Axis Object State	Axis Status LED
Nonexistent – Power Off	Off	Off	Off
Device Self-Testing	Flash Red/Green	Self Test	Flash Red/Green
Standby	Flashing Green	Initialization – Bus not Up	Off
		Initialization – Bus Up	Flashing Green
		Shutdown – Bus not Up	Off
		Shutdown – Bus Up	Flashing Amber ^a
		Pre-Charge - Bus not Up	Off
		Start Inhibit	Flashing Amber ^a
Operational	Solid Green	Stopped	Flashing Green ^{a,b}
		Stopping	Solid Green ^{a,b}
		Starting	Solid Green ^{a,b}
		Running	Solid Green ^{a,b}
		Testing	Solid Green ^{a,b}

Identity Object State	Module Status LED	Motion Axis Object State	Axis Status LED
Major Recoverable Fault	Flashing Red	Aborting	Flashing Red
		Major Faulted	Flashing Red
Major Unrecoverable Fault	Solid Red	Aborting	Solid Red
		Major Faulted	Solid Red
^a The Motion Axis Object and the Identity Object define minor fault conditions. While a minor fault does not affect the Module Status LED, it does affect the Axis Status LED. When a minor fault condition is detected, a normally Solid Green LED indication changes to alternating Red-Green-Red-Green, a normally Flashing Green LED indication changes to alternating Red-Off-Green-Off, and a normally Flashing Amber indications changes to Red-Off-Amber-Off. ^b The Motion Axis Object also defines alarm conditions. When an alarm condition is detected, a normally Solid Green LED indication changes to alternating Amber-Green-Amber-Green, while a normally Flashing Green LED indication changes to alternating Amber-Off-Green-Off.			

7.6.5.4 Alphanumeric Display

7.6.5.4.1 General

In addition to the required LED visualization provided by the Module Status LED and Axis Status LED, the Motion Axis Object also defines the behavior of an optional alphanumeric display to more explicitly indicate the condition of the device. Such a display is particularly useful for monitoring progress through the initialization process and providing detailed diagnostic information concerning fault, alarm, and inhibit conditions. Alphanumeric displays can range from simple seven-segment displays to multi-character alphanumeric displays.

7.6.5.4.2 Seven-Segment Display

If the device is equipped with a seven-segment display, the display can be used to indicate progress through the Initialization state and various fault, alarm, and inhibit conditions. As a minimum, the display should support the mapping to various conditions of the device specified in Table 96.

Table 96 – CIP Motion Device Seven-Segment Display Behavior

Display Digit	Device Condition
8	Executing device Self-Test
0	Waiting for connection to controller
1	Configuring device attributes
2	Waiting for group synchronization
3	Waiting for DC Bus to charge
4	Device is operational
I##	Initialization Fault Code
IE##	Initialization Fault Code - Manufacturer extensions
F##	Axis Fault Code
FE##	Axis Fault Code - Manufacturer extensions
A##	Axis Alarm Code
AE##	Axis Alarm Code - Manufacturer extensions
S##	Start Inhibit Code
SE##	Start Inhibit Code - Manufacturer extensions

7.6.5.4.3 Multi-Character Alphanumeric Display

If the device is equipped with a multi-character alphanumeric display, even more useful device information can be conveyed to the user via scrolling or static character fields. The

capabilities of the display generally dictate the length of the character strings that can be effectively displayed. Therefore, the CIP Motion device profile does not dictate exactly what is displayed. Nevertheless, for the purpose of enforcing consistent behavior among CIP Motion compliant device products, the Motion Axis Object dictates exactly what is displayed for the conditions outlined in Table 97.

Table 97 – CIP Motion Multi-Character Alphanumeric Display Behavior

Display String	Device Condition
SELF-TEST	Executing device Self-Test
CONNECTING	Connecting (waiting for Fwd Open)
CONFIGURING	Configuring Motion Device Attributes
SYNCING	Synchronizing (waiting for Group Sync)
PRE-CHARGE	Waiting for DC Bus Up
SHUTDOWN	Axis has been Shutdown
STOPPED	Axis has stopped
INHIBITED	Axis is Start Inhibited
STARTING	Axis is Starting
RUNNING	Axis is Running
TESTING	Axis is executing a Test procedure
STOPPING	Decelerating to a stop as a result of a disable
ABORTING	Decelerating to a stop as a result of a fault
INIT FLT S##	Initialization Fault – Std and Fault Code
INIT FLT M##	Initialization Fault – Mfg and Fault Code
FLT S##	Axis Fault – Std and Fault Code
FLT M##	Axis Fault – Mfg and Fault Code
ALARM S##	Axis Alarm – Std and Fault Code
ALARM M##	Axis Alarm – Mfg and Fault Code
INHIBIT S##	Start Inhibit – Std and Fault Code
INHIBIT M##	Start Inhibit – Mfg and Fault Code

In addition to this basic functionality, the alphanumeric display can also provide detailed text descriptions of fault, alarm, and inhibit conditions. Definition of these specific strings is well beyond the scope of this object and is left to the discretion of the device vendor.

7.6.5.4.4 Multi-Axis Device Visualization

In the case where there are multiple motion axis instances supported by the device, the above behavior needs further explanation.

First of all, there is only one Module Status LED per device node, so its condition is a roll-up of the states of all the Motion Axis Object instances. By contrast, one Axis Status LED is associated with each Motion Axis Object instance in the motion device that has a power structure. A single multi-character alphanumeric display can easily manage multiple motion axis instances.

In the case of a motion axis instance configured as a Master Feedback, there is very little state information to visualize; the motion axis is in the Stand-by state called Stopped and remains in that state unless there is a feedback fault that transitions it to Major Faulted. A Master Feedback axis, therefore, has no affect on the Module Status LED unless a fault occurs, in which case the Module Status shows Flashing Red (Major Recoverable Fault). A

Master Feedback axis has no effect on the Axis Status LED nor is a separate LED required for such an axis.

In the case of multiple motion axis instances, each with a separate power structure, the behavior of the Module Status LED is a roll-up of the states of the device axes. When the device axes are in disparate states, the Module Status LED condition is based on the following precedence.

- 1) Major Unrecoverable Fault
- 2) Major Recoverable Fault
- 3) Standby
- 4) Operational

In other words, as far as the Module Status LED is concerned, a Major Recoverable Fault on Axis 1 trumps Axis 2 that is in the Standby state, Start Inhibit.

NOTE Minor faults and alarms are not recognized by the Module Status LED.

Multi-character displays easily handle multiple axis instances by adding a "X#" prefix to the display string to specify the associated motion axis instance number. For Master Feedback axis instances, no specific display string is shown unless the Master Axis has a fault or alarm condition. Table 98 specifies the mapping between the display and the device condition in the case of multiple motion axis instances.

Table 98 – Multi-Axis Multi-Character Alphanumeric Display Behavior

Display String	Device Condition
SELF-TEST	Executing device Self-Test
CONNECTING	Connecting (waiting for Fwd Open)
CONFIGURING	Configuring Motion Device Attributes
SYNCING	Synchronizing (waiting for Group Sync)
PRE-CHARGE	Waiting for DC Bus Up
X#:SHUTDOWN	Axis has been Shutdown
X#:STOPPED	Axis has stopped
X#:INHIBITED	Axis is Start Inhibited
X#:STARTING	Axis is Starting
X#:RUNNING	Axis is Running
X#:TESTING	Axis is executing a Test procedure
X#:STOPPING	Decelerating to a stop as a result of a disable
X#:ABORTING	Decelerating to a stop as a result of a fault
X#:INIT FLT S##	Initialization Fault – Std and Fault Code
X#:INIT FLT M##	Initialization Fault – Mfg and Fault Code
X#:FLT S##	Fault – Std. and Fault Code
X#:FLT M##	Fault – Mfg and Fault Code
X#:ALARM S##	Initialization Fault – Std and Fault Code
X#:ALARM M##	Initialization Fault – Mfg and Fault Code
X#:INHIBIT S##	Initialization Fault – Std and Fault Code
X#:INHIBIT M##	Initialization Fault – Mfg and Fault Code
X# = Axis Instance Number (0, 1, 2, 3 ...) → 0 = class instance	

7.6.6 Command generation behaviour

7.6.6.1 Command data sources

Command data can take the form of Controller Position, Velocity, Acceleration, and Torque Commands (see Figure 69). The command data element provided is specified by the Command Data Set attribute, which is based on the selected Control Mode. The primary command data element can be augmented by higher order command elements for the purposes of generating high quality feed-forward signals.

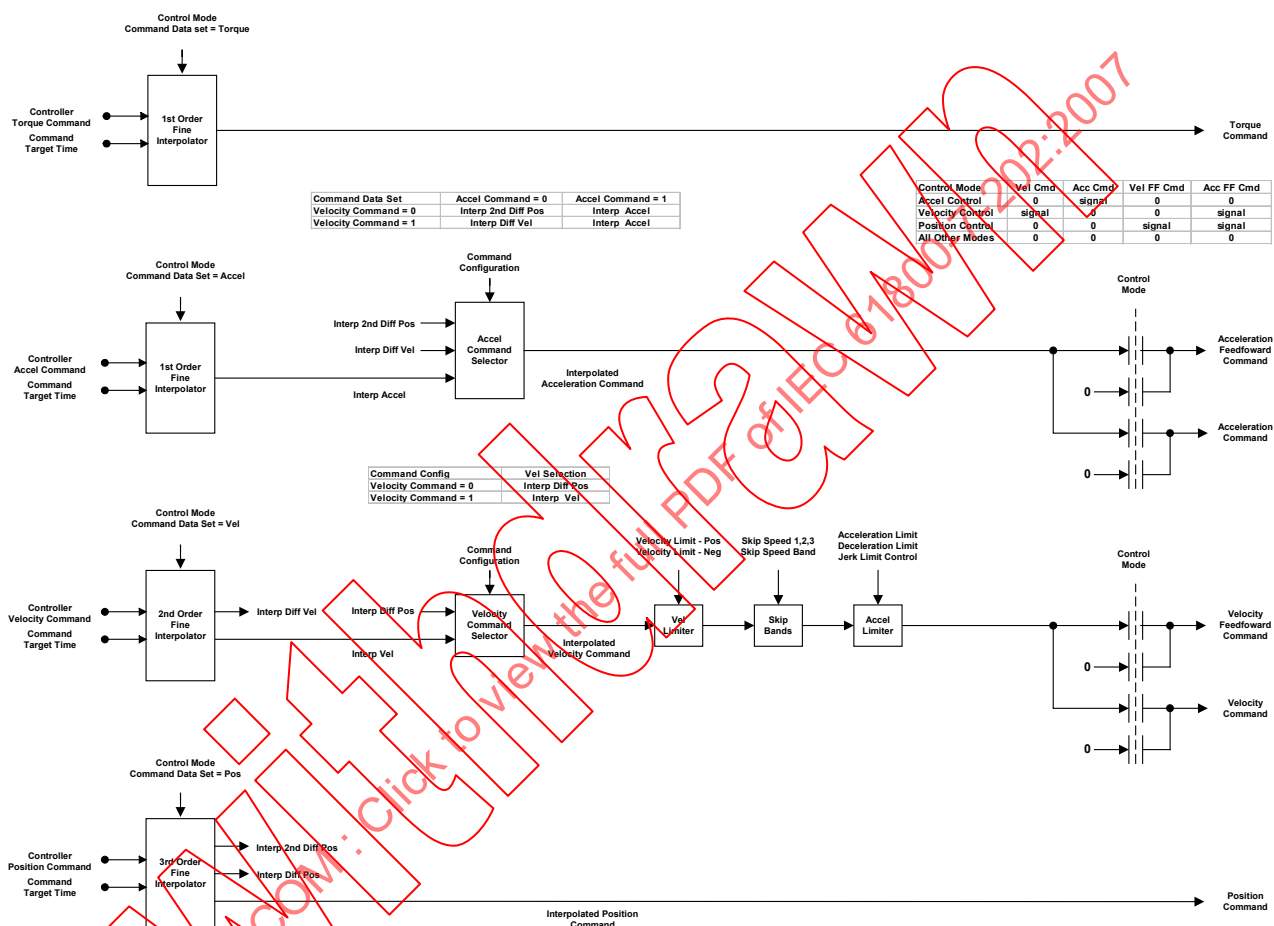


Figure 69 – Command Generator

7.6.6.2 Command fine interpolation

For synchronized, high-performance applications using CIP Motion, command data is received from the CIP Motion C-to-D Connection and processed by a fine interpolator. The job of the fine interpolator is to compute coefficients to a trajectory polynomial that is designed to reach the command data value at its associated Command Target Time. Depending on the specific command data element, the trajectory can follow a 1st, 2nd, or 3rd order polynomial trajectory with initial conditions based on current axis dynamics. Since the polynomial is a function of time, a new fine command value can be calculated any time the CIP Motion device needs to perform a control calculation. As a result, it is not necessary that the device's control calculation period be integrally divisible into the Controller Update Period.

To improve device interchangeability, the Motion Axis Object recommends a minimum order for the fine interpolators. Since contemporary motion planners typically generate their trajectories based on 3rd order polynomials in position, it is important that the fine interpolators reproduce these trajectories with high fidelity. Therefore, the position fine

interpolator is defined as 3rd order, the velocity interpolator is 2nd order, and the acceleration and torque interpolators are both 1st order.

Position Fine Interpolation Polynomial:

$$P(t) = a_0 + a_1 \times (t-t_0) + a_2 \times (t-t_0)^2 + a_3 \times (t-t_0)^3$$

Velocity Fine Interpolation Polynomial:

$$V(t) = b_0 + b_1 \times (t-t_0) + b_2 \times (t-t_0)^2$$

Acceleration Fine Interpolation Polynomial:

$$A(t) = c_0 + c_1 \times (t-t_0)$$

Torque Fine Interpolation Polynomial:

$$T(t) = d_0 + d_1 \times (t-t_0)$$

In these equations, time t_0 represents the Command Target Time for the previous motion planner update such that when $t = t_0$ the position, velocity, acceleration, and torque command values are equal to the values sent in the previous motion planner update, i.e. P_{-1} , V_{-1} , A_{-1} , and T_{-1} . This establishes the 0th order coefficients of the polynomials.

$$P(t_0) = P_{-1} = a_0$$

$$V(t_0) = V_{-1} = b_0$$

$$A(t_0) = A_{-1} = c_0$$

$$T(t_0) = T_{-1} = d_0$$

The higher order polynomial coefficients are calculated such that by the next motion planner update, corresponding to Command Target Time, t_1 , the position, velocity, acceleration, and torque command values are the values sent in the latest motion planner update, i.e. P_0 , V_0 , A_0 , and T_0 .

$$P(t_1) = P_0$$

$$V(t_1) = V_0$$

$$A(t_1) = A_0$$

$$T(t_1) = T_0$$

Using the above polynomial interpolation equations, the CIP Motion device can compute position, velocity, acceleration, and torque command values at any time by plugging in the current System Time value of the CIP Motion device into the variable, t . This allows the device's control calculation to be performed according to a schedule that is independent of the controller's update schedule.

One thing that shall be done, however, is to adjust the Command Target Time, t_0 , should there be a shift in the System Time Offset for the device; t_0 and t shall always be based on the same System Time reference system. For example, assume the device's System Time Offset when the control command timestamp, t_0 , was received is Offset_0 . If the command interpolation equation is to be applied at $t = t_1$ and the current System Time Offset is defined as Offset_1 then t_0 shall be adjusted as follows before executing the polynomial:

$$\text{Adjusted } t_0 = t_0 \times (\text{Offset}_1 - \text{Offset}_0)$$

The polynomial coefficients are computed based on standard formulas that are a function of the history of command values over the last few updates. The number of historical command values used in the formula depends on the order of the polynomial. For example, the third order command position polynomial uses the three previous command position values. For convenience, the interpolator polynomial coefficient formulae are as follows:

Position Fine Interpolation Polynomial Coefficients:

$$\begin{aligned} a_0 &= P_{-1} \\ a_1 &= 1/T \times (\Delta P_0 - 1/2 \times \Delta V_0 + 1/6 \times \Delta A_0) \\ a_2 &= 1/T^2 \times (1/2 \times \Delta V_0 - 1/2 \times \Delta A_0) \\ a_3 &= 1/T^3 \times (1/3 \times \Delta A_0) \end{aligned}$$

Velocity Fine Interpolation Polynomial Coefficients:

$$\begin{aligned} b_0 &= V_{-1} \\ b_1 &= 1/T \times (\Delta V_0 - 1/2 \times \Delta A_0) \\ b_2 &= 1/T^2 \times (1/2 \times \Delta A_0) \end{aligned}$$

Acceleration Fine Interpolation Polynomial Coefficients (Torque is same form as Accel):

$$\begin{aligned} c_0 &= A_{-1} \\ c_1 &= 1/T \times \Delta A_0 \end{aligned}$$

The above equations are based on the following nomenclature:

$$\begin{aligned} T &= \text{Controller Update Period} \\ \Delta P_0 &= (P_0 - P_{-1}) \\ \Delta V_0 &= (V_0 - V_{-1}) = (P_0 - 2P_{-1} + P_{-2}) \\ \Delta A_0 &= (A_0 - A_{-1}) = (V_0 - V_{-1} - V_{-2}) = (P_0 - 3P_{-1} + 3P_{-2} - P_{-3}) \end{aligned}$$

When $t > t_1$, the fine interpolation polynomial becomes an extrapolation polynomial. In the absence of a fresh update from the motion planner, the extrapolation polynomial can be used to provide estimated command data to the control structure until fresh motion planner command data is available. Once fresh command data is made available, new polynomial coefficients shall be computed without delay. In this way, the motion control can be maintained (“ride-thru”) despite occasional late or lost connection data packets resulting in a robust distributed motion control network solution. To be clear, late connection data is always applied and never thrown away: late data still represents the freshest data available from the controller and the extrapolation polynomial ensures that the command data is applied in such a way as to maintain a smooth motion trajectory despite variations in command data delivery.

When the update period of the motion planner is short enough relative to the dynamics of the command trajectory, or is comparable to the device’s control calculation period, fine interpolation may not be necessary. The motion planner can make this determination by comparing the planner update period to that of the device’s control calculation period. When fine interpolation is used, the planner shall add additional planner update periods to the planner time stamp, so it is advantageous to eliminate this planner update period delay if interpolation is not necessary.

Even though fine interpolation may not be necessary in some cases, it does not mean that the command data is to be applied directly to the control structure. It still may be necessary to calculate the above polynomials so the device can extrapolate the command value when the device’s control update occurs. That is because, in general, the device’s control update time stamp does not need to match the time stamp of the command data.

Finally, there are CIP Motion device types and control modes that do not require the dynamic accuracy that time-stamped interpolation and extrapolation provide. Various velocity and torque control applications, for example, may fall in this category. In general, command data can also be applied to the control structures of Variable Frequency drives without interpolation or extrapolation.

7.6.6.3 Feed-Forward signal selection

The fine interpolators defined as part of the Motion Axis Object can generate higher derivatives of the command data input to serve as feedforward signals. Superior signal quality, however, can be provided by the motion planner trajectory generators. The feedforward selection blocks pick the best feed-forward signal to apply based on the bits set in the Command Data Set attribute. The best signal is defined as the signal derived using the fewest differencing operations. The interpolated command position is applied directly to the position control loop without any of the typical de-referencing and offsets. It is assumed that these operations are performed by the controller based motion planner.

7.6.6.4 Velocity Limiter

The Interpolated Velocity Command signal passes through a series of dynamic limiters that are commonly used for velocity control. The Velocity Limiter provides independent positive and negative limits for the velocity command reference that can be applied to the device's control structure.

7.6.6.5 Skip Bands

Skip Bands are typically used in Frequency Control applications when certain speeds excite mechanical resonance frequencies of the motor and load. The Skip Band feature allows three separate Skip Speeds to be defined that shift the Interpolated Velocity Command signal to avoid, or skip, these problematic speeds. The Skip Speed Band determines the range of speeds centered on the three Skip Speeds that the device avoids. If the Interpolated Velocity Command falls within the Skip Band, but below the Skip Speed, the Velocity Command output is set to the Skip Speed, minus half the Skip Speed Band. If the Interpolated Velocity Command falls within the Skip Band, but above the Skip Speed, the Velocity Command output is set to the Skip Speed, plus half the Skip Speed Band.

7.6.6.6 Acceleration Limiter

The Acceleration Limiter limits the rate of change of the Velocity Command based on independent Acceleration and Deceleration Limits. Such a limiter is particularly valuable when the Interpolated Velocity Command is changed in discrete steps, such as when the motor speed is being controlled via a series of Controller Velocity Command set-points. The Acceleration Limiter turns the step changes of the Interpolated Velocity Command into controlled acceleration and deceleration ramps. The Acceleration Limiter can also support Jerk Limit Control that produces an S-Curve response to step changes in velocity command, providing even smoother speed transitions. The Jerk Limit Control attribute determines what percentage of the acceleration or deceleration ramps is S-Curve, with the remaining portion of the ramp governed by the fixed Acceleration or Deceleration Limit.

7.6.7 Feedback Interface Behavior

7.6.7.1 Feedback Sources

Feedback signals defined by the Motion Axis Object can be derived from any of 4 different feedback interface channels (see Figure 70 and Figure 71). The two primary feedback channels employed by the various closed loop control modes are designated Feedback 1 and Feedback 2. This allows the control loops to operate with either a motor based feedback device that is typically attached to the Feedback 1 channel or a load-side feedback device that is connected to the Feedback 2 channel. Which feedback source is used by the loop is governed by the Feedback Configuration attribute.

Each feedback interface is capable of supporting a number of different feedback device types as enumerated by the Feedback Type attribute. The feedback interface output is the number of feedback counts that the feedback device has moved since the last time the device was sampled. If the feedback device is an absolute device, the feedback interface also determines the absolute position of the feedback device at power-up and communicates that value to the Feedback Accumulator to preset the accumulator.

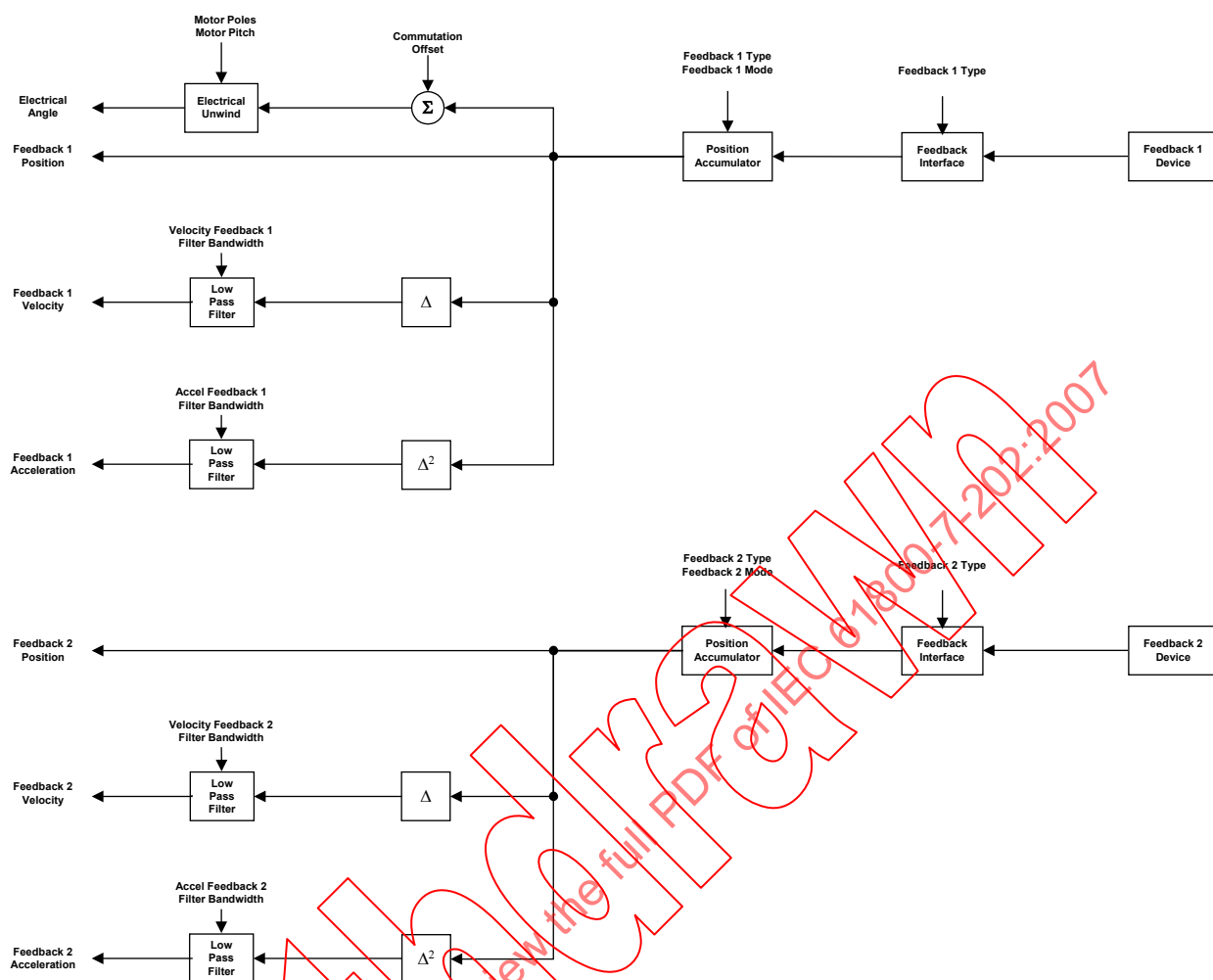


Figure 70 – Feedback Channels 1 and 2

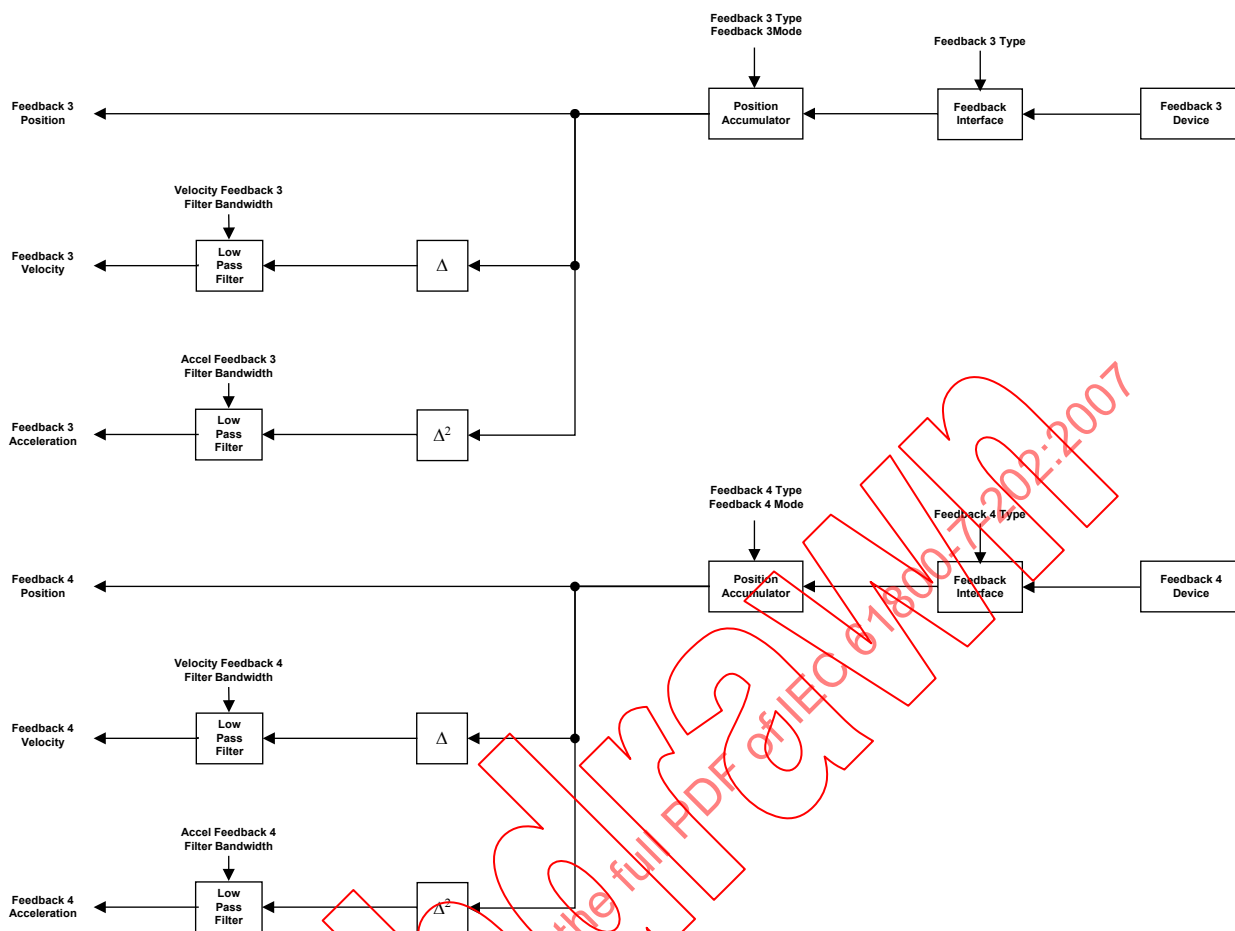


Figure 71 – Feedback Channels 3 and 4

7.6.7.2 Feedback Accumulator

The role of the Feedback Accumulator depends on the configured Feedback Mode which can be either Incremental or Absolute. If Incremental is selected, the accumulator simply accumulates changes to the feedback count value, a 32-bit signed integer, with every device update. If Absolute is selected, the Feedback Accumulator works basically the same way as in Incremental mode. The only difference is the initialization of the accumulator at device power-up. In Incremental mode, the Feedback Accumulator is set to zero, while in Absolute mode, the accumulator is initialized to the absolute position of the feedback device. This allows for the recovery of absolute position through a power-cycle as long as power-off movement of the absolute feedback device is limited to $\frac{1}{2}$ of the absolute feedback range of the device. There is no requirement to extend the absolute position range of the feedback device through non-volatile storage of the accumulator. This simple absolute feedback handling mechanism is due to the fact that the CIP Motion places the responsibility of extending the absolute position range of the axis, and establishing the absolute machine position reference, on the controller.

7.6.7.3 Commutation Unwind And Offset

An Electronic Unwind block is also connected to the Feedback 1 interface. This block is designed to unwind, or modulo, the position accumulator output to generate a signal that is proportional to the electrical angle of a Permanent Magnet motor based on the Pole Count or Pole Pitch of the motor. To align this signal with the physical UVW windings of the motor rather than the zero of the feedback device, a configurable Commutation Offset is added prior to the Electrical Unwind block.

7.6.7.4 Feedback Filtering

A configurable low-pass IIR filter is defined by the Motion Axis Object for filtering the velocity and acceleration estimates for each feedback channel. These filters can be used to reduce the level of quantization noise associated with differencing digital feedback signals. The bandwidth of the velocity and acceleration IIR filters for each feedback channel are individually programmable.

7.6.8 Event Capture Behavior

7.6.8.1 Event Input Sources

The Motion Axis Object defines a mechanism to capture both the feedback position and time stamp associated with specific state transitions of selected event input sources. Event input sources currently supported by the object are Registration 1, Registration 2, Marker, and Home Switch. These 4 event input sources apply to each supported feedback channel (see Figure 72).

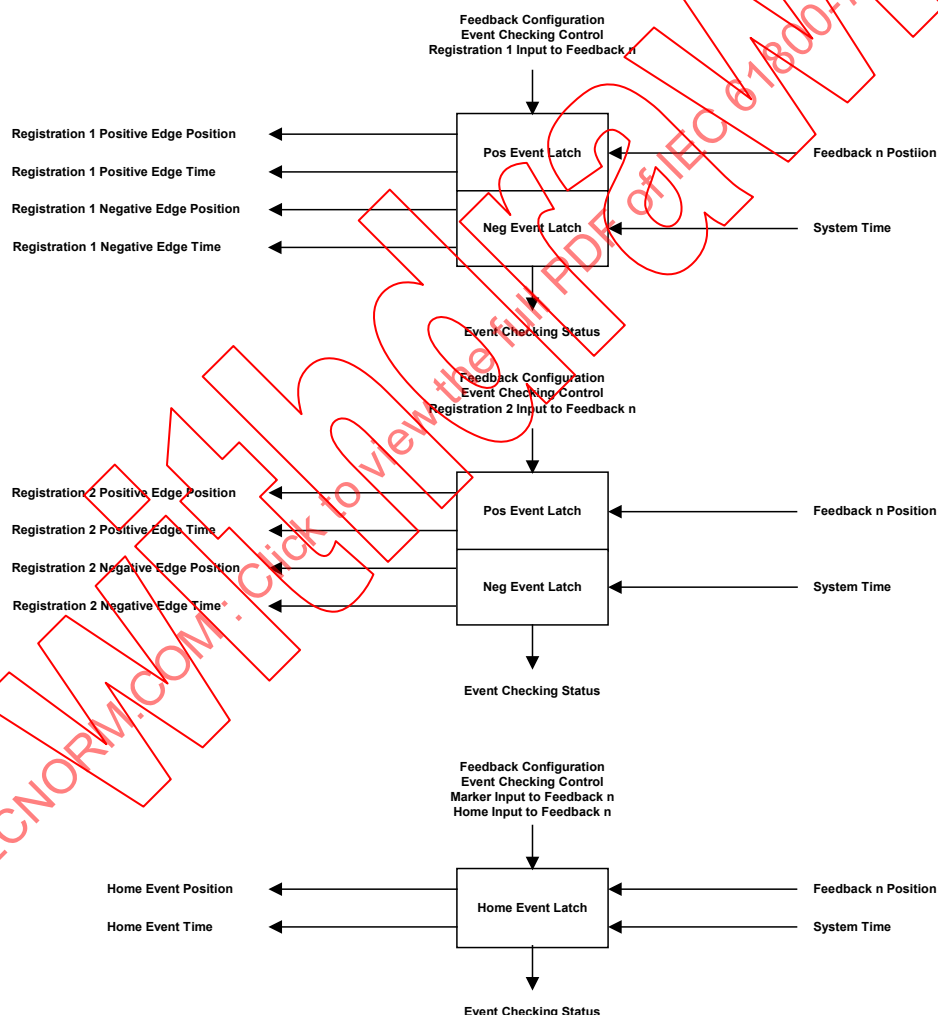


Figure 72 – Event Capture Functionality

7.6.8.2 Event Latches

To facilitate accurate capture of both feedback position and time, hardware event latches are typically implemented as shown in the following block diagram. Two independent latches are defined for each registration input, one latch to capture positive edge transition events and one to capture negative edge transition events. This design enables capture of both

registration events in applications with narrow registration pulses where the rising and falling edges occur nearly simultaneously. In addition to the registration latches, a separate latch is also defined for the home event capture. The home input event that triggers the Home Event Latch can be any of a number of different combinations of home switch and marker input events, i.e. marker transitions, switch transitions, or switch transitions followed by a marker transition.

With hardware based event latches, event capture accuracy is, in general, only limited by the latency of the associated event input. Registration and Marker event inputs are lightly filtered so event capture accuracy is in the order of 1 μ s. In terms of position capture accuracy, that would be calculated as the product of the event capture accuracy and the speed of the axis. Home switch inputs are heavily filtered, in general, and therefore limited to an event capture accuracy of 1 ms to 10 ms. Thus, to get an accurate position capture based on a home switch input transition, a homing sequence with a slow homing speed is required.

7.6.8.3 Event Time Stamps

Since the registration time stamp is passed to the controller as part of the Event Notification data, the controller can apply the event time stamp to the position history of other axes in the system to interpolate their positions. This is particularly useful in applications where it is necessary to determine the location of several axes at the time of a single registration event. The more accurate the time stamp, the more accurately the controller can determine these positions.

7.6.9 Control Mode Behaviour

7.6.9.1 General

The instance attributes defined in 7.3 affect device behavior in the context of the Control Mode, Control Method, and Feedback Configuration. As discussed in 4.2 concerning the scope of the Motion Axis Object, there are basically 4 Control Modes common to CIP Motion devices, position control, velocity control, torque control, and no control. To this list we add a new control mode, acceleration control, to complete the progression from velocity control to torque control. This subclause provides a block diagram for each of the control modes in an effort to further define the collective behavior of the various Motion Axis Object attributes.

7.6.9.2 No Control (Feedback Only) Mode

A Motion Axis Object instance can be configured for No Control mode to allow for the position, velocity, and acceleration of any one of four possible feedback channels to be accessed by the controller via the Device-to-Controller (D-to-C) Connection. These signals can then be distributed across the motion control system as a master axis for gearing and camming operations. In this mode, the Feedback Configuration attribute determines which feedback channel produces the Position, Velocity, and Acceleration Feedback signals (see Figure 73).

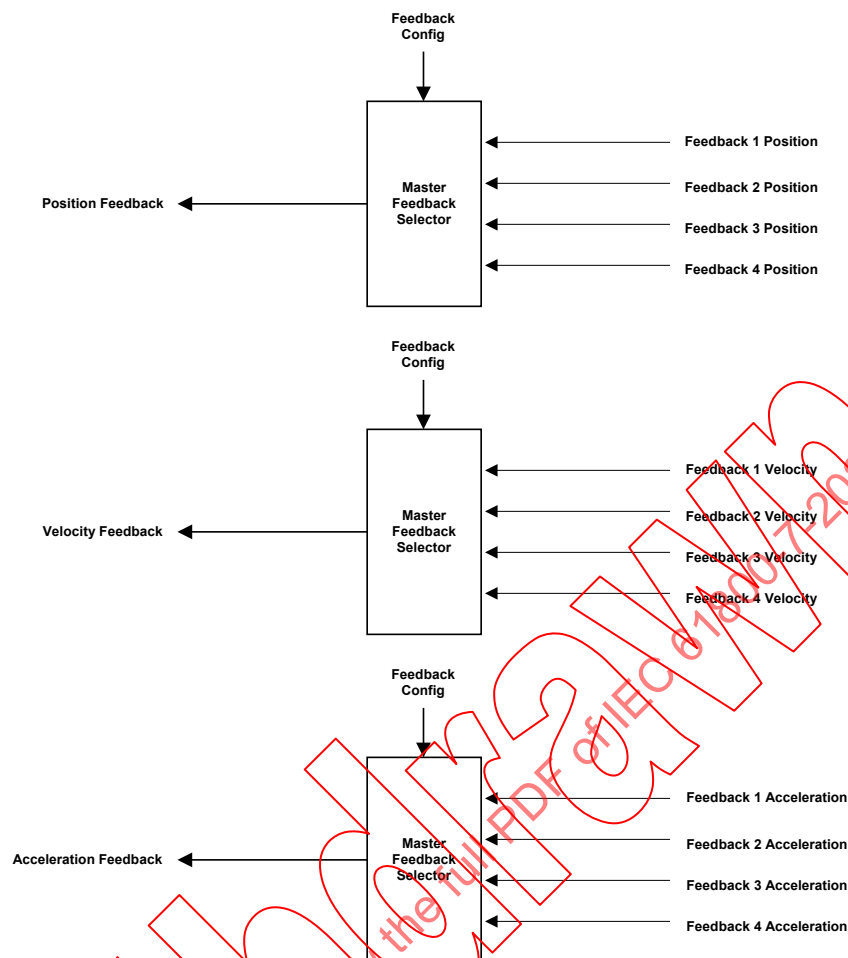


Figure 73 – No Control (Feedback Only)

7.6.9.3 Position Control Mode

In Position Control mode, the only operative Control Method supported by the object currently is Closed Loop servo control. At a later date, the object could be expanded to include a Stepper based position Control Method.

7.6.9.4 Closed Loop Position Control

7.6.9.4.1 General

When performing closed loop position control, the CIP Motion device applies the Position Command signal output of the Command Generator to the position loop summing junction. In addition to the Position Command, a Position Trim input is provided which can be used to provide an offset to the position loop. The classic PI control loop generates a Position Loop Output signal to an inner velocity loop (see Figure 74).

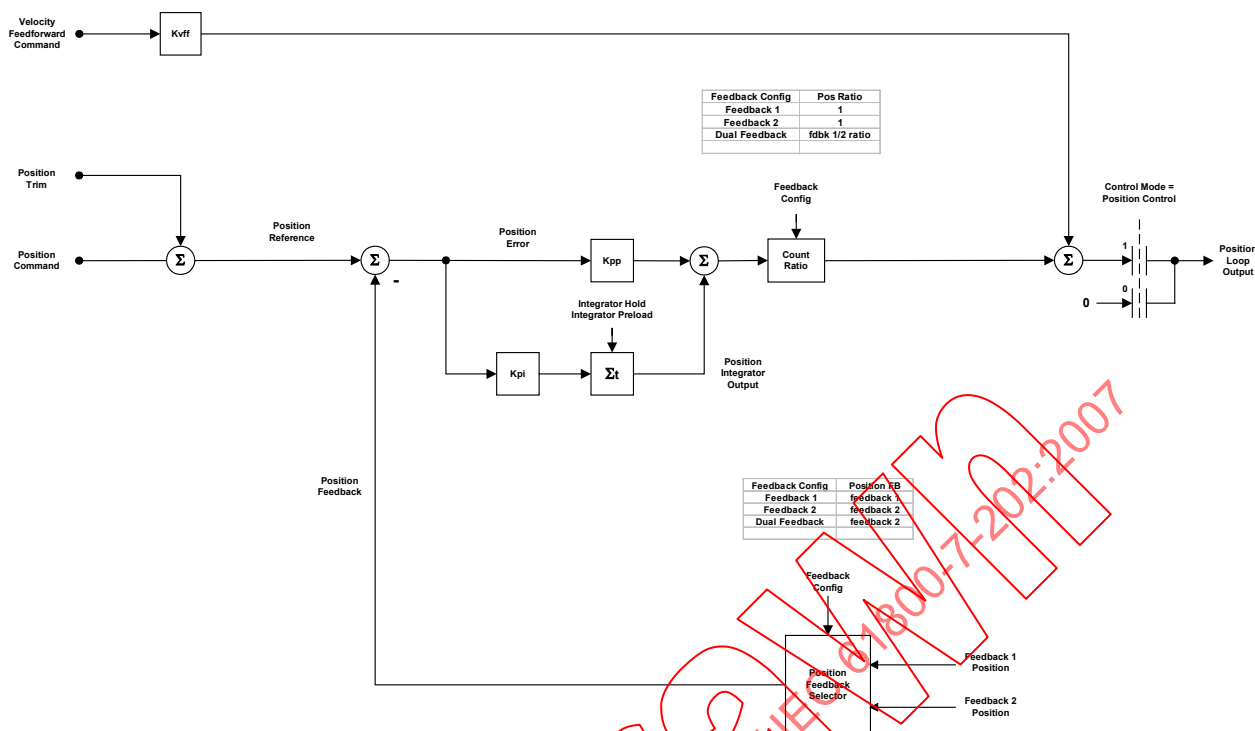


Figure 74 – Closed Loop Position Control

7.6.9.4.2 Position feedback configuration

Feedback to the PI regulator can be derived from two different feedback channels. This flexibility allows the position loop to operate with either a motor based feedback device that is typically attached to the Feedback 1 channel or a load-side feedback device that is connected to the Feedback 2 channel. Which feedback source is used by the loop is governed by the Feedback Configuration attribute.

When the Feedback Configuration calls for Dual Feedback operation, the position loop utilizes the Feedback 2 channel and the velocity loop uses the Feedback 1 channel. Since the two feedback channels may not have the same feedback resolution, it is necessary to convert position loop output from Feedback 1 counts to Feedback 2 counts prior to applying the output to the velocity loop summing junction. This is done by scaling the position loop output by the Feedback 1/2 Count Ratio.

7.6.9.4.3 Position PI gains

The Proportional Gain of the classic PI controller sets the unity gain bandwidth of the position loop in rad/s, while the Integral Gain is used to drive the Position Error signal to zero to compensate for the effect of any static and quasi-static torque or forces applied to the load.

7.6.9.4.4 Velocity Feed-Forward

The inner velocity loop requires a non-zero command input to generate steady-state axis motor velocity. To provide the non-zero output from the CIP Motion device to the motor, a non-zero position loop output is required, which translates to a non-zero position error. This dynamic error between command position and actual position while moving is often called "following error". Most closed loop motion control applications desire zero following error -- all the time. This could be achieved to some extent through use of the position integral gain control as described above, but typically, the response time of the integrator action is too slow to be effective in high-performance motion control applications. An alternative approach that has superior dynamic response is to use Velocity Feedforward.

The Velocity Feedforward feature is used in Position Control mode to provide the bulk of the Velocity Reference input necessary to generate the desired motor velocity. It does this by scaling the Velocity Feed-forward Command signal output of the Command Generator by the Velocity Feedforward Gain and adding to the Position Loop Output generated by the position loop to form the Velocity Reference signal. With this feature, the position loop does not need to generate much effort to produce the required velocity command level, hence the Position Error value is significantly reduced. The Velocity Feedforward Gain allows the following error of the position control loop to be reduced to nearly zero when running at a constant velocity. This is important in applications such as electronic gearing and synchronization applications where it is necessary that the actual axis position does not significantly lag behind the commanded position at any time.

The optimal value for Velocity Feedforward Gain is theoretically 100 %. In reality, however, the value may need to be tweaked to accommodate velocity loops with finite loop gain. One aspect that may force a smaller Velocity Feedforward value is that increasing amounts of feedforward tends to exacerbate axis overshoot. For this reason, feedforward is not recommended for point-to-point positioning applications.

7.6.9.5 Velocity Control Mode

In Velocity Control mode, there are two operative control methods supported by the object, Closed Loop Velocity Control and Open Loop Frequency Control.

7.6.9.6 Closed Loop Velocity Control

7.6.9.6.1 General

The Closed Loop velocity control method is targeted for applications that require tight speed regulation. The command input to the velocity loop can be derived directly from the Velocity Command of the Command Generator when configured for Velocity Control Mode or from the Position Loop Output when configured for Position Control Mode as described in 7.6.9.3 (see Figure 75).

When serving as an outer velocity loop in Velocity Control Mode, the device applies the Velocity Command input to the velocity summing junction as the Velocity Reference of a classic PI regulator. Also contributing to the Velocity Reference input signal is the Velocity Trim input, which can be used in conjunction with an outer control loop to make minor adjustments to the velocity of the motor.

When serving as an inner velocity loop in Position Control Mode, the device applies the Position Loop Output signal to the input of the velocity summing junction. Input signals that are not applicable to the configured control mode are generally set to zero.

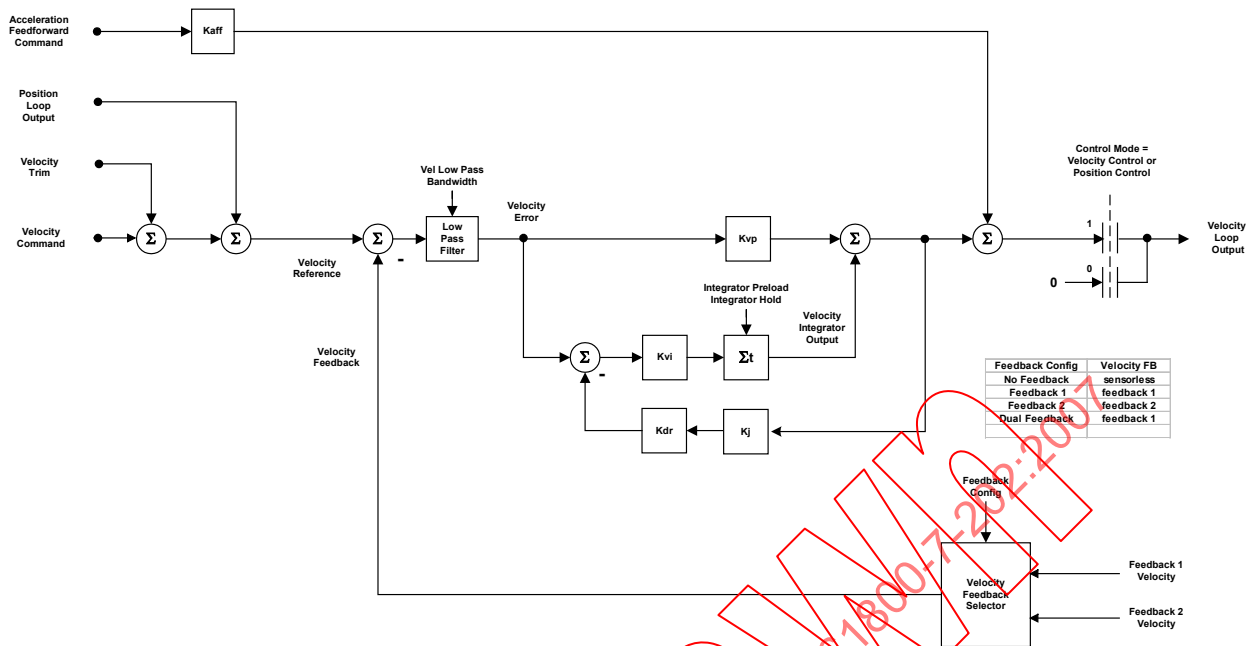


Figure 75 – Closed Loop Velocity Control

7.6.9.6.2 Velocity Feedback Configuration

Feedback to the PI regulator can be derived from either of the two available feedback transducers, Feedback 1 or Feedback 2. Which feedback source is used by the loop is governed by the Feedback Configuration enumeration.

7.6.9.6.3 Velocity Error Filter

A low pass filter can be optionally applied to the velocity error signal generated by velocity loop summing junction. The output of this filter becomes the Velocity Error signal that is subsequently operated on by the velocity loop PI control algorithm. When used, the filter is typically set between 5 to 10 times the velocity loop bandwidth. It is recommended that this filter be a two pole IIR filter to maximum its effectiveness at quantization noise filtering.

7.6.9.6.4 Velocity PI gains

The velocity loop generates a Velocity Loop Output signal to the next inner loop via a classic PI control loop structure. The Proportional Gain of the controller sets the unity gain bandwidth of the velocity loop in rad/s, while the Integral Gain is used to drive the Velocity Error signal to zero to compensate for any static and quasi-static torque or forces applied to the load.

The integral section of the velocity regulator includes an anti-windup feature. The anti-windup feature automatically holds the regulator's integral term when a limit condition is reached in the forward path. The anti-windup feature is conditioned by the arithmetic sign of the integrator's input. The integrator is held when the input's sign is such that the integrator output moves further into the active limit. In other words, the integrator is allowed to operate (not held) when the input would tend to bring the integrator output value off the active limit.

The integrator may also be configured for integrator hold operation. When the Integrator Hold attribute is set true, the regulator holds the integrator from accumulating while the axis is being commanded to move. This behavior is helpful in point-to-point positioning applications.

An automatic preset feature of the velocity regulator's integral term occurs when a transition is made from a torque control mode to speed control mode, using the Control Mode selection parameter. Upon transition to speed mode, the speed regulator's integral term is preset to the

motor torque reference parameter. If the speed error is small, this provides a 'bumpless' transition from the last torque reference value present just prior to entering speed mode.

7.6.9.6.5 Velocity droop

Another feature of the velocity regulator is the velocity droop function. The velocity error input to the integral term is reduced by a fraction of the velocity regulator's output, as controlled by the droop gain setting, Kdr. As torque loading on the motor is increased, actual motor speed is reduced in proportion to the droop gain. This is helpful when some level of compliance is required due to rigid mechanical coupling between two motors.

7.6.9.6.6 Acceleration Feed-Forward

The velocity loop requires a non-zero velocity loop output to generate steady-state axis motor acceleration. To provide the non-zero output from the drive device to the motor, a non-zero velocity error is generally required. In position control applications, this non-zero velocity error translates to a non-zero position loop error. Since many closed loop motion control applications require near zero control loop error, this behavior is not desirable. Again, the position and velocity loop error could be reduced by applying the velocity integral gain control as described above, but the integrator action is still too slow to be very effective. The preferred approach with superior dynamic response is to use Acceleration Feed-Forward.

The Acceleration Feed-forward Gain attribute is used to generate the output necessary to generate the commanded acceleration. It does this by scaling the Acceleration Feed-forward Command generated by the Command Generator by the Acceleration Feedforward Gain and adding it as an offset to the output of the velocity loop. With this feature, the velocity loop does not need to generate much control effort, thus reducing the amount of control loop error.

The optimal value for Acceleration Feedforward is 100 % theoretically. In reality, however, the value may need to be tweaked to accommodate variations in load inertia and the torque constant of the motor. Like Velocity Feedforward, Acceleration Feedforward can result in overshoot behavior and therefore should not be used in point-to-point positioning applications.

When used in conjunction with the Velocity Feedforward, the Acceleration Feedforward allows the following error of the position or velocity control loop to be reduced to nearly zero during the acceleration and deceleration phases of motion. This is important in tracking applications that use electronic gearing and camming operations to precisely synchronize a slave axis to the movements of a master axis.

7.6.9.7 Open Loop Frequency Control

7.6.9.7.1 General

Another Velocity Control method is the open loop Frequency Control method associated with so called Volts/Hertz or Variable Frequency Drives (VFDs) that do not have a current control loop and typically drive an induction motor. Velocity control with this method is achieved by controlling the voltage and frequency output of the drive device in some manner where voltage is generally proportional to frequency. For an induction motor, the velocity of the motor is determined by the Output Frequency of the drive device divided by the Motor Pole count. This control method is applicable to velocity control applications that do not require tight speed regulation and therefore do not require a feedback device. Figure 76 illustrates this open loop velocity control method.

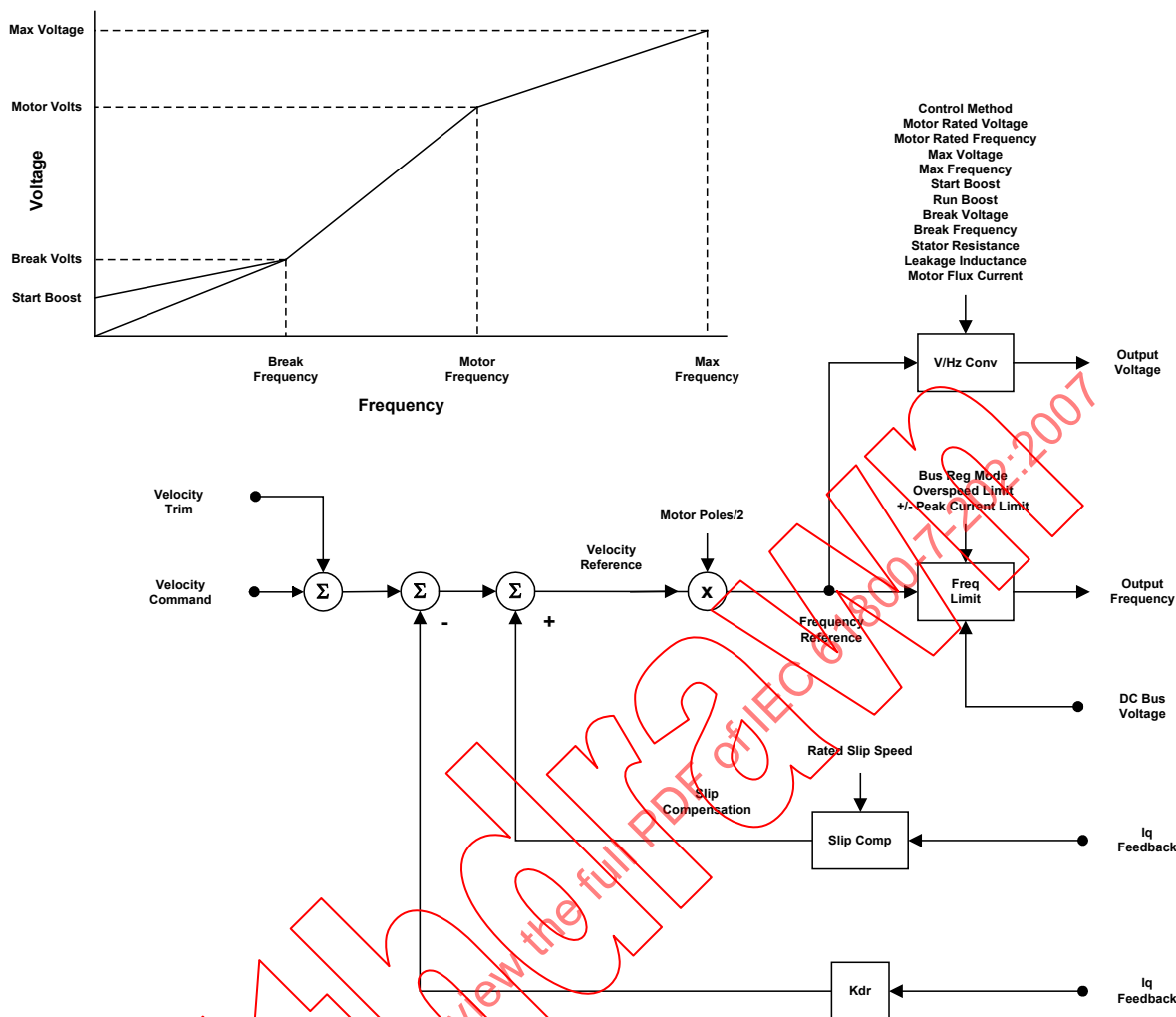


Figure 76 – Open Loop Frequency Control

7.6.9.7.2 Basic Volts/Hertz Operation

The Motion Axis Object provides a number attributes that are used to specify the relationship the CIP Motion device uses between output frequency (speed) and output voltage for a given (induction) motor. The Break Frequency and Break Voltage attributes define the point on the Volts/Hertz curve below which the Start Boost feature is applied. As the name indicates, Start Boost is used to provide a non-zero output voltage to the motor at stand-still to assist start-up. The contribution of the Start Boost to the output voltage of the drive device tapers off to zero when the motor reaches the Break Frequency. Above the break point, output voltage and output frequency follow a linear slope to the point defined by the Motor Rated Frequency and Motor Rated Voltage. From this point on, the Volts/Hertz curve follows another linear slope to the point defined by the Max Frequency and Max Voltage attributes. This segment of the Volts/Hertz curve allows for operation above the rated frequency and voltage of the motor in applications where that is required.

7.6.9.7.3 Slip Compensation

When driving an induction motor at a specific frequency, the actual motor velocity is generally less than the command speed, given by the output frequency divided by the motor pole count, by an amount that is proportional to the load torque applied to the motor. This difference in speed is called “Slip” and is a configuration attribute associated with the motor. The Motion Axis Object supports a Slip Compensation feature that is common to Variable Frequency Drives. The amount of Slip Compensation applied to the Velocity Reference is the product of the measured torque producing current, I_q , and the configured Rated Slip Speed.

7.6.9.7.4 Velocity Droop

Another feature defined for the Frequency Control method is the droop function. The droop function reduces the velocity reference by a scaled fraction of the torque producing current, I_q , as controlled by the droop gain setting, K_{dr} . As torque loading on the motor is increased, actual motor speed is reduced in proportion to the droop gain. This is helpful when some level of compliance is required when performing torque sharing between two motors on a common load.

7.6.9.8 Acceleration Control Mode

While dynamic motor control via an acceleration command is not common in the industry, Acceleration Control was added to the Motion Axis Object to complete the dynamic progression from velocity control to torque control. The output of the velocity loop, Velocity Loop Output, also has units of acceleration. So, like the other control modes we sum the contributions of the Acceleration Command, Acceleration Trim, and Velocity Loop Output to form the Acceleration Reference signal that serves as one of the primary inputs to the Torque Control section (see Figure 77).

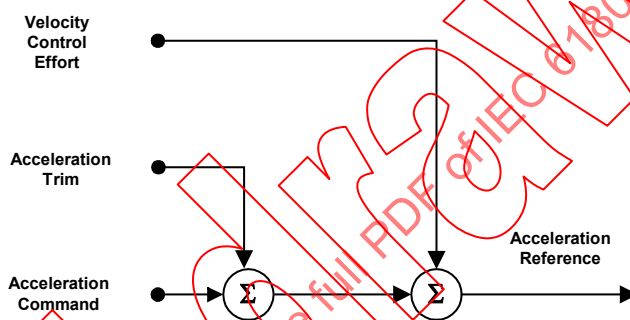


Figure 77 – Acceleration Control

7.6.9.9 Torque Control

7.6.9.9.1 General

Torque is generally proportional to acceleration and to the torque producing motor current, I_q . The purpose of the Torque Control structure is to combine input signals to create a Torque Reference from a number of different sources based on the Control Mode and apply various filters and compensation algorithms to the Torque Reference to create a Filtered Torque Reference. The Filtered Torque Reference signal is scaled by the torque constant of the motor to become the Current Command input to the current loop. Since the motor current is also per unitized to the % Rated current of the motor, the torque constant, K_t , is nominally 1.

7.6.9.9.2 Torque Input Sources

The Torque Control section can take input from a variety of sources depending on the Control Mode. Input to the Torque Reference path can come via the cyclic Torque Command or Torque Trim signal in Torque Control mode. In Position or Velocity Control mode, torque input can also be derived from the outer velocity loop by bringing in the Velocity Loop Output into the torque loop summing junction. In Acceleration Control Mode, the Acceleration Reference signal is applied, representing the sum of the cyclic Acceleration Command and Acceleration Trim data (see Figure 78).

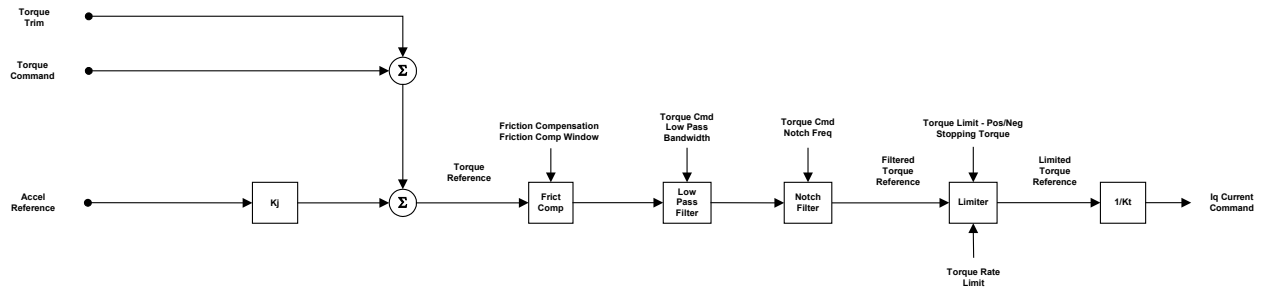


Figure 78 – Torque Control

7.6.9.9.3 Torque Scaling

Since the Velocity Loop Output and Acceleration related values have units of acceleration, these values are multiplied by a Torque Scaling gain, K_j , to convert to “per unitized” torque. The torque units defined by the Motion Axis Object are expressed as % of Rated Torque of the motor. The Torque Scaling gain, when properly configured, represents the total inertia or mass of the system that includes the motor and the load.

7.6.9.9.4 Low Pass Filter

The Low Pass Filter is effective in resonance control when the natural resonance frequency is much higher ($>5x$) than the control loop bandwidth. This filter works by reducing the amount of high-frequency energy in the output signal that excite the natural resonance. The Low Pass Filter design can be single pole or multiple poles. Care shall be taken, however, to limit the amount of phase lag introduced by this filter to the control loop to avoid potential instability.

7.6.9.9.5 Notch Filter

The notch filter is effective in resonance control when the natural resonance frequency is higher than the control loop bandwidth. Like the Low Pass filter, the notch filter works by significantly reducing the amount of energy in the output signal that can excite the natural resonance. It can be used even when the natural resonance frequency is relatively close to the control loop bandwidth. That is because the phase lag introduced by the notch filter is localized around the notch frequency. For the notch filter to be effective, the Notch Frequency has to be set very close to the natural resonance frequency of the load.

A typical equation for the notch filter is as follows:

$$G(s) = \frac{s^2 + \omega_n^2}{s^2 + s \times \omega_n/Q + \omega_n^2}$$

In this equation, Q represents the sharpness of the notch. In most implementations, the sharpness, Q , is typically hard-coded. The attenuation depth of the notch filter is infinite.

7.6.9.9.6 Torque Limiter

The Filtered Torque Reference signal passes through a limiter block to produce the Limited Torque Reference signal. The Torque Limiter block applies a torque limit to the signal that is based on the sign of the torque reference signal input and the state of the axis. During normal operation it is the Torque Limit – Positive and Torque Limit – Negative attributes, set by the user, that are applied to the torque reference signal. When the axis is commanded to stop as part of a disable request or major fault condition, the device applies the Stopping Torque Limit.

Also included with the torque limit block is a built in Torque Rate of Change Limit. This feature limits the rate of change of the torque reference output.

7.6.9.10.1 General

In the case of an induction motor, the flux producing current, I_d , is solely responsible for generating rotor flux. In the case of permanent magnet motors, rotor flux is generated by the rotor magnets and I_d is only used to in some cases to extend the speed range of the motor by changing the angle of stator field relative to the rotor field. In this case, the angle of I_q relative to the rotor field remains the same, i.e. at quadrature. But since the vector combination of I_q and I_d determines the stator flux angle relative to the rotor, increasing amounts of I_d can shift the stator flux away from quadrature to extend the speed range of the motor at the expense of torque.

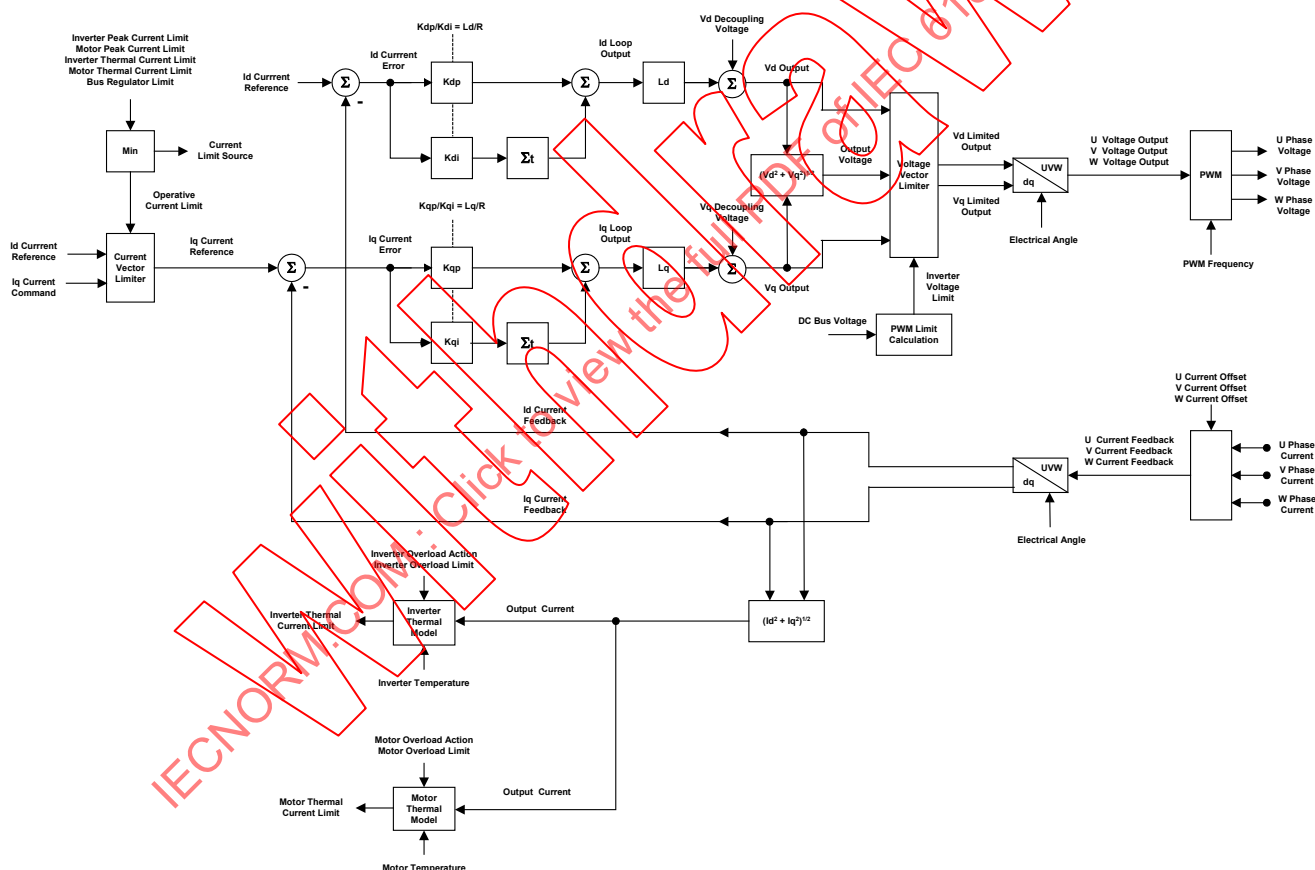


Figure 79 – Closed Loop Current Vector Control

7.6.9.10.2 Current vector limiter

The Iq Current Command passes through a Current Vector Limiter block before becoming the Iq Current Reference signal. This limiter block computes the combined vector magnitude of the Iq Current Command and the Id Current Reference signals. The resultant current vector magnitude is compared to the Operative Current Limit that represents the minimum current limit from among a set of potential current limits of the drive device and motor. If the vector magnitude exceeds the Operative Current Limit, the Iq Current Command is reduced so the vector magnitude equals the Operative Current Limit. Potential current limit sources can be

the Peak Current Limit ratings as well as the Thermal Limits for the Motor and Drive Inverter. Some of these limits are conditional and dynamic, such as the Motor and Inverter Thermal Current Limits derived from the thermal models for these devices. These limits are only active when the corresponding Motor and Inverter Overload Action attributes are set to provide current fold-back. The thermal current limits in this case would decrease as the simulated temperature of the modeled devices increases. The Bus Regulator Limit is only applied when the motor is regenerating power onto the DC Bus and is based on the Bus Regulator Power Limit.

With all these potential current limit sources that could be operative, a Current Limit Source attributes was included with the Motion Axis Object to identify the source of the active current limit.

7.6.9.10.3 Voltage output

The output of each current loop is scaled by the motor inductance to generate a voltage command to the vector transformation block. It is the job of the vector transformation block to transform the torque producing, V_q , and flux producing, V_d , command signals from the rotating synchronous reference frame to the stationary stator reference frame. The resultant U, V, and W Output Voltage values are then applied to the motor by Pulse Width Modulation (PWM). The PWM Frequency is also a configurable attribute of the Motion Axis Object.

The magnitude of the V_q , V_d vector is calculated in real time as the total Output Voltage signal. The maximum Output Voltage signal that can be applied to the motor is ultimately limited by the DC Bus Voltage and enforced by the Voltage Vector Limiter. Any attempt to exceed this value results in an Inverter Voltage Limit condition.

7.6.9.10.4 Current feedback

Current feedback signals to the current loop are provided by two or three current sensors. The signals from these sensors are conditioned and corrected for device specific offsets to become the U, V, and W Current Feedback signals associated with the stationary motor stator frame. These three signals are transformed back to the synchronous reference frame to generate the I_q and I_d Current Feedback signals. The magnitude of the I_q , I_d current vector is calculated in real-time and used as an input to the thermal models for the inverter and motor.

7.6.9.10.5 Motor commutation

Motor commutation is critical to closed loop motor control. The orientation of the motor rotor can be determined from a feedback source mounted to the motor. The actual commutation source is the motor feedback device assigned to Feedback 1. Once the feedback device is calibrated to the absolute orientation of the rotor using the Commutation Offset attribute, the commutation block can then generate the true Electrical Angle of the rotor. This signal is used to perform the vector transforms between the rotary and stationary motor frames and can also be used for any other algorithms that require knowledge of rotor position.

Bibliography

IEC 60050-351, *International Electrotechnical Vocabulary – Part 351: Control technology*¹⁴

IEC 61158 (all parts), *Industrial communication networks – Fieldbus specifications*

IEC 61158-2, *Industrial communication networks – Fieldbus specifications – Part 2 (Ed.4.0): Physical layer specification and service definition*

IEC 61158-3-2, *Industrial communication networks – Fieldbus specifications – Part 3-2 (Ed.1.0): Data-link layer service definition – Type 2 elements*

IEC 61158-4-2, *Industrial communication networks – Fieldbus specifications – Part 4-2 (Ed.1.0): Data-link layer protocol specification – Type 2 elements*

IEC 61499-1:2005, *Function blocks – Part 1: Architecture*

IEC 61784-1, *Industrial communication networks – Profiles – Part 1: Fieldbus profiles*

IEC 61784-2, *Industrial communication networks – Profiles – Part 2: Additional fieldbus profiles for real-time networks based on ISO/IEC 8802-3*

IEC 61800 (all parts), *Adjustable speed electrical power drive systems*

IEC 61800-7-201, *Adjustable speed electrical power drive systems – Part 7-201: Generic interface and use of profiles for power drive systems – Profile type 1 specification*

IEC 61800-7-203, *Adjustable speed electrical power drive systems – Part 7-203: Generic interface and use of profiles for power drive systems – Profile type 3 specification*

IEC 61800-7-204, *Adjustable speed electrical power drive systems – Part 7-204: Generic interface and use of profiles for power drive systems – Profile type 4 specification*

IEC 61800-7-301, *Adjustable speed electrical power drive systems – Part 7-301: Generic interface and use of profiles for power drive systems – Mapping of profile type 1 to network technologies*

IEC 61800-7-302, *Adjustable speed electrical power drive systems – Part 7-302: Generic interface and use of profiles for power drive systems – Mapping of profile type 2 to network technologies*

IEC 61800-7-303, *Adjustable speed electrical power drive systems – Part 7-303: Generic interface and use of profiles for power drive systems – Mapping of profile type 3 to network technologies*

IEC 61800-7-304, *Adjustable speed electrical power drive systems – Part 7-304: Generic interface and use of profiles for power drive systems – Mapping of profile type 4 to network technologies*

IEC/TS 61915, *Low-voltage switchgear and controlgear – Principles for the development of device profiles for networked industrial devices*

¹⁴ See also the IEC Multilingual Dictionary – Electricity, Electronics and Telecommunications.

IEC 62026-3, *Low-voltage switchgear and controlgear – Controller-device interfaces (CDIs) – Part 3: DeviceNet™*

IEC/TR 62390:2005, *Common Automation Device – Profile Guideline*

ISO/IEC 2382-15:1999, *Information technology – Vocabulary – Part 15: Programming languages*

ISO/IEC 19501, *Information technology – Open Distributed Processing - Unified Modeling Language (UML) Version 1.4.2*

ISO 15745-1:2003, *Industrial automation systems and integration – Open systems application integration framework – Part 1: Generic reference description*

EN 50325-4, *Industrial communication subsystem based on ISO 11898 (CAN) for controller-device interfaces – Part 4: CANopen*

ODVA: *THE CIP NETWORKS LIBRARY – Volume 1: Common Industrial Protocol (CIP™) – Edition 3.1, November 2006, available at <<http://www.odva.org>>*

ODVA: *THE CIP NETWORKS LIBRARY – Volume 2: EtherNet/IP™ Adaptation of CIP – Edition 1.3, November 2006, available at <<http://www.odva.org>>*

ODVA: *THE CIP NETWORKS LIBRARY – Volume 3: DeviceNet™ Adaptation of CIP – Edition 1.3, November 2006, available at <<http://www.odva.org>>*

ControlNet International: *THE CIP NETWORKS LIBRARY – Volume 4: ControlNet™ Adaptation of CIP – Edition 1.1, December 2005, available at <<http://www.controlnet.org>>*

IECNORM.COM: Click to view the full PDF of IEC 61800-7-202:2007

Withdrawn

SOMMAIRE

AVANT-PROPOS	191
INTRODUCTION	193
1 Domaine d'application	197
2 Références normatives	197
3 Termes, définitions et abréviations	198
3.1 Termes et définitions	198
3.2 Abréviations	204
4 Généralités	204
4.1 Généralités	204
4.2 Modes de commande	205
4.2.1 Généralités	205
4.2.2 Méthodes de commande	205
4.2.3 Nomenclature de commande	206
4.2.4 Asservissement de position	206
4.2.5 Commande de vitesse	208
4.2.6 Commande d'accélération	210
4.2.7 Asservissement de couple	211
4.2.8 Pas d'asservissement	212
5 Types de données	213
5.1 Type de données - Présentation	213
5.2 Conventions	214
6 Profil de dispositif d'entraînement CIP	214
6.1 Modèle d'objet	214
6.1.1 Présentation de l'objet	214
6.1.2 Description de l'objet	215
6.2 Influence des objets sur le comportement	216
6.3 Définition des interfaces d'objet	216
6.4 Messages de connexion E-S	216
6.4.1 Généralités	216
6.4.2 Connexion E/S CIP Motion	217
6.4.3 Connexion Contrôleur - Dispositif	221
6.4.4 Connexion Dispositif - Contrôleur	233
6.4.5 Format de connexion de dispositif fixe	241
6.4.6 Temporisation de la connexion CIP Motion	242
6.5 Procédure de démarrage du dispositif d'entraînement	255
6.5.1 Généralités	255
6.5.2 Création d'une connexion E-S Motion	255
6.5.3 Configuration de l'Objet Axe de mouvement	255
6.5.4 Synchronisation temporelle	255
6.6 Visualisation du dispositif	257
6.7 Quality of Service (QoS) Ethernet	258
7 Objet Axe de mouvement	258
7.1 Considérations générales	258
7.1.1 Généralités	258
7.1.2 Présentation de l'objet	258

7.1.3	Abstraction de l'Objet Axe de mouvement.....	259
7.1.4	Objet Axe de commande de mouvement.....	260
7.1.5	Classification de commande d'entraînement.....	260
7.1.6	Obligatoire par rapport à Facultatif dans une mise en œuvre.....	261
7.2	Attributs de classe.....	268
7.2.1	Généralités.....	268
7.2.2	Sémantique.....	271
7.3	Attributs d'instance.....	275
7.3.1	Généralités.....	275
7.3.2	Attributs de configuration de commande de mouvement.....	276
7.3.3	Attributs du moteur.....	278
7.3.4	Attributs de réaction.....	286
7.3.5	Attributs Capture d'événement.....	293
7.3.6	Attributs de génération de référence de consigne.....	298
7.3.7	Attributs du mode de commande.....	303
7.3.8	Attributs Arrêt et freinage.....	315
7.3.9	Attributs Commande de bus c.c.....	319
7.3.10	Attributs de gestion de l'alimentation et de gestion thermique.....	320
7.3.11	Attributs Statut de l'axe.....	322
7.3.12	Attributs de statut d'exception, de défaut et d'alarme.....	326
7.3.13	Attributs de limite d'exception.....	332
7.3.14	Attribut de configuration d'action d'exception.....	335
7.3.15	Attributs de statut de défaut d'initialisation.....	336
7.3.16	Attributs Statut d'inhibition de démarrage.....	338
7.3.17	Attributs statistiques de l'axe.....	339
7.3.18	Attributs d'informations sur l'axe.....	340
7.3.19	Attributs E-S à usage général.....	341
7.3.20	Attributs Mode local.....	341
7.4	Services communs.....	342
7.4.1	Services pris en charge.....	342
7.4.2	Données spécifiques au service.....	343
7.5	Services spécifiques à l'objet.....	344
7.5.1	Services pris en charge.....	344
7.5.2	Données spécifiques au service.....	345
7.6	Comportement.....	356
7.6.1	Modèle d'état.....	356
7.6.2	Comportement d'état.....	360
7.6.3	Comportement de défaut et d'alarme.....	362
7.6.4	Comportement Inhibition de démarrage.....	364
7.6.5	Comportement de visualisation.....	364
7.6.6	Comportement de génération de consigne.....	368
7.6.7	Comportement de l'interface de réaction.....	374
7.6.8	Comportement de capture d'événement.....	377
7.6.9	Comportement du mode de commande.....	379
	Bibliographie.....	395
	Figure 1 – Structure de la CEI 61800-7.....	196

Figure 2 – Asservissement de position en boucle ouverte	207
Figure 3 – Asservissement de position en boucle fermée	208
Figure 4 – Commande de vitesse en boucle ouverte	209
Figure 5 – Commande de vitesse en boucle fermée	210
Figure 6 – Commande d'accélération	211
Figure 7 – Asservissement de couple	212
Figure 8 – Pas d'asservissement	213
Figure 9 – Modèle d'objet d'un dispositif d'entraînement CIP Motion	215
Figure 10 – Modèle de connexion E-S CIP Motion	217
Figure 11 – Format de connexion Contrôleur - Dispositif	218
Figure 12 – Format de connexion Dispositif - Contrôleur	219
Figure 13 – Canaux de connexion E-S CIP Motion	220
Figure 14 – Format de connexion Contrôleur - Dispositif CIP Motion	221
Figure 15 – En-tête de connexion	221
Figure 16 – Format de connexion	221
Figure 17 – En-tête de connexion	222
Figure 18 – Bloc de données d'instance	224
Figure 19 – En-tête de données d'instance	224
Figure 20 – En-tête de données cycliques	225
Figure 21 – Bloc de données d'écriture cycliques	230
Figure 22 – Bloc de données d'événement	230
Figure 23 – Bloc de données de service	233
Figure 24 – Format de connexion Contrôleur - Dispositif CIP Motion	233
Figure 25 – En-tête de connexion	233
Figure 26 – En-tête de connexion	235
Figure 27 – Ajustement des données de position réelles en fonction de la datation du dispositif	236
Figure 28 – Bloc de données d'instance	237
Figure 29 – En-tête de données d'instance	237
Figure 30 – Bloc de données cycliques	238
Figure 31 – Bloc de données de lecture cycliques	239
Figure 32 – Bloc de données d'événement	239
Figure 33 – Bloc de données de service	241
Figure 34 – Format de connexion Contrôleur – Dispositif fixe (taille fixe = 16 octets)	242
Figure 35 – Format de connexion Dispositif – Contrôleur fixe (taille fixe = 16 octets)	242
Figure 36 – Modèle de temporisation de la connexion E-S CIP Motion	243
Figure 37 – Temporisation de connexion Contrôleur – Dispositif avec interpolation fine	245
Figure 38 – Temporisation de connexion Contrôleur – Dispositif avec périodes d'actualisation inégales	247
Figure 39 – Utilisation de la datation pour ajuster la position réelle à la base de temps du contrôleur	248
Figure 40 – Coordination des deux dispositifs d'entraînement avec différentes périodes d'actualisation	250
Figure 41 – Coordination de plusieurs axes de dispositif d'entraînement en cas de retard des paquets de connexion Contrôleur - Dispositif	252

Figure 42 – Propagation d'une variation brusque dans le temps.....	253
Figure 43 – Synchronisation de groupe de dispositifs d'entraînement CIP Motion	256
Figure 44 – Composants d'objet de l'architecture de mouvement CIP	259
Figure 45 – Format de données de connexion utilisées du contrôleur	272
Figure 46 – Format de données de connexion générées du contrôleur.....	272
Figure 47 – Champ du bit Mode de commande	277
Figure 48 – IEEE par modèle de moteur phasé	278
Figure 49 – Champ du bit Configuration de réaction.....	293
Figure 50 – Champ de mot Commande de vérification d'événement.....	296
Figure 51 – Champ de mot Statut de vérification d'événement	297
Figure 52 – Champ de mot Commande d'interpolation	302
Figure 53 – Format de demande Extraction de la liste d'attribut d'axe	345
Figure 54 – Format de réponse Extraction de la liste d'attribut d'axe.....	346
Figure 55 – Format de réponse Extraction de la liste d'attribut d'axe – Exemple 1	347
Figure 56 – Format de réponse Extraction de la liste d'attribut d'axe – Exemple 2	347
Figure 57 – Format de réponse Extraction de la liste d'attribut d'axe – Exemple 3	347
Figure 58 – Format de réponse Extraction de la liste d'attribut d'axe – Exemple 4	348
Figure 59 – Format de demande Définition de la liste d'attribut d'axe.....	348
Figure 60 – Demande Définition de la liste d'attribut d'axe – Exemple 1	349
Figure 61 – Demande Définition de la liste d'attribut d'axe – Exemple 2.....	349
Figure 62 – Demande Définition de la liste d'attribut d'axe – Exemple 3.....	350
Figure 63 – Format de réponse Définition de la liste d'attribut d'axe.....	350
Figure 64 – Format de demande Définition de la liste d'écriture cyclique.....	350
Figure 65 – Format de réponse Définition de la liste d'écriture cyclique	351
Figure 66 – Format de demande Définition de la liste de lecture cyclique.....	351
Figure 67 – Format de réponse Définition de la liste de lecture cyclique	352
Figure 68 – Modèle d'état de l'Objet Axe de mouvement.....	357
Figure 69 – Générateur de consigne	370
Figure 70 – Voies de réaction 1 et 2	375
Figure 71 – Voies de réaction 3 et 4	376
Figure 72 – Fonctionnalité de capture d'événement	378
Figure 73 – Pas d'asservissement (Réaction uniquement)	380
Figure 74 – Asservissement de position en boucle fermée	382
Figure 75 – Commande de vitesse en boucle fermée	384
Figure 76 – Contrôle de fréquence en boucle ouverte	388
Figure 77 – Commande d'accélération	389
Figure 78 – Asservissement de couple.....	390
Figure 79 – Commande vectorielle de courant en boucle fermée	393
Tableau 1 – Types de données	213
Tableau 2 – Objets présents dans un dispositif d'entraînement CIP Motion.....	215
Tableau 3 – Influence de l'objet sur le comportement.....	216
Tableau 4 – Interfaces d'objet.....	216

Tableau 5 – Ensemble de données temporelles	223
Tableau 6 – Commande d'axe.....	226
Tableau 7 – Ensemble de données de consigne	226
Tableau 8 – Mise en correspondance de l'élément de données de consigne avec l'attribut d'Objet Axe de mouvement.....	227
Tableau 9 – Ensemble de données réelles.....	227
Tableau 10 – Mise en correspondance de l'élément de données réelles avec l'attribut d'Objet Axe de mouvement	228
Tableau 11 – Configuration de l'ensemble de données d'état	228
Tableau 12 – Commande d'interpolation	229
Tableau 13 – Réponse d'axe.....	238
Tableau 14 – Type d'événement	240
Tableau 15 – Propagation d'une variation brusque dans le temps (exemple 1).....	253
Tableau 16 – Propagation d'une variation brusque dans le temps (exemple 2).....	254
Tableau 17 – Composants de visualisation CIP Motion	257
Tableau 18 – Mise en œuvre d'attribut d'instance par rapport au Code de commande de dispositif	262
Tableau 19 – Attributs de classe de l'Objet Axe de mouvement	269
Tableau 20 – Définitions de bit Nœud de contrôle.....	272
Tableau 21 – Définitions du bit Statut du nœud.....	273
Tableau 22 – Définitions du code Défauts de nœud.....	274
Tableau 23 – Définitions du code Alarmes de nœud	274
Tableau 24 – Unité dynamique par rapport à la Configuration de réaction.....	276
Tableau 25 – Attributs de configuration de commande de mouvement	276
Tableau 26 – Définitions de l'énumération du champ Commande de moteur	277
Tableau 27 – Définitions de l'énumération du champ Méthode de commande	277
Tableau 28 – Attributs généraux d'informations sur le moteur	279
Tableau 29 – Attributs généraux de configuration du moteur.....	279
Tableau 30 – Attributs généraux de configuration du moteur à aimant permanent.....	282
Tableau 31 – Attributs généraux de configuration du moteur rotatif.....	283
Tableau 32 – Attributs généraux de configuration du moteur linéaire	283
Tableau 33 – Attributs de configuration du moteur à aimant permanent rotatif	284
Tableau 34 – Attributs de configuration du moteur à aimant permanent linéaire	285
Tableau 35 – Attributs de configuration du moteur à induction	285
Tableau 36 – Attributs généraux d'informations sur la réaction	287
Tableau 37 – Attributs de Configuration de réaction.....	287
Tableau 38 – Définitions de l'énumération du champ Sélection de réaction.....	293
Tableau 39 – Attributs d'événement	295
Tableau 40 – Définitions de bit Commande de vérification d'événement.....	296
Tableau 41 – Définitions de bit Statut de vérification d'événement	298
Tableau 42 – Attributs Signal du générateur de consigne.....	299
Tableau 43 – Attributs Configuration du générateur de consigne.....	300
Tableau 44 – Définition de l'énumération Actualisation cible de consigne	303
Tableau 45 – Attributs Signal de boucle de position	303

Tableau 46 – Attributs Configuration de boucle de position	304
Tableau 47 – Attributs Signal de boucle de vitesse	305
Tableau 48 – Attributs Configuration de boucle de vitesse	306
Tableau 49 – Attributs Signal d'accélération	308
Tableau 50 – Attributs Signal de référence de couple/force.....	308
Tableau 51 – Attributs Configuration de référence de couple/force	309
Tableau 52 – Attributs Signal de boucle de courant	310
Tableau 53 – Attributs Configuration de boucle de courant	312
Tableau 54 – Attributs Signal de contrôle de fréquence	313
Tableau 55 – Attributs Configuration de contrôle de fréquence	313
Tableau 56 – Attributs Sortie de dispositif d'entraînement.....	314
Tableau 57 – Attributs Arrêt/Freinage	315
Tableau 58 – Définitions de l'énumération Mode d'arrêt	317
Tableau 59 – Attributs Commande de bus c.c.	319
Tableau 60 – Attributs de statut de gestion de l'alimentation et de gestion thermique	320
Tableau 61 – Attributs de configuration de gestion de l'alimentation et de gestion thermique	321
Tableau 62 – Attributs Statut de l'axe	323
Tableau 63 – Définitions du bit Statut de l'axe	324
Tableau 64 – Définitions du bit Statut d'E-S de l'axe	325
Tableau 65 – Définitions du bit Ensemble de données d'état.....	325
Tableau 66 – Attributs de statut d'exception, de défaut et d'alarme	326
Tableau 67 – Tableau d'exception standard.....	330
Tableau 68 – Attributs d'informations de tolérance de fabrication d'exception	332
Tableau 69 – Attributs de configuration de limite utilisateur d'exception	334
Tableau 70 – Attribut de configuration d'action d'exception	335
Tableau 71 – Définitions du bit Action d'exception	335
Tableau 72 – Attributs Statut de défaut d'initialisation	336
Tableau 73 – Tableau de défaut d'initialisation standard	338
Tableau 74 – Attributs Statut d'inhibition de démarrage	338
Tableau 75 – Tableau Inhibition de démarrage standard	339
Tableau 76 – Attributs statistiques	340
Tableau 77 – Attributs d'informations sur l'axe	340
Tableau 78 – Attributs E-S à usage général	341
Tableau 79 – Attributs de configuration du mode local	342
Tableau 80 – Objet Axe de mouvement – Services communs.....	342
Tableau 81 – Structure des données de demande Group_Sync	343
Tableau 82 – Structure des données de réponse Group_Sync	343
Tableau 83 – Objet Axe de mouvement – Services spécifiques à l'objet.....	344
Tableau 84 – Structure de données de la demande Exécution des essais moteur	353
Tableau 85 – Structure de données de la réponse Extraction des essais moteur	353
Tableau 86 – Structure de données de la demande Exécution des essais d'inertie	353
Tableau 87 – Structure de données de la réponse Extraction des essais d'inertie	354

Tableau 88 – Structure de données de la demande Exécution des essais de connexion	355
Tableau 89 – Structure de données de la réponse Extraction des essais de connexion.....	355
Tableau 90 – Code de demande de Commande d'axe.....	357
Tableau 91 – Codes d'acquiescement de réponse d'axe.....	358
Tableau 92 – Critères d'exécution du fonctionnement demandé	358
Tableau 93 – Cycle de demande de commande d'axe réussie.....	359
Tableau 94 – Cycle de demande de commande d'axe non réussie	359
Tableau 95 – Mise en correspondance du statut de l'axe avec l'Objet Identité grâce au comportement de la DEL.....	365
Tableau 96 – Comportement de l'affichage à sept segments du dispositif CIP Motion	366
Tableau 97 – Comportement de l'affichage alphanumérique à plusieurs caractères CIP Motion	366
Tableau 98 – Comportement de l'affichage alphanumérique à plusieurs caractères multi-axe.....	368

Withdrawing
IECNORM.COM: Click to view the full PDF of IEC 61800-7-202:2007

COMMISSION ÉLECTROTECHNIQUE INTERNATIONALE

ENTRAÎNEMENTS ÉLECTRIQUES DE PUISSANCE À VITESSE VARIABLE –

**Partie 7-202: Interface générique et utilisation de profils
pour les entraînements électriques de puissance –
Spécification du profil de type 2**

AVANT-PROPOS

- 1) La Commission Electrotechnique Internationale (CEI) est une organisation mondiale de normalisation composée de l'ensemble des comités électrotechniques nationaux (Comités nationaux de la CEI). La CEI a pour objet de favoriser la coopération internationale pour toutes les questions de normalisation dans les domaines de l'électricité et de l'électronique. A cet effet, la CEI – entre autres activités – publie des Normes internationales, des Spécifications techniques, des Rapports techniques, des Spécifications accessibles au public (PAS) et des Guides (ci-après dénommés "Publication(s) de la CEI"). Leur élaboration est confiée à des comités d'études, aux travaux desquels tout Comité national intéressé par le sujet traité peut participer. Les organisations internationales, gouvernementales et non gouvernementales, en liaison avec la CEI, participent également aux travaux. La CEI collabore étroitement avec l'Organisation Internationale de Normalisation (ISO), selon des conditions fixées par accord entre les deux organisations.
- 2) Les décisions ou accords officiels de la CEI concernant les questions techniques représentent, dans la mesure du possible, un accord international sur les sujets étudiés, étant donné que les Comités nationaux de la CEI intéressés sont représentés dans chaque comité d'études.
- 3) Les Publications de la CEI se présentent sous la forme de recommandations internationales et sont agréées comme telles par les Comités nationaux de la CEI. Tous les efforts raisonnables sont entrepris afin que la CEI s'assure de l'exactitude du contenu technique de ses publications; la CEI ne peut pas être tenue responsable de l'éventuelle mauvaise utilisation ou interprétation qui en est faite par un quelconque utilisateur final.
- 4) Dans le but d'encourager l'uniformité internationale, les Comités nationaux de la CEI s'engagent, dans toute la mesure possible, à appliquer de façon transparente les Publications de la CEI dans leurs publications nationales et régionales. Toutes divergences entre toutes Publications de la CEI et toutes publications nationales ou régionales correspondantes doivent être indiquées en termes clairs dans ces dernières.
- 5) La CEI elle-même ne fournit aucune attestation de conformité. Des organismes de certification indépendants fournissent des services d'évaluation de conformité et, dans certains secteurs, accèdent aux marques de conformité de la CEI. La CEI n'est responsable d'aucun des services effectués par les organismes de certification indépendants.
- 6) Tous les utilisateurs doivent s'assurer qu'ils sont en possession de la dernière édition de cette publication.
- 7) Aucune responsabilité ne doit être imputée à la CEI, à ses administrateurs, employés, auxiliaires ou mandataires, y compris ses experts particuliers et les membres de ses comités d'études et des Comités nationaux de la CEI, pour tout préjudice causé en cas de dommages corporels et matériels, ou de tout autre dommage de quelque nature que ce soit, directe ou indirecte, ou pour supporter les coûts (y compris les frais de justice) et les dépenses découlant de la publication ou de l'utilisation de cette Publication de la CEI ou de toute autre Publication de la CEI, ou au crédit qui lui est accordé.
- 8) L'attention est attirée sur les références normatives citées dans cette publication. L'utilisation de publications référencées est obligatoire pour une application correcte de la présente publication.
- 9) L'attention est attirée sur le fait que certains des éléments de la présente Publication de la CEI peuvent faire l'objet de droits de propriété intellectuelle ou de droits analogues. La CEI ne saurait être tenue pour responsable de ne pas avoir identifié de tels droits de propriété et de ne pas avoir signalé leur existence.

La Commission électrotechnique internationale (CEI) attire l'attention sur le fait qu'il est déclaré que la conformité avec les dispositions du présent document peut impliquer l'utilisation d'un brevet intéressant ce qui suit:

Numéro de série de la publication / Application	Dé détenteur	Titre
US 11/241,539	[RA]	Time Stamped Motion Control Network Protocol That Enables Balanced Single Cycle Timing and Utilization of Dynamic Data Structures

La CEI ne prend pas position quant à la preuve, à la validité et à la portée de ces droits de propriété.

Le détenteur de ces droits de propriété a donné l'assurance à la CEI qu'il consent à négocier des licences avec des demandeurs du monde entier, à des termes et conditions raisonnables et non discriminatoires. A ce propos, la déclaration du détenteur des droits de propriété est enregistrée à la CEI. Des informations peuvent être demandées à

[RA]	Rockwell Automation, Inc. 1201 S. Second Street Milwaukee, WI 53204 USA Attention: Intellectual Property Dept.

L'attention est attirée sur le fait que certains des éléments du présent document peuvent faire l'objet de droits de propriété intellectuelle ou de droits analogues. La CEI ne saurait être tenue pour responsable de ne pas avoir identifié de tels droits de propriété et de ne pas avoir signalé leur existence.

La Norme internationale CEI 61800-7-202 a été établie par le sous-comité 22G: Systèmes d'entraînement électrique à vitesse variable, comprenant des convertisseurs à semi-conducteurs, du comité d'études 22 de la CEI: Systèmes et équipements électroniques de puissance.

La présente version bilingue (2013-05) correspond à la version anglaise monolingue publiée en 2007-11.

Le texte anglais de cette norme est issu des documents 22G/184/FDIS et 22G/192/RVD.

Le rapport de vote 22G/192/RVD donne toute information sur le vote ayant abouti à l'approbation de cette norme.

La version française de cette norme n'a pas été soumise au vote. Cette publication a été rédigée selon les Directives ISO/CEI, Partie 2.

Une liste de toutes les parties de la série CEI 61800, sous le titre général *Entraînements électriques de puissance à vitesse variable*, peut être consultée sur le site web de la CEI.

Le comité a décidé que le contenu de cette publication ne sera pas modifié avant la date de stabilité indiquée sur le site web de la CEI sous "<http://webstore.iec.ch>" dans les données relatives à la publication recherchée. A cette date, la publication sera

- reconduite;
- supprimée;
- remplacée par une édition révisée, ou
- amendée.

IMPORTANT – Le logo "colour inside" qui se trouve sur la page de couverture de cette publication indique qu'elle contient des couleurs qui sont considérées comme utiles à une bonne compréhension de son contenu. Les utilisateurs devraient, par conséquent, imprimer cette publication en utilisant une imprimante couleur.

INTRODUCTION

La série CEI 61800 est destinée à fournir un ensemble commun de spécifications dédiées aux entraînements électriques de puissance à vitesse variable.

La CEI 61800-7 décrit une interface générique entre les systèmes de commande et les entraînements électriques de puissance. Cette interface peut être intégrée au système de commande. Le système de commande proprement dit peut également être situé dans le dispositif d'entraînement (parfois appelé "dispositif d'entraînement intelligent").

Il existe un grand nombre d'interfaces physiques disponibles (entrées et sorties analogiques et numériques, interfaces séries et parallèles, bus de terrain et réseaux). Les profils établis sur des interfaces physiques spécifiques sont déjà définis pour certains domaines d'application (par exemple, commande de mouvement) et certaines classes de dispositifs (par exemple, dispositifs d'entraînement classiques, positionneur). Les applications des interfaces de programmes de commande et de programmeurs d'application associées sont exclusives et varient dans une large mesure.

La CEI 61800-7 définit un ensemble de fonctions, paramètres et diagrammes d'états communs de commande d'entraînement ou une description de séquences d'opérations à mettre en correspondance avec les profils.

La CEI 61800-7 fournit une procédure d'accès aux fonctions et données d'un dispositif d'entraînement, indépendante du profil d'entraînement et de l'interface de communication employés. Il s'agit de définir un modèle d'entraînement commun comportant des fonctions génériques et des objets pouvant être mis en correspondance sur des interfaces de communication différentes. Ceci permet de prévoir des applications communes de commande de mouvement (ou applications de commande de vitesse ou de commande d'entraînement) dans les contrôleurs sans aucune connaissance spécifique de la mise en œuvre du dispositif d'entraînement.

Il y a plusieurs raisons de définir une interface générique:

Pour un constructeur de dispositif d'entraînement

- Assistance plus aisée des intégrateurs de systèmes
- Description plus aisée des fonctions d'entraînement du fait d'une terminologie commune
- Le choix des dispositifs d'entraînement ne dépend pas de la disponibilité d'une assistance spécifique

Pour un constructeur de dispositif de commande

- Aucune influence de la technologie de bus
- Intégration aisée des dispositifs
- Indépendance par rapport à un fournisseur de dispositifs d'entraînement

Pour un intégrateur de systèmes (modules de construction, machines, usines, etc.)

- Effort d'intégration moindre des dispositifs
- Méthode intelligible unique de modélisation
- Indépendance par rapport à la technologie de bus

Concevoir une application de commande de mouvement avec plusieurs dispositifs d'entraînement différents et un système de commande spécifique nécessite un effort certain. Les tâches de mise en œuvre des logiciels systèmes et de compréhension de la description fonctionnelle des composants individuels peuvent contribuer à l'épuisement des ressources d'un projet. Dans certains cas, les dispositifs d'entraînement ne partagent pas la même interface physique. Certains dispositifs de commande prennent simplement en charge une interface unique qui n'est pas prise en charge par un dispositif d'entraînement spécifique. Par

ailleurs, les fonctions et les structures de données sont spécifiées avec des incompatibilités. Cela exige de l'intégrateur de systèmes d'établir des interfaces spéciales dédiées aux logiciels d'application et il convient que cette opération ne relève pas de sa responsabilité.

Certaines applications nécessitent de pouvoir permuter des dispositifs, voire intégrer de nouveaux dispositifs dans une configuration existante. Elles sont également confrontées à différentes solutions incompatibles. Les efforts visant à adopter une solution relative à un profil d'entraînement et aux extensions spécifiques au constructeur peuvent se révéler inacceptables. Ceci réduit le degré de liberté concernant le choix d'un dispositif le mieux adapté à cette application de sélection du dispositif disponible pour une interface physique spécifique et pris en charge par le contrôleur.

La CEI 61800-7-1 est divisée en une partie générique et en plusieurs annexes (comme l'illustre la Figure 1). Les types de profil d'entraînement pour CiA 402¹, CIP MotionTM², PROFIdrive³ et SERCOS InterfaceTM⁴ sont mis en correspondance avec l'interface générique dans l'annexe correspondante. Les annexes ont été soumises par des organismes internationaux indépendants spécialisés dans les réseaux ou les bus de terrain, et responsables du contenu de l'annexe qui y est associée, ainsi que de l'utilisation des marques connexes.

La présente partie de la CEI 61800-7 spécifie le type de profil 2 (CIP MotionTM).

Les types de profil 1, 3 et 4 sont spécifiés dans la CEI 61800-7-201, dans la CEI 61800-7-203 et dans la CEI 61800-7-204.

La CEI 61800-7-301, la CEI 61800-7-302, la CEI 61800-7-303 et la CEI 61800-7-304 spécifient la ou les méthodes de mise en correspondance des types de profil 1, 2, 3 et 4 avec différentes technologies de réseau (CANopen⁵, EtherCATTM⁶, Ethernet PowerlinkTM⁷, DeviceNetTM⁸, ControlNetTM⁹, EtherNet/IPTM¹⁰, PROFIBUS¹¹, PROFINET¹² et SERCOS Interface).

1 CiA 402 est une marque de CAN in Automation, e.V. Cette information est fournie pour la commodité des utilisateurs de la présente norme internationale et ne constitue en aucun cas un entérinement par la CEI du détenteur de la marque ou de l'un quelconque de ses produits. La conformité à ce profil n'implique pas l'utilisation de la marque CiA 402.

2 CIP MotionTM est une marque de Open DeviceNet Vendor Association, Inc. Cette information est fournie pour la commodité des utilisateurs de la présente Norme internationale et ne constitue en aucun cas un entérinement par la CEI du détenteur de la marque ou de l'un quelconque de ses produits. La conformité à ce profil n'implique pas l'utilisation de la marque CIP MotionTM. L'utilisation de la marque CIP MotionTM nécessite l'autorisation de Open DeviceNet Vendor Association, Inc.

3 PROFIdrive est une marque de PROFIBUS International. Cette information est fournie pour la commodité des utilisateurs de la présente Norme internationale et ne constitue en aucun cas un entérinement par la CEI du détenteur de la marque ou de l'un quelconque de ses produits. La conformité à ce profil n'implique pas l'utilisation de la marque PROFIdrive. L'utilisation de la marque PROFIdrive nécessite l'autorisation de PROFIBUS International.

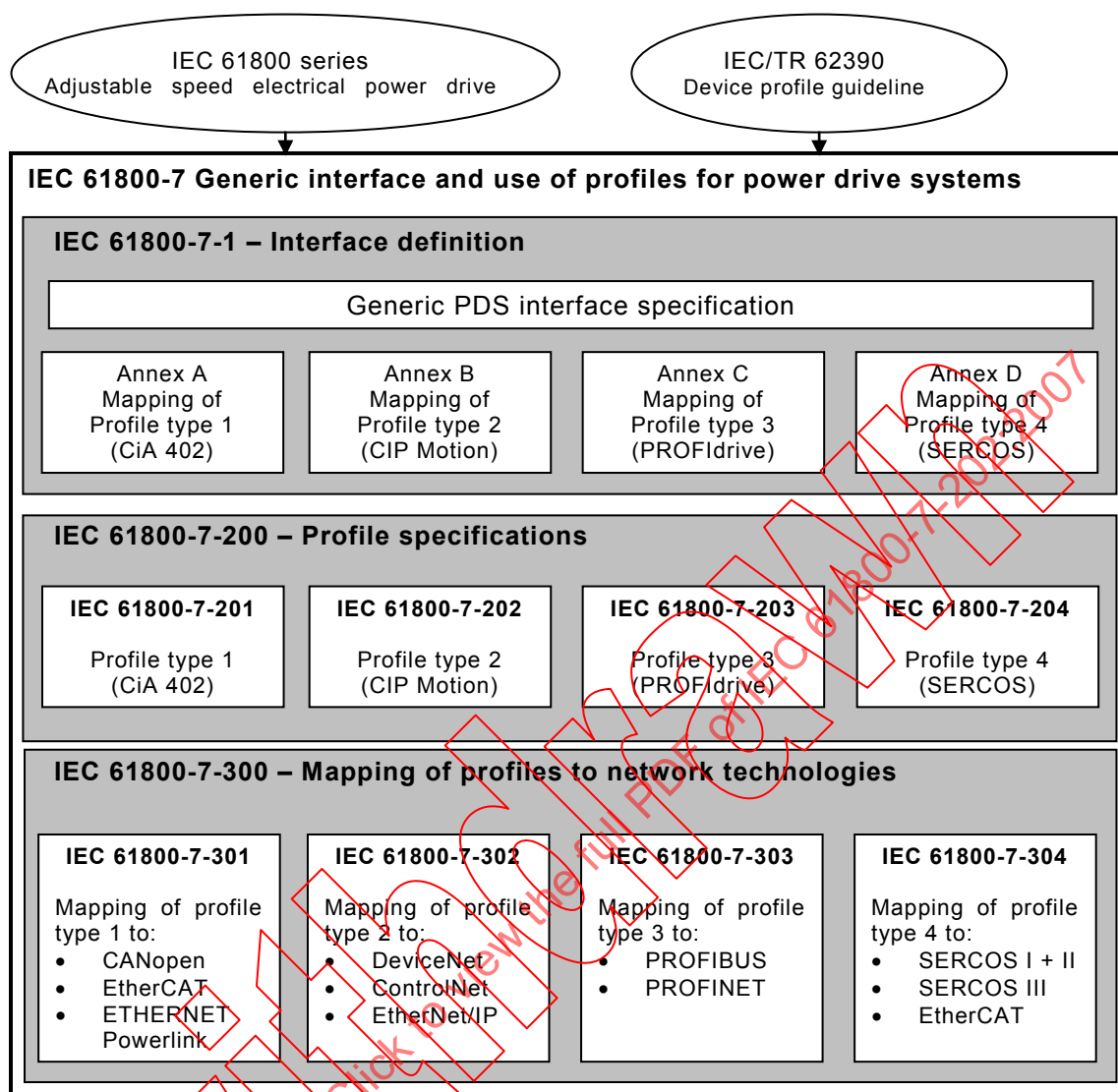
4 SERCOSTM et SERCOS interfaceTM sont des marques de SERCOS International e.V. Cette information est fournie pour la commodité des utilisateurs de la présente Norme internationale et ne constitue en aucun cas un entérinement par la CEI du détenteur de la marque ou de l'un quelconque de ses produits. La conformité à ce profil n'implique pas l'utilisation de la marque SERCOS et de l'interface SERCOS. L'utilisation de la marque SERCOS et de l'interface SERCOS nécessite l'autorisation de leur détenteur.

5 CANopen est l'acronyme de Controller Area Network *open* (*Gestionnaire de réseau de communication ouvert*) et fait référence à l'EN 50325-4.

6 EtherCATTM est une marque de Beckhoff, Verl. Cette information est fournie pour la commodité des utilisateurs de la présente Norme internationale et ne constitue en aucun cas un entérinement par la CEI du détenteur de la marque ou de l'un quelconque de ses produits. La conformité à ce profil n'implique pas l'utilisation de la marque EtherCATTM. L'utilisation de la marque EtherCATTM nécessite l'autorisation de son détenteur.

7 Ethernet PowerlinkTM est une marque de B&R, le contrôle de son utilisation est confié à l'organisme à but non lucratif EPSG. Cette information est fournie pour la commodité des utilisateurs de la présente Norme internationale et ne constitue en aucun cas un entérinement par la CEI du détenteur de la marque ou de l'un quelconque de ses produits. La conformité à ce profil n'implique pas l'utilisation de la marque Ethernet PowerlinkTM. L'utilisation de la marque Ethernet PowerlinkTM nécessite l'autorisation de son détenteur.

8 DeviceNetTM est une marque de Open DeviceNet Vendor Association, Inc. Cette information est fournie pour la commodité des utilisateurs de la présente Norme internationale et ne constitue en aucun cas un entérinement



par la CEI du détenteur de la marque ou de l'un quelconque de ses produits. La conformité à ce profil n'implique pas l'utilisation de la marque DeviceNet™. L'utilisation de la marque DeviceNet™ nécessite l'autorisation de Open DeviceNet Vendor Association, Inc.

- 9 ControlNet™ est une marque de ControlNet International, Ltd. Cette information est fournie pour la commodité des utilisateurs de la présente Norme internationale et ne constitue en aucun cas un entérinement par la CEI du détenteur de la marque ou de l'un quelconque de ses produits. La conformité à ce profil n'implique pas l'utilisation de la marque ControlNet™. L'utilisation de la marque ControlNet™ nécessite l'autorisation de ControlNet International, Ltd.
- 10 EtherNet/IP™ est une marque de ControlNet International, Ltd. et de Open DeviceNet Vendor Association, Inc. Cette information est fournie pour la commodité des utilisateurs de la présente Norme internationale et ne constitue en aucun cas un entérinement par la CEI du détenteur de la marque ou de l'un quelconque de ses produits. La conformité à ce profil n'implique pas l'utilisation de la marque EtherNet/IP™. L'utilisation de la marque EtherNet/IP™ nécessite l'autorisation de ControlNet International, Ltd. ou de Open DeviceNet Vendor Association, Inc.
- 11 PROFIBUS est une marque de PROFIBUS International. Cette information est fournie pour la commodité des utilisateurs de la présente norme internationale et ne constitue en aucun cas un entérinement par la CEI du détenteur de la marque ou de l'un quelconque de ses produits. La conformité à ce profil n'implique pas l'utilisation de la marque PROFIBUS. L'utilisation de la marque PROFIBUS nécessite l'autorisation de PROFIBUS International.
- 12 PROFINET est une marque de PROFIBUS International. Cette information est fournie pour la commodité des utilisateurs de la présente norme internationale et ne constitue en aucun cas un entérinement par la CEI du détenteur de la marque ou de l'un quelconque de ses produits. La conformité à ce profil n'implique pas l'utilisation de la marque PROFINET. L'utilisation de la marque PROFINET nécessite l'autorisation de PROFIBUS International.

Légende

Anglais	Français
IEC 61800 series Adjustable speed electrical power drive	Série CEI 61800 Entraînement électrique de puissance à vitesse variable
IEC/TR 62390 Device profile guideline	IEC/TR 62390 Device profile guideline (disponible en anglais uniquement)
IEC 61800-7 Generic interface and use of profiles for power drive systems	IEC 61800-7 Generic interface and use of profiles for power drive systems (disponible en anglais uniquement)
IEC 61800-7-1 Interface definition	IEC 61800-7-1 Interface definition (disponible en anglais uniquement)
Generic PDS interface specification	Spécification d'interface PDS générique
Annex A, Mapping of Profile type 1 (CiA 402)	Annexe A, Mise en correspondance de profil de type 1 (CiA 402)
Annex B, Mapping of Profile type 2 (CIP Motion)	Annexe B, Mise en correspondance de profil de type 2 (CIP Motion)
Annex C, Mapping of Profile type 3 (PROFIdrive)	Annexe C, Mise en correspondance de profil de type 3 (PROFIdrive)
Annex D, Mapping of Profile type 4 (SERCOS)	Annexe D, Mise en correspondance de profil de type 4 (SERCOS)
IEC 61800-7-200 – Profile specifications	IEC 61800-7-200 – Profile specifications (disponible en anglais uniquement)
IEC 61800-7-201 Profile type 1 (CiA 402)	CEI 61800-7-201 Profil de type 1 (CiA 402)
IEC 61800-7-202 Profile type 2 (CIP Motion)	CEI 61800-7-202 Profil de type 2 (CIP Motion)
IEC 61800-7-203 Profile type 3 (PROFIdrive)	CEI 61800-7-203 Profil de type 3 (PROFIdrive)
IEC 61800-7-204 Profile type 4 (PROFIdrive)	CEI 61800-7-204 Profil de type 4 (SERCOS)
IEC 61800-7-300 – Mapping of profiles to network technologies	IEC 61800-7-300 – Mapping of profiles to network technologies (disponible en anglais uniquement)
IEC 61800-7-301 Mapping of profile type 1 to CANopen EtherCAT ETHERNET Powerlink	CEI 61800-7-301 Mise en correspondance du profil de type 1 avec : CANopen EtherCAT ETHERNET Powerlink
IEC 61800-7-302 Mapping of profile type 2 to DeviceNet ControlNet EtherNet/IP	CEI 61800-7-302 Mise en correspondance du profil de type 2 avec : DeviceNet ControlNet EtherNet/IP
IEC 61800-7-303 Mapping of profile type 3 to PROFIBUS PROFINET	CEI 61800-7-303 Mise en correspondance du profil de type 3 avec : PROFIBUS PROFINET
IEC 61800-7-304 Mapping of profile type 4 to SERCOS I + II SERCOS III EtherCAT	CEI 61800-7-304 Mise en correspondance du profil de type 4 avec : SERCOS I + II SERCOS III EtherCAT

Figure 1 – Structure de la CEI 61800-7

ENTRAÎNEMENTS ÉLECTRIQUES DE PUISSANCE À VITESSE VARIABLE –

Partie 7-202: Interface générique et utilisation de profils pour les entraînements électriques de puissance – Spécification du profil de type 2

1 Domaine d'application

La CEI 61800-7 spécifie les profils dédiés aux entraînements électriques de puissance (PDS) et leur mise en correspondance avec les systèmes de communication existants grâce à un modèle d'interface générique.

Les fonctions spécifiées dans la présente partie de la CEI 61800-7 ne sont pas destinées à assurer la sécurité fonctionnelle. Ceci exige l'application de mesures supplémentaires conformes aux normes, conventions et lois pertinentes.

La présente partie de la CEI 61800-7 spécifie le type de profil 2 (CIP Motion™) pour les entraînements électriques de puissance (PDS). Le type de profil 2 peut être mis en correspondance avec différentes technologies de réseau de communication.

2 Références normatives

Les documents de référence suivants sont indispensables pour l'application du présent document. Pour les références datées, seule l'édition citée s'applique. Pour les références non datées, la dernière édition du document de référence s'applique (y compris les éventuels amendements).

CEI 60204-1, *Sécurité des machines – Équipement électrique des machines – Partie 1: Règles générales*

CEI 61131-3, *Automates programmables – Partie 3: Langages de programmation*

IEC 61158-5-2, *Industrial communication networks – Fieldbus specifications – Part 5-2 (Ed.1.0): Application layer service definition – Type 2 elements* (disponible en anglais uniquement)

IEC 61158-6-2, *Industrial communication networks – Fieldbus specifications – Part 6-2 (Ed.1.0): Application layer protocol specification – Type 2 elements* (disponible en anglais uniquement)

IEC 61588:2004, *Precision clock synchronization protocol for networked measurement and control systems* (disponible en anglais uniquement)

IEC 61800-7 (toutes les parties), *Adjustable speed electrical power drive systems – Generic interface and use of profiles for power drive systems* (disponible en anglais uniquement)

IEC 61800-7-1, *Adjustable speed electrical power drive systems – Part 7-1: Generic interface and use of profiles for power drive systems – Interface definition* (disponible en anglais uniquement)

IEEE 802.1Q, *IEEE standards for local and metropolitan area networks – Virtual bridged local area networks*

3 Termes, définitions et abréviations

3.1 Termes et définitions

Pour les besoins du présent document, les termes et définitions suivants s'appliquent.

3.1.1

algorithme

séquence finie d'instructions complètement déterminée par laquelle la valeur des données de sortie peut être calculée à partir de la valeur des données d'entrée

[VEI 351-21-37]

3.1.2

application

élément fonctionnel logiciel spécifique à la résolution d'un problème en termes de mesure et de commande de procédés industriels

NOTE Une application peut être répartie entre les ressources, et peut communiquer avec d'autres applications.

[CEI/TR 62390:2005, 3.1.2, modifiée]

3.1.3

mode d'application

type d'application pouvant être sollicité par un PDS

NOTE Les différents modes d'application reflètent la boucle de régulation dédiée à l'asservissement de couple ou de position, à la commande de vitesse ou à d'autres applications telles que le retour à la position de référence.

3.1.4

attribut

propriété ou caractéristique d'une entité

[CEI/TR 62390:2005, 3.1.3]

3.1.5

axe

élément logique d'un système de commande de mouvement qui représente une certaine forme de mouvement

NOTE 1 Les axes peuvent être rotatifs ou linéaires, physiques ou virtuels, commandés ou simplement observés.

NOTE 2 Un axe physique peut contenir un ou plusieurs des éléments suivants: un capteur de mouvement, une structure de commande de mouvement, un amplificateur de puissance et un actionneur de mouvement.

3.1.6

régulateur de bus

toute méthode permettant de limiter la hausse du niveau de tension de bus en c.c. suite à la décélération d'un moteur

3.1.7

CIP Motion™¹³

extensions des services et du protocole CIP afin de prendre en charge la commande de mouvement sur les réseaux CIP

¹³ CIP Motion™ et CIP Sync™ sont des marques de Open DeviceNet Vendor Association, Inc. Cette information est fournie pour la commodité des utilisateurs de la présente Norme internationale et ne constitue en aucun cas un entérinement par la CEI du détenteur de la marque ou de l'un quelconque de ses produits. La conformité à ce profil n'implique pas l'utilisation des marques CIP Motion™ ou CIP Sync™. L'utilisation de la marque CIP Motion™ ou CIP Sync™ nécessite l'autorisation de Open DeviceNet Vendor Association, Inc.

3.1.8**dispositif d'entraînement CIP (Motion)**

dispositif d'entraînement conforme à la spécification CIP (Motion)

3.1.9**profil d'entraînement CIP Motion**

ensemble d'objets permettant de mettre en œuvre un dispositif d'entraînement CIP Motion comprenant l'Objet Axe de mouvement, ainsi que des objets de prise en charge standard tels que l'Objet Identité et l'Objet Synchronisation temporelle

3.1.10**connexion E/S CIP Motion**

connexion CIP bidirectionnelle périodique de classe 1 entre un contrôleur et un dispositif d'entraînement, définie comme partie intégrante de la spécification de CIP Motion

3.1.11**CIP Sync™¹³**

extensions des services et du protocole CIP visant à intégrer la fonctionnalité de synchronisation temporelle définie dans la CEI 61588:2004 dans un réseau CIP (voir "Objet Synchronisation temporelle" (Time Sync Object) dans la CEI 61158-5-2 et la CEI 61158-6-2)

3.1.12**classe**

description d'un ensemble d'objets qui partagent les mêmes attributs, opérations, méthodes, relations et sémantique

[ISO/CEI 19501, modifiée]

3.1.13**boucle fermée**

méthodes de commande dans lesquelles un signal de réaction d'un type donné est utilisé pour entraîner la dynamique réelle du moteur afin de correspondre à la dynamique commandée par asservissement

NOTE Dans la plupart des cas, un dispositif de réaction littéral fournit ce signal, mais dans certains cas, le signal est dérivé de l'excitation du moteur (c'est-à-dire fonctionnement sans capteur).

3.1.14**commandes, consignes**

ensemble de commandes entre le programme de contrôle d'application et le PDS permettant de contrôler le comportement du PDS ou les éléments fonctionnels de celui-ci

NOTE 1 Les états ou les modes de fonctionnement reflètent le comportement.

NOTE 2 Les différentes commandes peuvent être représentées par un bit chacune.

3.1.15**commande, régulation**

action délibérée sur (ou dans) un processus, en vue d'atteindre des objectifs définis

[VEI 351-21-29, modifiée]

3.1.16**dispositif de commande/régulation**

unité physique contenant – dans un module/sous-ensemble ou dispositif – un programme d'application de commande du PDS

3.1.17**convertisseur**

dispositif qui convertit généralement une entrée c.a. en sortie c.c.

NOTE Un convertisseur est également appelé Alimentation d'entraînement. Dans le contexte d'un système d'entraînement, le convertisseur est chargé de convertir l'alimentation en courant alternatif en alimentation de bus à courant continu.

3.1.18

bloc de données cycliques

bloc de données en temps réel à priorité élevée transféré par une connexion CIP Motion de manière périodique

3.1.19

type de données

ensemble de valeurs associé à un ensemble d'opérations autorisées

[ISO/CEI 2382-15:1999, 15.04.01, modifiée]

3.1.20

dispositif

dispositif de terrain

entité physique indépendante sur réseau, d'un système d'automatisation industriel, capable d'exécuter des fonctions spécifiées dans un contexte particulier et délimitée par ses interfaces

[CEI 61499-1:2005, 3.30, modifiée]

entité qui exécute des fonctions de commande/régulation, activation et/ou détection et qui est en interface avec d'autres entités de ce type d'un système d'automatisation

[ISO 15745-1:2003, 3.11]

3.1.21

profil de dispositif

représentation d'un dispositif en termes de ses paramètres, ensembles de paramètres et comportement selon un modèle de dispositif qui décrit les données et le comportement du dispositif perçus par le biais d'un réseau

NOTE Cette définition est issue de la CEI/TS 61915, que l'on a étendue par l'ajout de la structure fonctionnelle du dispositif.

[CEI/TR 62390:2005, 3.19, modifiée]

3.1.22

dispositif d'entraînement

dispositif conçu pour commander la dynamique d'un moteur

3.1.23

bloc de données d'événement

bloc de données en temps réel à priorité moyenne transférées par une connexion CIP Motion uniquement après occurrence d'un événement spécifié

NOTE L'enregistrement et les marqueurs de transitions d'entrée constituent des événements d'entraînement typiques.

3.1.24

élément fonctionnel

entité de logiciel ou logiciel combiné au matériel, capable d'accomplir une fonction spécifiée d'un dispositif

NOTE 1 Un élément fonctionnel comporte une interface et des associations à d'autres éléments fonctionnels et fonctions.

NOTE 2 Un élément fonctionnel peut être constitué de bloc(s) de fonctions, d'objet(s) ou de liste(s) de paramètres.

[CEI/TR 62390:2005, 3.1.12]

3.1.25

obtenir/lire

opération qui implique la recherche d'une valeur d'attribut à partir de la partie contrôleur de l'interface

3.1.26

données d'entrée

données transférées d'une source externe dans un dispositif, une ressource ou un élément fonctionnel

[CEI/TR 62390:2005, 3.1.14]

3.1.27

interface

frontière commune entre deux unités, définie par des caractéristiques fonctionnelles, des caractéristiques de signal ou d'autres caractéristiques selon le cas

[VEI 351-21-35, modifiée]

3.1.28

onduleur

dispositif qui convertit généralement une entrée c.c. en sortie c.a.

NOTE Un onduleur est également souvent appelé Amplificateur d'entraînement. Dans le contexte d'un système d'entraînement, l'onduleur est chargé de contrôler l'application de l'alimentation de bus à courant continu à un moteur à courant alternatif.

3.1.29

modèle

représentation mathématique ou physique d'un système ou d'un processus, basée, avec une précision suffisante, sur des lois connues, sur une identification ou sur des hypothèses spécifiées

[VEI 351-21-36]

3.1.30

mouvement

tout aspect de la dynamique d'un axe

NOTE Dans le contexte de la présente partie de la CEI 61800-7, cette définition ne se limite pas aux entraînements asservis, mais englobe toutes les formes de commande de moteur par entraînement.

3.1.31

objet Axe de mouvement (Motion Axis Object)

objet qui définit les attributs, services et comportement d'un axe de dispositif de mouvement en fonction de la spécification de CIP Motion

3.1.32

objet Axe de commande de mouvement (Motion control axis object)

objet qui définit les attributs, services et comportement d'un axe de contrôleur en fonction de la spécification de CIP Motion

3.1.33

boucle ouverte

méthodes de commande dans lesquelles il n'existe aucune application de réaction pour forcer la dynamique réelle du moteur afin de correspondre à la dynamique commandée

EXEMPLE Les exemples de commande à boucle ouverte sont les entraînements pas à pas et les dispositifs d'entraînement à fréquence variable.

3.1.34

mode de fonctionnement

caractérisation de la manière et du degré avec lequel l'opérateur humain intervient sur l'équipement de commande

[VEI 351-31-01]

3.1.35

données de sortie

données provenant d'un dispositif, d'une ressource ou d'un élément fonctionnel et transférées de ces derniers vers des systèmes externes

[CEI/TR 62390:2005, 3.1.21]

3.1.36

paramètre

élément de donnée qui représente les informations d'un dispositif qui peuvent être lues ou saisies dans un dispositif, par exemple, par le biais du réseau ou d'une IHM locale

NOTE 1 Adaptée de la CEI/TS 61915.

NOTE 2 Un paramètre est généralement caractérisé par un nom de paramètre, un type de données et une direction d'accès.

[CEI/TR 62390:2005, 3.1.22, modifiée]

3.1.37

profil

représentation d'une interface PDS en termes de ses paramètres, ensembles de paramètres et comportement selon un profil de communication et un profil de dispositif

3.1.38

bloc de données de service

bloc de données en temps réel de priorité moins élevée associées à un message de service du contrôleur, transférées par une connexion CIP Motion de manière périodique

NOTE Les données de service incluent les messages de demande de service qui permettent d'accéder aux attributs de l'Objet Axe de mouvement ou d'effectuer divers diagnostics de dispositif d'entraînement.

3.1.39

point de consigne

(dans le contexte d'un dispositif) valeur envoyée au dispositif d'entraînement et utilisée pour commander directement certains aspects liés à la dynamique du moteur, comprenant (sans toutefois s'y limiter) la position, la vitesse, l'accélération et le couple

3.1.40

définir/écrire

opération qui implique le réglage d'un attribut à une valeur spécifiée à partir de la partie contrôleur de l'interface

3.1.41

régulateur de shunt

méthode de régulation de bus spécifique qui fait basculer le bus à courant continu sur une résistance de dissipation d'énergie afin de dissiper la force régénératrice d'un moteur en cours de décélération

3.1.42

état

ensemble d'informations entre le PDS et le programme de contrôle d'application, qui reflète l'état ou le mode du PDS ou un élément fonctionnel de ce dernier

NOTE Les différentes informations d'état peuvent être codées avec un bit chacune.

3.1.43

synchronisé

état de verrouillage de la valeur d'horloge locale du dispositif d'entraînement sur l'horloge maîtresse du temps système réparti

NOTE Lorsqu'ils sont synchronisés, le dispositif d'entraînement et le contrôleur peuvent utiliser les datations associées aux données de connexion CIP Motion.

3.1.44

temps système (System Time)

valeur temporelle absolue définie dans la spécification de synchronisation CIP dans le contexte d'un système temporel réparti où tous les dispositifs comportent une horloge locale synchronisée avec une horloge maîtresse commune

NOTE Dans le contexte de CIP Motion, le temps système est une valeur entière à 64 bits en nanosecondes, avec une valeur de 0 correspondant à la date 1970-01-01.

3.1.45

modèle thermique

algorithme visant à modéliser le comportement thermique du dispositif concerné en cours de fonctionnement

3.1.46

décalage temporel

valeur de décalage du temps système associée aux données de connexion CIP Motion liées au dispositif source

NOTE La valeur de décalage du temps système est une valeur de 64 bits ajoutée à l'horloge locale d'un dispositif afin de générer son temps système.

3.1.47

datation (horodatage)

valeur de datation de temps système associée aux données de connexion CIP Motion qui achemine le temps absolu lorsque les données associées ont été saisies, ou qui peut également être utilisée pour déterminer le moment où les données associées doivent être appliquées

3.1.48

type

élément matériel ou logiciel qui spécifie les attributs communs partagés par toutes les instances du type

[CEI/TR 62390:2005, 3.1.25]

3.1.49

variable

entité logicielle qui peut prendre différentes valeurs, mais une seule valeur à la fois

[CEI/TR 62390:2005, 3.1.27]

NOTE Les valeurs d'une variable, ainsi que d'un paramètre, se limitent habituellement à un certain type de données.

3.1.50

dispositif d'entraînement à fréquence variable

VFD – Variable Frequency Drive

classe de produits d'entraînement visant à réguler le régime d'un moteur, généralement un moteur à induction, au moyen d'une relation proportionnelle entre la tension de sortie du dispositif d'entraînement et la fréquence de sortie commandée

NOTE Par conséquent, les dispositifs d'entraînement à fréquence variable sont parfois désignés dispositifs d'entraînement à Volts/Hertz.

3.1.51

dispositif d'entraînement vectoriel

classe de produits d'entraînement visant à commander les éléments dynamiques d'un moteur par l'intermédiaire d'une boucle fermée comprenant, sans toutefois s'y limiter, le contrôle de boucle fermée du couple et des composantes vectorielles de flux du courant du stator par rapport au vecteur de flux du rotor

3.2 Abréviations

Ack	Acquittement (Acknowledge)
Act	Actuel
Blk	Bloc
CIP™	Protocole industriel commun (Common Industrial Protocol) (voir CEI 61158 Type 2, CEI 61784-1 et CEI 61784-2 CPF2)
Cmd	Commande
Cyc	Cyclique
IHM	Interface homme-machine
ID	Identificateur
E/S	Entrée-sortie
Opt.	Facultatif (Optional)
PDS	Entraînement électrique de puissance (Power Drive System)
Req.	Obligatoire (Required)
RMS (eff)	Moyenne quadratique (Root Mean Square) (valeur efficace)
r/min	Tours par minute
Sync'd	Synchronisé
TCP	Protocole de contrôle de transmission (Transmission Control Protocol) (voir RFC 793)
UDP	User Datagram Protocol (voir RFC 768)
VFD	Dispositif d'entraînement à fréquence variable (Variable Frequency Drive)

4 Généralités

4.1 Généralités

Un dispositif d'entraînement CIP Motion permet de contrôler le mouvement d'un ou de plusieurs axes d'entraînement. En règle générale, il est composé d'un actionneur de moteur rotatif ou linéaire et d'un ou de plusieurs dispositifs de réaction facultatifs. Chaque moteur associé est entraîné par une structure de puissance d'entraînement associée faisant également partie intégrante du dispositif. La fonctionnalité de commande de moteur prise en charge par le profil de dispositif d'entraînement CIP Motion peut être appliquée aux dispositifs d'entraînement à fréquence variable en "boucle ouverte" ou aux entraînements asservis à commande vectorielle en "boucle fermée". Dans tous les cas, le mouvement de l'axe est commandé par une référence de consigne qui peut être configurée pour l'asservissement de position, la commande de vitesse, le contrôle d'accélération ou l'asservissement de courant/couple. Le profil de dispositif d'entraînement CIP Motion prend également en charge

l'utilisation de voies supplémentaires de réaction d'entraînement en tant que sources de réaction maître.

La spécification du profil d'entraînement CIP Motion et la spécification d'Objet d'axe de mouvement associée définissent l'interface, les attributs spécifiques et les comportements de consigne requis pour la prise en charge de la commande de mouvement lors de la connexion à un contrôleur à conformité CIP Motion par l'intermédiaire d'un réseau CIP.

Les réseaux CIP applicables sont DeviceNet™, ControlNet™ et EtherNet/IP™. EtherNet/IP est le réseau privilégié pour les commandes multi-axe synchronisées haute performance. ControlNet et DeviceNet sont uniquement destinés aux applications de dispositif non synchronisé à faible performance (les dispositifs d'entraînement à fréquence variable simples, les dispositifs d'entraînement à boucle de vitesse et les dispositifs d'entraînement indexés, par exemple).

4.2 Modes de commande

4.2.1 Généralités

L'Objet Axe de mouvement porte sur le comportement des différents dispositifs de système de commande de mouvement, y compris les dispositifs de réaction et les dispositifs d'entraînement. Concernant les dispositifs d'entraînement, l'Objet Axe de mouvement couvre un large éventail de types d'entraînement, des simples dispositifs d'entraînement à fréquence variable (V/Hz) aux entraînements d'asservissement de position les plus sophistiqués. En effet, la plupart des produits d'entraînement disponibles dans le commerce sont équipés d'axes qui peuvent être configurés pour fonctionner dans l'un quelconque de ces différents modes de commande de mouvement, en fonction des exigences spécifiques à l'application.

Les attributs de l'Objet Axe de mouvement sont donc organisés de manière à satisfaire cette large gamme de fonctionnalités, en utilisant le cadre des "modes d'application" définis dans la CEI 61800-7-1, présentés plus en détail dans les paragraphes ci-après.

La CEI 61800-7-1 et la présente partie de la série CEI 61800-7 s'appuient sur la philosophie générale selon laquelle l'asservissement de position est la forme la plus élevée de commande dynamique. En d'autres termes, l'asservissement de position implique la commande de vitesse, qui elle-même implique la commande d'accélération. L'accélération est liée au couple ou à la force par l'inertie ou la masse de la charge, respectivement, la commande d'accélération impliquant l'asservissement de couple. Enfin, le couple moteur ou la force du moteur est généralement lié au courant du moteur par une constante de couple ou de force, respectivement, l'asservissement de couple impliquant l'asservissement de courant. La constante de couple ou de force peut être fonction des aimants de moteur (dans les moteurs à aimant permanent, par exemple) ou du flux induit d'un moteur à induction.

L'accélération, le couple/la force et le courant étant généralement liés par une constante, ces termes sont parfois utilisés les uns pour les autres dans l'industrie. Par exemple, la CEI 61800-7-1 fait référence à une boucle d'asservissement de couple plutôt qu'à une boucle d'asservissement de courant. Cette spécification tente de distinguer ces propriétés de commande, le cas échéant. Cela est particulièrement utile si leur relation n'est pas statique (lorsque l'inertie/la masse change en fonction de la position ou de la durée ou lorsque la constante de couple/force change au grès des variations de la température ou du flux du moteur, par exemple).

4.2.2 Méthodes de commande

Ce paradigme de commande de base offre toute la latitude nécessaire aux différentes méthodes de commande, en boucle fermée et ouverte. Une boucle fermée implique généralement la présence d'un signal de réaction utilisé pour entraîner les éléments dynamiques réels du moteur afin de correspondre aux éléments dynamiques commandés par l'action asservie. Dans la plupart des cas, un dispositif de réaction littéral fournit ce signal, mais dans d'autres cas, le signal est déduit de l'excitation du moteur (c'est-à-dire d'un

fonctionnement sans capteur). Une boucle ouverte implique l'absence d'application de réaction pour forcer les éléments dynamiques réels à correspondre aux éléments dynamiques commandés. La précision et les performances étant les marques du contrôle de boucle fermée, la simplicité et l'économie sont celles du contrôle de boucle ouverte.

4.2.3 Nomenclature de commande

Enfin, comme le montre la description ci-dessus, les applications de commande linéaires et rotatives peuvent avoir un impact sur la nomenclature de commande. Les applications rotatives parlent de couple et d'inertie, alors que les applications linéaires parlent de force et de masse. Pour les besoins de la présente partie de la série CEI 61800-7, s'il est fait référence à la nomenclature rotative, le comportement défini peut en général être appliqué aux applications linéaires en remplaçant les termes utilisés ("force" pour "couple" et "masse" pour "inertie"). A cet égard, les expressions de la Figure 2 à la Figure 8 suivent la nomenclature de la CEI 61800-7-1 en utilisant le terme "couple" plutôt que le terme "force" sans réduire le caractère général.

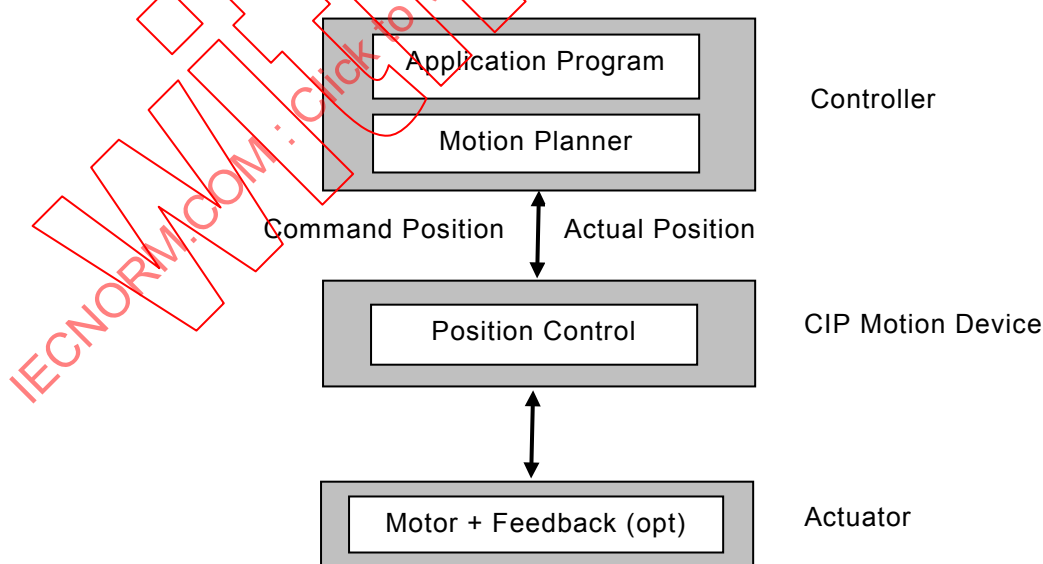
4.2.4 Asservissement de position

4.2.4.1 Concepts généraux

En mode d'application d'asservissement de position, le programme de contrôle d'application (fonction d'exécution de commande) ou le planificateur de mouvement (fonction de commande de la trajectoire de déplacement) fournit une valeur de point de consigne au dispositif CIP Motion par l'intermédiaire d'une connexion de données cycliques. La méthode d'asservissement de position peut être une boucle ouverte ou une boucle fermée.

4.2.4.2 Asservissement de position en boucle ouverte

Un dispositif configuré pour l'asservissement de position en boucle ouverte s'applique à une classe de dispositifs d'entraînement appelés "entraînements pas à pas". Ce type d'entraînement est représenté à la Figure 2.



Légende

Anglais	Français
Application program	Programme d'application
Controller	Contrôleur

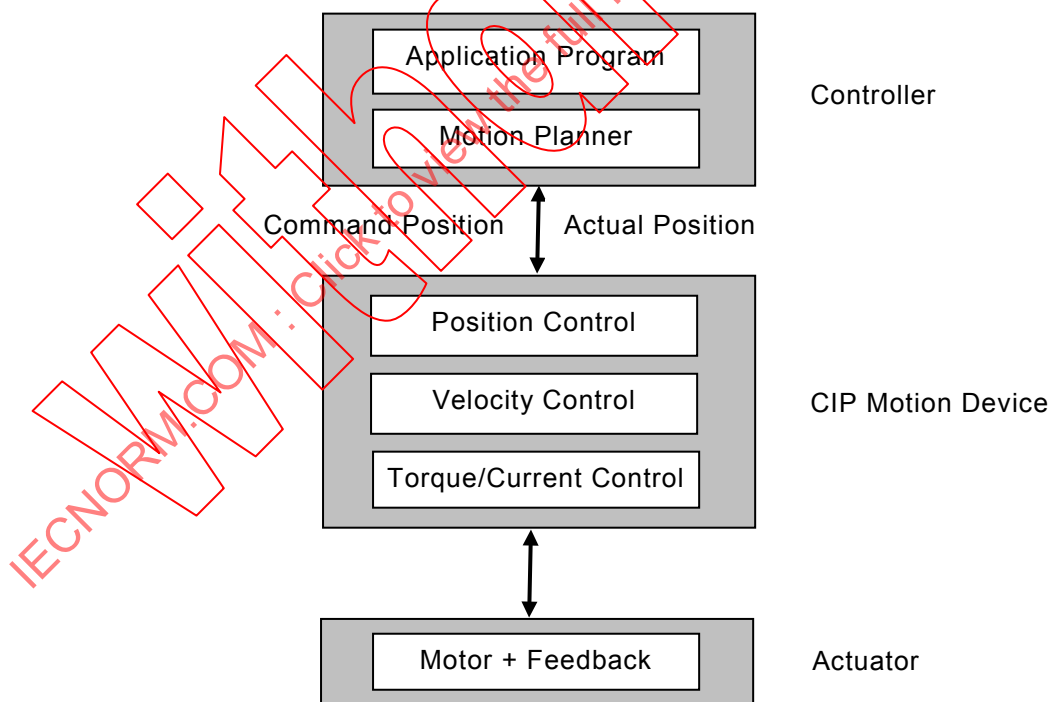
Anglais	Français
Motion planner	Planificateur de mouvement
Command position	Position de consigne
Actual position	Position réelle
Position control	Asservissement de position
CIP motion device	Dispositif CIP Motion
Motor + feedback (opt)	Moteur + réaction (facultative)
Actuator	Actionneur

Figure 2 – Asservissement de position en boucle ouverte

Dans cette configuration, la présence d'un dispositif de réaction est facultative. En l'absence de dispositif de réaction, la position réelle peut être estimée par le dispositif d'entraînement et renvoyée au contrôleur.

4.2.4.3 Asservissement de position en boucle fermée

Un dispositif de commande de mouvement configuré pour l'asservissement de position en boucle fermée est en général appelé dispositif d'entraînement de boucle de position ou entraînement asservi de position. Un entraînement asservi de position implique une vitesse interne et une boucle d'asservissement de couple (voir Figure 3). La présence d'une boucle d'asservissement de couple/courant donne parfois lieu à ce type de dispositif d'entraînement, appelé dispositif d'entraînement vectoriel.



Légende

Anglais	Français
Application Program	Programme d'application
Motion Planner	Planificateur de mouvement
Actual Position	Position réelle

Anglais	Français
Command Position	Position de consigne
Controller	Contrôleur
Actuator	Actionneur
CIP Motion Device	Dispositif CIP Motion
Motor + Feedback	Moteur + Réaction
Torque/Current Control	Asservissement de couple/courant
Position Control	Asservissement de position
Velocity Control	Commande de vitesse

Figure 3 – Asservissement de position en boucle fermée

Dans cette configuration, un dispositif de réaction est en général requis pour obtenir une bonne précision de positionnement. Le dispositif de réaction peut également être utilisé pour renvoyer les données de vitesse réelle et d'accélération réelle au contrôleur par l'intermédiaire de la connexion de données cycliques.

Outre la position de consigne, le contrôleur peut transmettre la vitesse de consigne et l'accélération de consigne pour les besoins de l'asservissement avancé.

4.2.5 Commande de vitesse

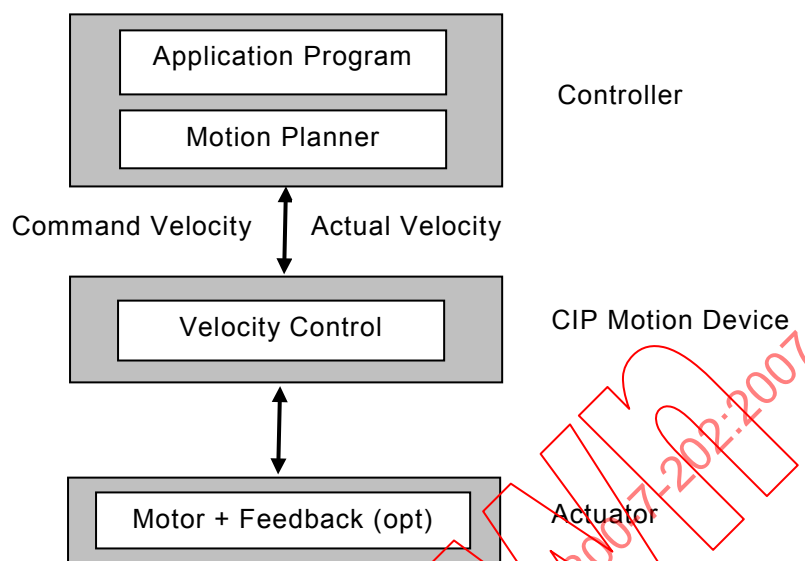
4.2.5.1 Concepts généraux

En mode d'application de commande de vitesse, le programme de contrôle d'application et le planificateur de mouvement fournissent une valeur de point de consigne au dispositif CIP Motion par l'intermédiaire de la connexion de données cycliques. La méthode de commande de vitesse peut être une boucle ouverte ou une boucle fermée.

4.2.5.2 Commande de vitesse en boucle ouverte

Un dispositif de commande de mouvement configuré pour la commande de vitesse en boucle ouverte est en général appelé dispositif d'entraînement à fréquence variable, V/Hz ou VFD. Ce type de dispositif d'entraînement est représenté à la Figure 4.

Dans cette configuration, la présence d'un dispositif de réaction est facultative. En l'absence de dispositif de réaction, la vitesse réelle peut être estimée par le dispositif d'entraînement et renvoyée au contrôleur.



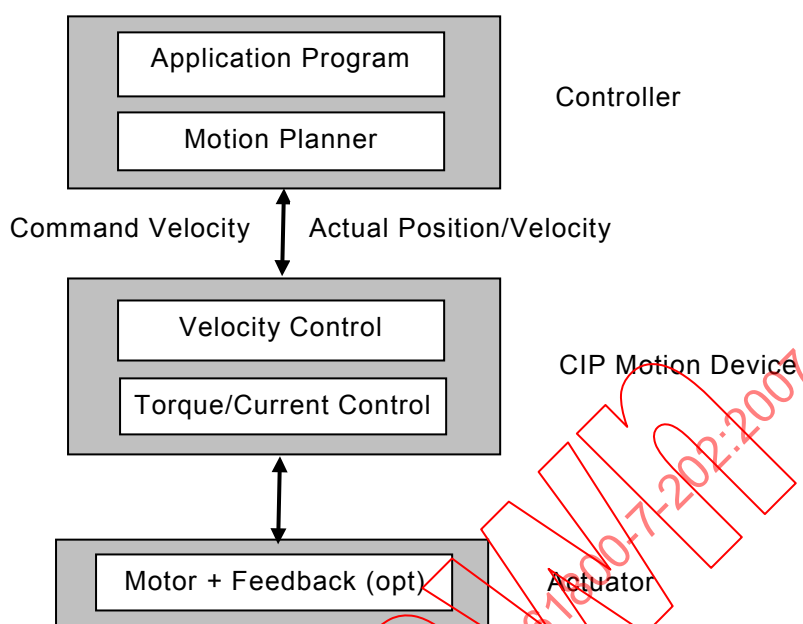
Légende

Anglais	Français
Application Program	Programme d'application
Motion Planner	Planificateur de mouvement
Velocity Control	Commande de vitesse
Command Velocity	Consigne de vitesse
Controller	Contrôleur
Actuator	Actionneur
CIP Motion Device	Dispositif CIP Motion
Motor + Feedback (opt)	Moteur + Réaction (facultatif)
Actual velocity	Vitesse réelle

Figure 4 – Commande de vitesse en boucle ouverte

4.2.5.3 Commande de vitesse en boucle fermée

Un dispositif de commande de mouvement configuré pour la commande de vitesse en boucle fermée est en général appelé dispositif d'entraînement de boucle de vitesse ou entraînement asservi de vitesse. Un entraînement de commande de vitesse en boucle fermée implique une boucle d'asservissement de couple/courant interne (voir Figure 5). Il est donc parfois appelé dispositif d'entraînement vectoriel.



Légende

Anglais	Français
Application Program	Programme d'application
Motion Planner	Planificateur de mouvement
Velocity Control	Commande de vitesse
Command Velocity	Consigne de vitesse
Controller	Contrôleur
Actuator	Actionneur
CIP Motion Device	Dispositif CIP Motion
Motor + Feedback (opt)	Moteur + Réaction (facultatif)
Actual position/velocity	Position/Vitesse réelles
Torque/current control	Asservissement de couple/courant

Figure 5 – Commande de vitesse en boucle fermée

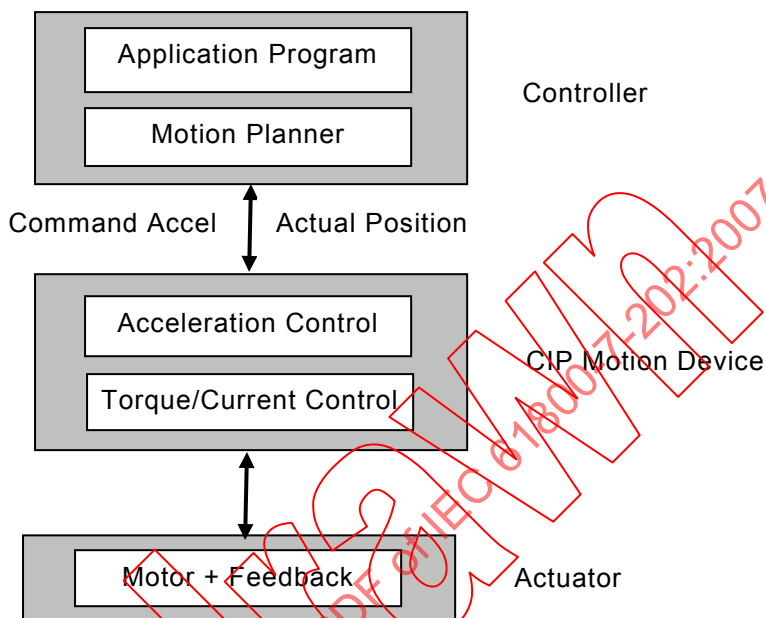
Dans cette configuration, la présence d'un dispositif de réaction pour entraînement de commande de vitesse est facultative. Une régulation de vitesse plus stricte est obtenue à l'aide d'un dispositif de réaction, particulièrement à faible vitesse. Si le dispositif de réaction est inclus, il peut être utilisé pour renvoyer les données de position, de vitesse et d'accélération réelles au contrôleur par l'intermédiaire de la connexion de données cycliques. Si le dispositif de réaction n'est pas inclus, seule la vitesse estimée peut en général être renvoyée au contrôleur.

Outre la vitesse de consigne, le contrôleur peut également transmettre l'accélération de consigne pour les besoins de l'asservissement avancé.

4.2.6 Commande d'accélération

Le mode de commande principal n'étant mentionné ni dans l'industrie ni dans la CEI 61800-7-1, le mode de commande d'accélération est présenté ici pour compléter la progression dynamique entre la commande de vitesse et l'asservissement de couple, et parce que l'Objet Axe de mouvement peut prendre en charge une consigne d'accélération, potentiellement déduite du planificateur de mouvement du contrôleur. En mode de commande d'accélération, le programme de contrôle d'application et le planificateur de mouvement

fournissent des valeurs de point de consigne d'accélération au dispositif CIP Motion par l'intermédiaire de la connexion de données cycliques. Le dispositif d'entraînement convertit le point de consigne d'accélération en consigne de couple à l'aide de l'inertie estimée du système. La commande d'accélération fonctionne de concert avec la boucle d'asservissement de couple/courant interne (voir Figure 6).



Légende

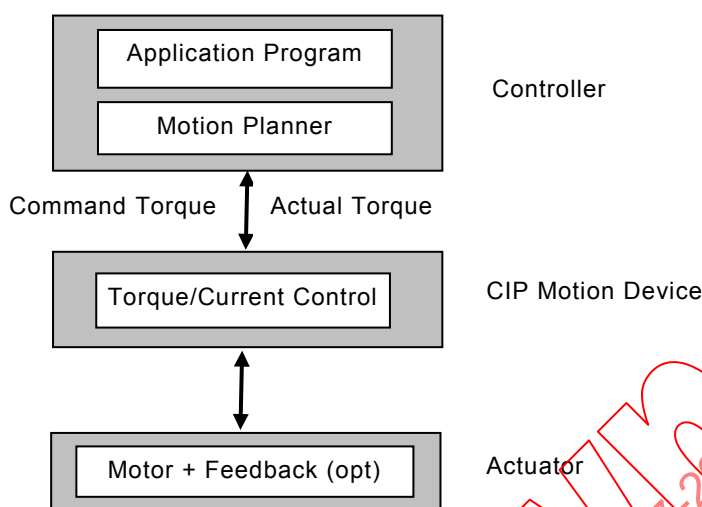
Anglais	Français
Application Program	Programme d'application
Motion Planner	Planificateur de mouvement
Actual Position	Position réelle
Torque/Current Control	Asservissement de couple/courant
Controller	Contrôleur
Actuator	Actionneur
CIP Motion Device	Dispositif CIP Motion
Motor + Feedback	Moteur + Réaction
Command Accel	Accélération de consigne
Acceleration Control	Commande d'accélération

Figure 6 – Commande d'accélération

La présence d'un dispositif de réaction de commande d'accélération est obligatoire. Il peut être utilisé pour renvoyer les données de position, de vitesse et d'accélération réelles au contrôleur par l'intermédiaire de la connexion de données cycliques.

4.2.7 Asservissement de couple

En mode d'application d'asservissement de couple, le programme de contrôle d'application et le planificateur de mouvement fournissent des valeurs de point de consigne de couple au dispositif par l'intermédiaire de la connexion de données cycliques (voir Figure 7). Le courant et le couple du moteur étant généralement liés par une constante de couple (K_t), l'asservissement de couple est souvent synonyme d'asservissement de courant.



Légende

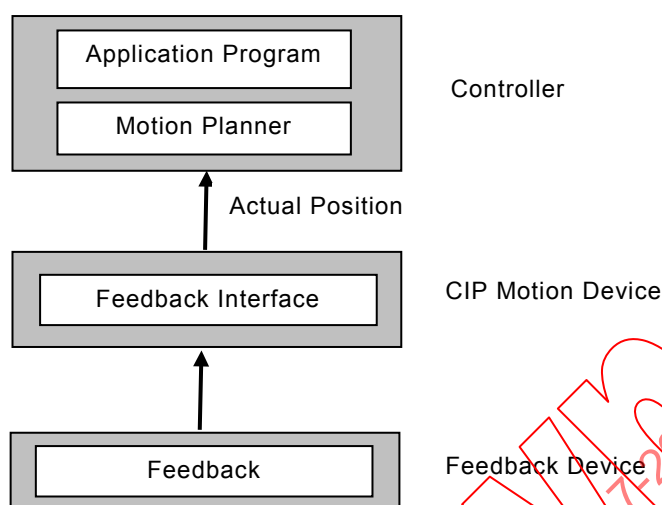
Anglais	Français
Application Program	Programme d'application
Motion Planner	Planificateur de mouvement
Actual Torque	Couple réel
Torque/Current Control	Asservissement de couple/courant
Controller	Contrôleur
Actuator	Actionneur
CIP Motion Device	Dispositif CIP Motion
Motor + Feedback (opt)	Moteur + Réaction (facultatif)
Command Torque	Couple de consigne

Figure 7 – Asservissement de couple

La présence d'un dispositif de retour en position est facultative pour ce mode d'asservissement. Si un dispositif de réaction est présent, il peut être utilisé pour renvoyer les données de position, de vitesse et d'accélération réelles au contrôleur par l'intermédiaire de la connexion de données cycliques.

4.2.8 Pas d'asservissement

L'Objet Axe de mouvement prend en charge un mode d'application "Pas d'asservissement". Cette configuration, également appelée "réaction uniquement" ou "réaction maître", peut être utile lorsqu'une voie particulière de réaction dans le dispositif d'entraînement CIP Motion consiste à faire simplement office de source de réaction maître vers le reste du système de commande. Ce mode d'application peut également être appliqué aux dispositifs de réaction CIP Motion. Dans cette configuration "Pas d'asservissement", aucune valeur de point de consigne n'est fournie au dispositif CIP Motion par l'intermédiaire de la connexion de données cycliques, mais les données de position, de vitesse et d'accélération réelles peuvent être fournies par le dispositif au contrôleur via le canal de données cycliques. La configuration d'entraînement en mode "Pas d'asservissement" est illustrée à la Figure 8.

**Légende**

Anglais	Français
Application Program	Programme d'application
Motion Planner	Planificateur de mouvement
Feedback Interface	Interface de réaction
Feedback	Réaction
Controller	Contrôleur
Actual Position	Position réelle
CIP Motion Device	Dispositif CIP Motion
Feedback Device	Dispositif de réaction

Figure 8 – Pas d'asservissement**5 Types de données****5.1 Type de données – Présentation**

Le Tableau 1 présente les références de types de données utilisées dans ce profil et leurs définitions connexes.

Tableau 1 – Types de données

Types de données utilisés dans CIP Motion	Référence à la définition
BOOL	Booléen (voir 5.3.1.1.2 de la CEI 61158-5-2)
SINT	Integer8 (voir 5.3.1.4.2.2 de la CEI 61158-5-2)
INT	Integer16 (voir 5.3.1.4.2.4 de la CEI 61158-5-2)
DINT	Integer32 (voir 5.3.1.4.2.6 de la CEI 61158-5-2)
LINT	Integer64 (voir 5.3.1.4.2.8 de la CEI 61158-5-2)
USINT	Unsigned8 (voir 5.3.1.4.3.2 de la CEI 61158-5-2)
UINT	Unsigned16 (voir 5.3.1.4.3.4 de la CEI 61158-5-2)
UDINT	Unsigned32 (voir 5.3.1.4.3.6 de la CEI 61158-5-2)

Types de données utilisés dans CIP Motion	Référence à la définition
ULINT	Unsigned64 (voir 5.3.1.4.3.8 de la CEI 61158-5-2)
REAL	Float32 (voir 5.3.1.4.1.2 de la CEI 61158-5-2)
LREAL	Float64 (voir 5.3.1.4.1.4 de la CEI 61158-5-2)
SWORD/BYTE	Bitstring8 (voir 5.3.1.2.2 de la CEI 61158-5-2)
WORD	Bitstring16 (voir 5.3.1.2.4 de la CEI 61158-5-2)
DWORD	Bitstring32 (voir 5.3.1.2.6 de la CEI 61158-5-2)
LWORD	Bitstring64 (voir 5.3.1.2.8 de la CEI 61158-5-2)

5.2 Conventions

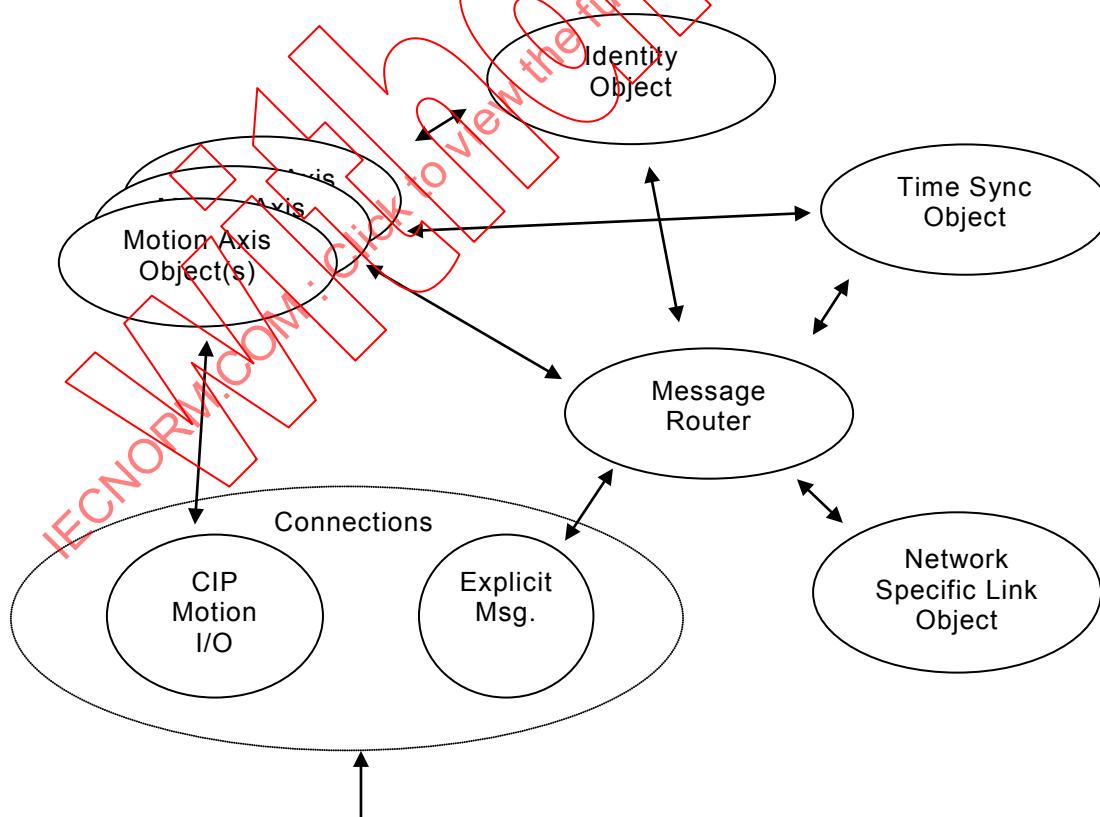
Les nombres hexadécimaux de la présente partie de la série CEI 61800-7 peuvent être représentés par nn_{hex} (si leur type de données exact n'est pas défini) ou $0xnn$, $0xnxxx$ (si leur type de données est spécifié).

6 Profil de dispositif d'entraînement CIP

6.1 Modèle d'objet

6.1.1 Présentation de l'objet

Le modèle d'objet de la Figure 9 représente un dispositif d'entraînement CIP Motion.



Légende

Anglais	Français
Identity Object	Objet Identité
Time Sync Object	Objet de synchronisation temporelle
Motion axis	Axe de mouvement
Network Specific Link	Liaison spécifique au réseau
Explicit Msg.	Message explicite
Connection	Connexion
CIP Motion I/O	E-S CIP Motion
Message Router	Routeur de message

Figure 9 – Modèle d'objet d'un dispositif d'entraînement CIP Motion

Le Tableau 2 indique:

- les classes d'objet présentes dans ce dispositif,
- le caractère obligatoire ou non de la classe,
- le comptage d'instances présentes dans chaque classe.

Tableau 2 – Objets présents dans un dispositif d'entraînement CIP Motion

Classe d'objet	Facultatif ou obligatoire	Comptage d'instances
Objets requis communs CIP	Obligatoire	Voir CEI 61158-5-2 et CEI 61158-6-2
Objet Axe de mouvement	Obligatoire	1 par axe
Objet Synchronisation temporelle	Facultatif	1

Voir la CEI 61158-5-2, la CEI 61158-6-2 et 6.4.6.1 pour plus de détails sur ces objets.

6.1.2 Description de l'objet

Le modèle d'objet de la Figure 9 illustre les principaux composants fonctionnels du profil de dispositif d'entraînement CIP Motion défini dans le cadre des extensions CIP Motion vers CIP.

Ce modèle d'objet illustre également l'utilisation de plusieurs instances d'un Objet Axe de mouvement pour mettre en œuvre un dispositif d'entraînement multi-axe. Chaque instance de l'Objet Axe de mouvement gère le comportement de l'axe d'entraînement associé. Dans ce profil de dispositif, le terme "axe d'entraînement" est synonyme d'une "instance d'Objet Axe de mouvement".

Une seule connexion E-S bidirectionnelle à l'instance de la classe Objet Axe de mouvement offre un chemin d'accès aux données cycliques entre le contrôleur et chaque instance d'Objet Axe de mouvement. Cette connexion transmet une structure de données particulière dont le format à définition automatique peut être utilisé pour transférer des données cycliques, d'événement et de service.

Un Objet Synchronisation temporelle facultatif est inclus dans le modèle d'objet. Il s'agit de faciliter la synchronisation temporelle précise entre les contrôleurs CIP Motion et les dispositifs d'entraînement pour une commande de mouvement haute performance. Pour les dispositifs d'entraînement présentant des performances moins élevées (les dispositifs d'entraînement V/Hz ou VFD, par exemple), l'Objet Synchronisation temporelle n'est pas nécessaire pour le fonctionnement du dispositif d'entraînement.

6.2 Influence des objets sur le comportement

Les objets de ce dispositif ont un impact sur son comportement (voir Tableau 3).

Tableau 3 – Influence de l'objet sur le comportement

Classe d'objet	Effet sur le comportement
Objets requis communs CIP	Voir CEI 61158-5-2 et CEI 61158-6-2
Objet Axe de mouvement	Offre une interface de commande dynamique aux composants d'entraînement, de moteur et de réaction comprenant un axe.
Objet Synchronisation temporelle	Offre des services de synchronisation temporelle absolue au réseau de commande.

6.3 Définition des interfaces d'objet

Les objets pris en charge pour le dispositif d'entraînement CIP Motion comportent les interfaces figurant au Tableau 4.

Tableau 4 – Interfaces d'objet

Objet	Interface
Objets requis communs CIP	Voir CEI 61158-5-2 et CEI 61158-6-2
Objet Axe de mouvement	Classe Routeur de message ou Objet Axe de mouvement
Objet Synchronisation temporelle	Classe Routeur de message ou Objet Axe de mouvement

6.4 Messages de connexion E-S

6.4.1 Généralités

Le profil de dispositif d'entraînement CIP Motion prend en charge une connexion E-S bidirectionnelle point à point entre le contrôleur et la classe Objet Axe de mouvement: cette connexion E-S est spécifiquement appelée connexion E-S CIP Motion.

L'Objet Axe de mouvement distribue les données de cette connexion à chaque instance Objet Axe de mouvement instanciée. Par conséquent, les chemins de connexion générés et utilisés ciblent explicitement les attributs de la classe Objet Axe de mouvement (c'est-à-dire l'instance 0). Le chemin Connexion utilisée correspond à la connexion Contrôleur - Dispositif (C-D), le chemin Connexion générée correspondant à la connexion Dispositif - Contrôleur (D-C), comme le détaille les paragraphes ci-dessous.

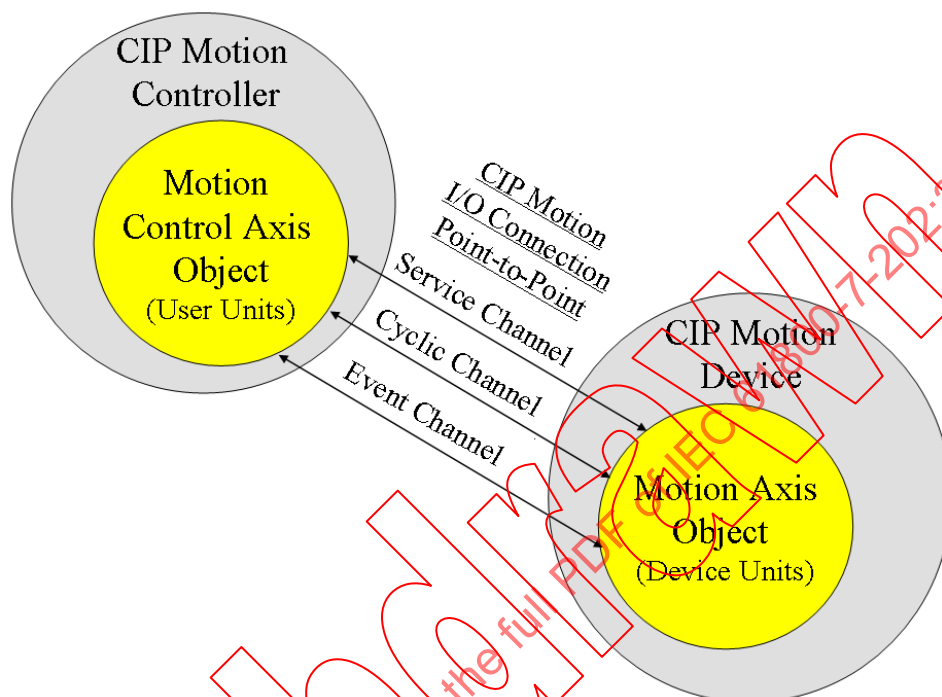
Les chemins Connexion E-S de ce dispositif sont les suivants:

Chemin Connexion utilisée	=	Classe Objet Axe de mouvement/Instance 0 / (attribut Données de connexion utilisée du contrôleur)
Chemin Connexion générée	=	Classe Objet Axe de mouvement/Instance 0 / (attribut Données de connexion générée du contrôleur)

6.4.2 Connexion E/S CIP Motion

6.4.2.1 Présentation générale

Le paragraphe ci-dessous spécifie le format de connexion E-S CIP Motion comprenant la connexion C-D et la connexion D-C dans le cadre d'un transfert bidirectionnel de données entre un contrôleur CIP Motion et un dispositif d'entraînement CIP Motion (voir Figure 10).



Légende

Anglais	Français
CIP Motion Controller	Contrôleur CIP Motion
Motion Control Axis Object (User units)	Objet Axe de commande de mouvement (Unités utilisateur)
CIP Motion Device	Dispositif CIP Motion
Motion Axis Object (Device Units)	Objet Axe de mouvement (Unités du dispositif)
CIP Motion I/O Connection	Connexion E-S CIP Motion
Point-to-Point	Point à point
Service Channel	Voie de service
Cyclic Channel	Voie cyclique
Event Channel	Voie d'événements

Figure 10 – Modèle de connexion E-S CIP Motion

6.4.2.2 Formats de connexion E-S

Les formats de connexion Contrôleur - Dispositif et Dispositif - Contrôleur sont respectivement présentés à la Figure 11 et à la Figure 12.

← Mot de 32 bits →			
Format de connexion Contrôleur - Dispositif			
En-tête de connexion			
Blocs de données d'instance			
En-tête de connexion			
Format de connexion	Révision de format	ID d'actualisation	Contrôle de nœud
Comptage d'instances	-	-	Ensemble de données temporelles
Période d'actualisation du contrôleur			
Décalage temporel du contrôleur			
Datation du contrôleur			
Bloc de données d'instance			
En-tête de données d'instance			
Bloc de données cycliques			
Bloc de données d'écriture cycliques			
Bloc de données d'événement			
Bloc de données de service			
En-tête de données d'instance			
Numéro d'instance	-	Taille de bloc d'instance	Taille de bloc cyclique
Taille de bloc de commande cyclique	Taille de bloc d'écriture cyclique	Taille de bloc d'événement	Taille de bloc de service
Bloc de données cycliques			
Mode de commande	Configuration de réaction	Commande d'axe	-
Ensemble de données de consigne	Ensemble de données réelles	Ensemble de données d'état	Commande d'interpolation
Données cycliques			
Bloc de données d'écriture cycliques			
ID de bloc d'écriture cyclique		ID de bloc de lecture cyclique	
Données d'écriture cycliques			
Bloc de données d'événement			
Commande de vérification d'événement			
Acquittement d'événement ID1	Acquittement d'événement État 1	Acquittement d'événement ID2	Acquittement d'événement État 2
...			
Bloc de données de service			
ID de transaction	Code de service	-	-
Données de demande spécifique au service			

Figure 11 – Format de connexion Contrôleur - Dispositif

← Mot de 32 bits →			
Format de connexion Dispositif - Contrôleur			
En-tête de connexion			
Blocs de données d'instance			
En-tête de connexion			
Format de connexion	Révision de format	ID d'actualisation	Statut de nœud
Comptage d'instances	Alarmes de nœud	Défauts de nœud	Ensemble de données temporelles
Période d'actualisation du dispositif			
Décalage temporel du dispositif			
Datation du dispositif			
Données de diagnostic de synchronisation			
Bloc de données d'instance			
En-tête de données d'instance			
Bloc de données cycliques			
Bloc de données de lecture cycliques			
Bloc de données d'événement			
Bloc de données de service			
En-tête de données d'instance			
Numéro d'instance	-	Taille de bloc d'instance	Taille de bloc cyclique
Taille du bloc réel cyclique	Taille de bloc de lecture cyclique	Taille de bloc d'événement	Taille de bloc de service
Bloc de données cycliques			
Mode de commande	Configuration de réaction	Réponse d'axe	Statut de réponse
-	Ensemble de données réelles	Ensemble de données d'état	État de l'axe
Données cycliques			
Bloc de données de lecture cycliques			
ID de bloc de lecture cyclique	-		
Données de lecture cycliques			
Bloc de données d'événement			
Statut de vérification d'événement			
ID d'événement	Statut d'événement	Type d'événement	-
Position d'événement			
Datation de l'événement			
...			
Bloc de données de service			
ID de transaction	Code de service	Statut général	Statut étendu
Données de réponse spécifique au service			

Figure 12 – Format de connexion Dispositif - Contrôleur

6.4.2.3 Structure de connexion CIP Motion

La connexion E-S CIP Motion est essentiellement composée de deux connexions à destination unique unidirectionnelles, l'une transmettant les données du contrôleur au dispositif d'entraînement, et l'autre dans le sens inverse.

Les structures de données de connexion CIP Motion (Contrôleur - Dispositif et Dispositif - Contrôleur) commencent par un En-tête de connexion contenant une datation 32 bits, suivie d'une série de blocs de données pour chaque instance d'axe prise en charge par le nœud d'entraînement.

Chaque bloc de données d'instance du paquet de connexions E-S CIP Motion est composé de trois ensembles de blocs de données associés aux canaux de données cycliques, d'événement et de service. La taille de ces blocs de données pour une actualisation donnée est variable. Elle est déterminée par les en-têtes de bloc de connexion et de données d'instance. Du point de vue du dispositif d'entraînement, ces trois canaux de données distincts font l'objet de différentes priorités de traitement de données (voir Figure 13).

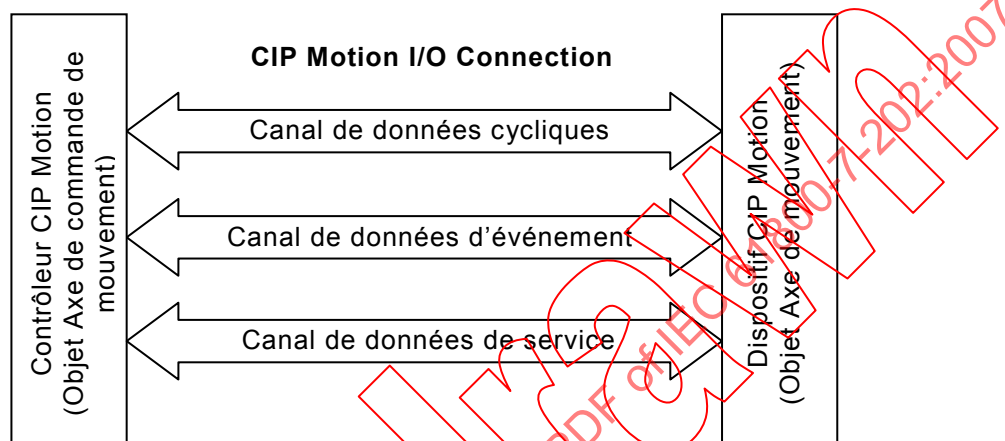


Figure 13 – Canaux de connexion E-S CIP Motion

Les fonctions spécifiques de ces trois canaux de données est la suivante:

- Canal de données cycliques – transporte des blocs de données cycliques échantillonnés ou calculés à chaque Période d'actualisation du contrôleur, et synchronisés avec d'autres nœuds dans le système de commande de mouvement grâce au Temps système réparti. Les données cycliques sont des données à priorité élevée qui doivent être immédiatement traitées et appliquées à l'axe d'entraînement dans une Période d'actualisation du dispositif.
- Canal de données d'événement – transporte les données d'événement associées aux événements d'entraînement (enregistrement, retour à la position de référence, etc.) qui se sont produits dans la dernière Période d'actualisation du contrôleur. Les données d'événement sont des données de priorité moyenne et doivent être traitées et appliquées dans la Période d'actualisation du contrôleur.
- Canal de données de service – transporte les données associées aux demandes de service de lecture et d'écriture des valeurs d'attribut de l'Objet Axe de mouvement dans le cadre d'une fonctionnalité de configuration et de diagnostic en ligne, ainsi que les demandes de service affectant le comportement de l'Objet Axe de mouvement dans le cadre de l'exécution d'une instruction du contrôleur. Les données de service sont des données de priorité basse et sont en général mises en mémoire tampon et traitées en arrière-plan. Rien ne peut garantir qu'une demande de service sera traitée dans la Période d'actualisation du contrôleur.

Considérés ensemble, ces trois canaux de données fournissent un contrôleur exhaustif à la solution de connexion des données d'entraînement pour la commande de mouvement industrielle.

6.4.3 Connexion Contrôleur - Dispositif

6.4.3.1 Généralités

Pour faciliter la description détaillée de chacun de ses éléments de données constituant, la connexion Contrôleur - Dispositif CIP Motion est organisée comme indiqué à la Figure 14.

Format de connexion Contrôleur - Dispositif CIP Motion
En-tête de connexion
Blocs de données d'instance

Figure 14 – Format de connexion Contrôleur - Dispositif CIP Motion

6.4.3.2 En-tête de connexion Contrôleur - Dispositif

6.4.3.2.1 Structure générale

L'en-tête de connexion Contrôleur - Dispositif contient les informations essentielles de configuration de l'axe nécessaires à l'analyse des blocs de données d'instance. La partie fixe de l'en-tête de connexion est définie comme illustré à la Figure 15.

En-tête de connexion			
Format de connexion	Révision de format	ID d'actualisation	Contrôle de nœud

Figure 15 – En-tête de connexion

- Format de connexion: cet octet énuméré détermine le format de la connexion CIP Motion conformément à la définition de la Figure 16.

Bit 7		Bit 0
Format de connexion		
(Réservé)		Type de connexion

Figure 16 – Format de connexion

- Type de connexion: cette énumération à 4 bits définit le type de connexion CIP Motion comme indiqué ci-dessous. Les valeurs valides d'une connexion Contrôleur - Dispositif sont 2 et 6. Les connexions fixes présentent une taille de connexion fixe pendant le fonctionnement, afin de prendre en charge les réseaux CIP basse performance (ControlNet et DeviceNet, par exemple) et généralement associés à un seul dispositif d'entraînement d'axe simple ne prenant pas en charge les services de synchronisation. Les connexions variables permettent à la taille de la structure de données de connexion de varier pendant le fonctionnement et sont destinées aux dispositifs d'entraînement haute performance et aux réseaux CIP tels qu'EtherNet/IP.

0 = Réservé

1 = Réservé

2 = Connexion Contrôleur - Dispositif fixe

3 = Connexion Dispositif - Contrôleur fixe

4 = Réservé

5 = Réservé

6 = Connexion Contrôleur - Dispositif variable

7 = Connexion Dispositif - Contrôleur variable

8 à 15 = Réservés.

- Révision de format: La présente édition de la CEI 61800-7-202 définit la révision de format 1, que seuls les dispositifs utilisant la présente édition de la CEI 61800-7-202 reconnaissent. Cette valeur est incrémentée de 1 à chaque révision du format de connexion Contrôleur - Dispositif ayant un impact sur l'interface. La révision de format permet aux dispositifs d'entraînement les plus récents de prendre en charge les formats de connexion générés par un contrôleur plus ancien. Elle permet également aux dispositifs d'entraînement plus anciens de reconnaître un format de connexion récent à

partir d'un contrôleur qui ne peut pas prendre en charge et générer un Code d'erreur de défaut de nœud 4 (Erreur de format de données).

- ID d'actualisation: Ce numéro de transaction cyclique est incrémenté à chaque période d'actualisation. L'ID d'actualisation s'apparente au nombre de séquences CIP et permet au dispositif d'entraînement de déterminer si la mémoire tampon de connexion contient de nouvelles données. Si l'ID d'actualisation n'a pas été modifié, le dispositif d'entraînement tente de passer sur l'actualisation manquée par extrapolation en fonction de la précédente trajectoire, tant que les nouvelles données ne sont pas arrivées. Si l'axe d'entraînement n'est pas synchronisé ou ne prend pas en charge les services de synchronisation temporelle, les données de datation ne sont pas incluses ou ne sont pas valides. L'ID d'actualisation est donc le seul moyen dont dispose le dispositif d'entraînement pour détecter les nouvelles données de connexion. Dans ce cas, l'ID d'actualisation de connexion Contrôleur - Dispositif est appliqué à l'ID d'actualisation de connexion Dispositif - Contrôleur suivant.
- Contrôle de nœud: Cet élément est appliqué à l'attribut de classe d'Objet Axe de mouvement "Contrôle de nœud", utilisé pour contrôler l'état du nœud de communication de dispositif d'entraînement associé. Voir 7.2.2.3 (Objet Axe de mouvement) pour plus de détails.

6.4.3.2.2 En-tête de connexion fixe

Si le format de connexion est une connexion Contrôleur - Dispositif fixe, l'en-tête ci-dessus est immédiatement suivi du bloc de données d'instance.

6.4.3.2.3 En-tête de connexion variable

Si le format de connexion est une connexion Contrôleur - Dispositif variable, l'en-tête de connexion contient des champs supplémentaires relatifs à l'adressage et à la datation d'entraînement multi-axe (voir Figure 17).

En-tête de connexion			
<i>Format de connexion</i>	<i>Révision de format</i>	<i>ID d'actualisation</i>	<i>Contrôle de nœud</i>
Comptage d'instances		-	Ensemble de données temporelles
Période d'actualisation du contrôleur			
Décalage temporel du contrôleur			
Datation du contrôleur			

Figure 17 – En-tête de connexion

- Comptage d'instances: cette valeur reflète le nombre de blocs de données d'instance présents dans la structure de données de connexion Contrôleur - Dispositif CIP Motion.
- Ensemble de données temporelles: cet octet de matriciel contient des indicateurs qui déterminent l'utilisation et le format des informations de synchronisation du contrôleur et du dispositif d'entraînement (voir Tableau 5). La valeur Ensemble de données temporelles envoyée par le contrôleur dans le cadre d'une connexion Contrôleur - Dispositif devient la valeur Ensemble de données temporelles envoyée par le dispositif d'entraînement dans le cadre d'une connexion Dispositif - Contrôleur. La valeur de cet élément est transférée vers l'attribut d'Objet Axe de mouvement "Ensemble de données de consigne".

Tableau 5 – Ensemble de données temporelles

Bit	Définition	Syntaxe
0	Période d'actualisation	0 = pas de période d'actualisation 1 = période d'actualisation incluse
1	Datation	0 = pas de datation et de décalage 1 = datation et décalage inclus
2	Données de diagnostic de synchronisation	0 = pas de données de diagnostic 1 = données de diagnostic incluses
3 à 7	(Réservé)	

- Période d'actualisation du contrôleur: Cet élément de 32 bits spécifie la période d'actualisation de la connexion Contrôleur - Dispositif en nanosecondes. La valeur de cet élément est transférée vers l'attribut de classe d'Objet Axe de mouvement du même nom. Lors de la synchronisation de la connexion, la période d'actualisation du contrôleur peut être utilisée par le dispositif d'entraînement avec la datation du contrôleur pour vérifier les actualisations manquées ou retardées. Si la différence entre la dernière datation du contrôleur et la datation en cours du dispositif d'entraînement local dépasse la valeur maximale donnée par,

Délai max = Limite supérieure du délai d'actualisation du contrôleur × Période d'actualisation du contrôleur,

le bit Défaut d'actualisation du contrôleur est défini dans l'attribut de classe Défauts de nœud de l'Objet Axe de mouvement. Un attribut facultatif Limite inférieure du délai d'actualisation du contrôleur est également disponible. Si la différence entre la dernière datation du contrôleur et la datation en cours du dispositif d'entraînement local dépasse la valeur maximale donnée par,

Délai max = Limite inférieure du délai d'actualisation du contrôleur × Période d'actualisation du contrôleur,

le bit Alarme d'actualisation du contrôleur est défini dans l'attribut de classe Alarme de nœud de l'Objet Axe de mouvement.

Lors de la synchronisation de la connexion et de l'activation de l'interpolation fine, la période d'actualisation du contrôleur est également utilisée par le dispositif d'entraînement avec la valeur Datation du contrôleur pour calculer les nouveaux coefficients de polynôme d'interpolation/extrapolation.

Si la connexion n'est pas synchronisée, le dispositif d'entraînement ne procède à aucune vérification de retard d'actualisation. Dans ce cas, il convient d'appliquer immédiatement les données de consigne, ce qui signifie de les appliquer immédiatement en tant que référence de consigne.

- Décalage temporel du contrôleur: Cet élément représente la valeur de décalage du temps système de 64 bits associée à la datation du contrôleur qui suit. La valeur de cet élément est transférée vers l'attribut de classe d'Objet Axe de mouvement du même nom. La valeur Décalage temporel du contrôleur permet au dispositif d'entraînement de déterminer si le temps système défini dans le contrôleur est étalé par rapport à celui du dispositif d'entraînement. En principe, le temps système entre le dispositif d'entraînement et le contrôleur est proche, quel que soit le temps considéré. Mais, conformément au protocole de la CEI 61588:2004, l'horloge maîtresse du système peut corriger toutes les secondes la référence du temps système d'une durée significative (une heure, par exemple). En effet, même le maître lui-même peut changer si une horloge maîtresse de qualité supérieure est détectée par le système. Ces modifications brusques apportées à la référence du temps système se propagent vers les différents dispositifs présents sur le réseau, de sorte que le contrôleur et le dispositif d'entraînement fonctionnent avec des valeurs de temps système étalées. La valeur Décalage temporel du contrôleur peut être utilisée conjointement avec la valeur Décalage du temps système du dispositif d'entraînement, afin de déterminer si le temps système entre le dispositif d'entraînement et le contrôleur est étalé et, si c'est le cas, pour compenser cet étalement.

- **Datation du contrôleur:** Cet élément représente la valeur Temps système de 64 bits au début de la période d'actualisation du contrôleur lorsqu'un événement de temporisation de l'actualisation du contrôleur se produit. Calculé par le contrôleur, il s'agit de la somme de la valeur d'horloge locale du contrôleur en cas d'événement de temporisation d'actualisation et de la valeur Décalage du temps système du contrôleur. La datation du contrôleur est donc directement liée aux données de consigne contenues dans la connexion. La valeur de cet élément est transférée vers l'attribut de classe d'Objet Axe de mouvement du même nom. Outre la période d'actualisation du contrôleur et le temps de consigne cible (présenté ci-après), le dispositif d'entraînement dispose de toutes les informations dont il a besoin pour calculer les polynômes d'interpolation/extrapolation de consigne, afin de corriger les différences de valeurs des données de consigne entre la temporisation d'actualisation du dispositif d'entraînement et du contrôleur. Ces différences peuvent apparaître lorsque la période d'actualisation du contrôleur n'est pas un entier multiple de la période d'actualisation du dispositif ou que les actualisations du dispositif d'entraînement présentent un déphasage par rapport au contrôleur.

6.4.3.3 Blocs de données d'instance

6.4.3.3.1 Structure générale

L'en-tête de connexion est suivi d'un ou de plusieurs Blocs de données d'instance, comme le montre le Comptage d'instances ci-dessus. La structure de base du Bloc de données d'instance est représentée à la Figure 18.

Bloc de données d'instance
En-tête de données d'instance
Bloc de données cycliques
Bloc de données d'événement
Bloc de données de service

Figure 18 – Bloc de données d'instance

6.4.3.3.2 En-tête de données d'instance

L'en-tête de données d'instance contient les informations essentielles de configuration de l'axe nécessaires à l'analyse et à l'application des données contenues dans les trois canaux de données (voir Figure 19). Cet en-tête est uniquement inclus dans le format Connexion variable afin de s'adapter aux applications d'entraînement multi-axe. Les informations qu'il contient peuvent être utilisées par l'interface de communication du dispositif d'entraînement pour copier les blocs de données individuels dans des mémoires fixes séparées pour le traitement.

S'il est configuré pour un format Connexion fixe, seul le Bloc de données cycliques d'une seule instance d'axe est pris en charge. Il n'est donc pas utile que les informations précisent le numéro d'instance ou la taille du bloc. L'en-tête de données d'instance n'est donc pas inclus.

En-tête de données d'instance			
Numéro d'instance	(Réservé)	Taille de bloc d'instance	Taille de bloc cyclique
Taille du bloc réel cyclique	Taille de bloc d'écriture cyclique	Taille de bloc d'événement	Taille de bloc de service

Figure 19 – En-tête de données d'instance

- **Numéro d'instance:** Il s'agit du numéro d'identification de l'instance Objet Axe de mouvement spécifique à laquelle s'applique le bloc de données d'instance suivant. Les instances Objet Axe de mouvement sont créées comme une série contiguë de numéros d'instance commençant par 1. Dans la structure de données de connexion, les numéros d'instance de chaque bloc de données d'instance consécutifs doivent être une séquence ordinale, c'est-à-dire 1, 2, 3, etc. (l'instance 0 est définie comme étant l'instance de classe et n'est en général pas utilisée dans le cadre des données de connexion). Par conséquent, en théorie, la connexion E-S CIP Motion peut proposer jusqu'à 255 instances.

- Taille de bloc d'instance: Cette valeur représente la taille du bloc de données d'instance, en mots de 32 bits, en-tête compris. La taille de bloc d'instance est utile lorsque le dispositif d'entraînement souhaite accéder directement au bloc de données d'instance suivant sans ajouter les tailles de bloc cyclique, d'événement et de service.
- Taille de bloc cyclique: Cette valeur représente la taille du bloc de données cycliques, en unités de mot de 32 bits, en-tête compris.
- Taille de bloc de consigne cyclique: Cette valeur représente la taille du bloc de données de consigne cycliques, en unités de mot de 32 bits, en-tête compris.
- Taille de bloc d'écriture cyclique: Cette valeur représente la taille du bloc de données d'écriture cycliques, en unités de mot de 32 bits, en-tête compris. Si la taille de bloc d'écriture cyclique est nulle, il n'est pas utile que le dispositif d'entraînement prenne en charge la fonction Lecture/écriture cyclique.
- Taille de bloc d'événement: Cette valeur représente la taille du bloc de données d'événement, en unités de mot de 32 bits, en-tête compris. Si la taille de bloc d'événement est nulle, il n'est pas utile que le dispositif d'entraînement prenne en charge la fonction d'événement.
- Taille de bloc de service: Cette valeur représente la taille du bloc de données de service, en unités de mot de 32 bits, en-tête compris. Si la valeur Taille de bloc de service est nulle, il n'existe aucune demande de service à traiter.

6.4.3.3.3 Bloc de données cycliques

L'en-tête de données cycliques, placé en haut du bloc de données cycliques, est toujours inclus, quel que soit le format de connexion. Cet en-tête contient les éléments essentiels relatifs au contenu du bloc de données cycliques de la connexion Contrôleur - Dispositif et de la connexion Dispositif - Contrôleur, et au contexte des données déterminé par le mode de commande et la Configuration de réaction (voir Figure 20). Il fournit également un mécanisme de contrôle de l'état de l'axe d'entraînement ciblé.

En-tête de données cycliques			
Mode de commande	Configuration de réaction	Commande d'axe	-
Ensemble de données de consigne	Ensemble de données réelles	Ensemble de données d'état	Commande d'interpolation
Données cycliques			

Figure 20 – En-tête de données cycliques

- Mode de commande: Cette valeur énumérée de 8 bits détermine le contexte de mode de commande des données de consigne configurées dans le contrôleur de mouvement. Cette valeur peut être modifiée en ligne, voire même lorsque l'axe d'entraînement est à l'état Exécution. Si une transition Mode de commande particulière n'est pas prise en charge par le dispositif d'entraînement, une exception Changement de mode interdit est générée, qui peut être configurée pour exécuter l'une d'un certain nombre d'actions en réponse à une transition interdite.

La valeur de cet élément est transférée vers l'attribut d'Objet Axe de mouvement du même nom. La définition de l'attribut Mode de commande complet est disponible dans les tables d'attribut de l'Objet Axe de mouvement (voir 7.3.2.2.1).

- Configuration de réaction: Cette valeur énumérée de 8 bits indique le contexte de réaction des données de consigne tel qu'il est configuré dans le contrôleur de mouvement. Les données de consigne peuvent faire référence à Réaction 1 ou Réaction 2 ou aux unités moteur pour les opérations sans capteur. Cette valeur peut être modifiée en ligne, voire même lorsque l'axe d'entraînement est à l'état Exécution. Si une transition Configuration de réaction particulière n'est pas prise en charge par le dispositif d'entraînement, une exception Changement de mode interdit est générée, qui peut être configurée pour exécuter l'une d'un certain nombre d'actions en réponse à une transition interdite.

La valeur de cet élément est transférée vers l'attribut d'Objet Axe de mouvement du même nom. La définition de l'attribut Configuration de réaction complet est disponible dans les tables d'attribut de l'Objet Axe de mouvement (voir 7.3.4.2).

- **Commande d'axe:** Ce code énuméré de 8 bits peut être utilisé pour lancer directement les opérations de changement d'état du dispositif d'entraînement n'impliquant pas de transmettre ou de renvoyer les paramètres du dispositif d'entraînement, et donc, de lancer un service CIP. Les énumérations valides de cet élément de données sont présentées au Tableau 6.

Tableau 6 – Commande d'axe

Code de demande	Opération demandée
0	Pas de demande
1	Demande d'activation
2	Demande de désactivation
3	Demande d'interruption
4	Demande de réinitialisation suite à une interruption
5	Demande d'abandon
6	Demande de réinitialisation suite à un défaut
7 à 127	(Réservé)
128 à 255	(Spécifique au fournisseur)

- **Ensemble de données de consigne:** Cette valeur de matriciel comporte un bit défini pour chaque référence de consigne en temps réel potentielle (voir Tableau 7). Les données de consigne apparaissent dans le même ordre dans l'ensemble de données de consigne que les numéros de bit. Ainsi, la Position de consigne apparaît avant le Couple de consigne dans la structure de données en temps réel de la connexion Contrôleur - Dispositif.

La valeur de cet élément est transférée vers l'attribut d'Objet Axe de mouvement "Ensemble de données de consigne".

Tableau 7 – Ensemble de données de consigne

Bit	Élément de données de consigne du contrôleur	Type de données
0	Position de consigne	DINT
1	Vitesse de consigne	REAL
2	Accélération de consigne	REAL
3	Couple de consigne	REAL
4	Compensation de position	DINT
5	Compensation de vitesse	REAL
6	Compensation d'accélération	REAL
7	Compensation de couple	REAL

Les éléments de données de consigne du contrôleur ci-dessus s'appliquent à la structure de données cycliques de la connexion Contrôleur - Dispositif et sont mis en correspondance avec les attributs correspondants de l'Objet Axe de mouvement (voir Tableau 8).

Les unités des éléments de données de consigne correspondent à celles définies pour l'attribut d'Objet Axe de mouvement associé.

Tableau 8 – Mise en correspondance de l'élément de données de consigne avec l'attribut d'Objet Axe de mouvement

Bit	Élément de données de consigne	Attribut d'Objet Axe de mouvement
0	Position de consigne	Consigne de position du contrôleur
1	Vitesse de consigne	Consigne de vitesse du contrôleur
2	Accélération de consigne	Consigne d'accélération du contrôleur
3	Couple de consigne	Consigne de couple du contrôleur
4	Compensation de position	Compensation de position
5	Compensation de vitesse	Compensation de vitesse
6	Compensation d'accélération	Compensation d'accélération
7	Compensation de couple	Compensation de couple

Il appartient au contrôleur de s'assurer que les données de connexion contiennent les éléments de données de consigne nécessaires à la prise en charge du mode de commande spécifié.

- Ensemble de données réelles: Cette valeur de matriciel comporte un bit défini pour chaque attribut potentiel de données réelles en temps réel à inclure dans l'ensemble de données réelles du bloc de données d'instance de la connexion Dispositif - Contrôleur dans l'actualisation suivante (voir Tableau 9). Les données réelles apparaissent dans le même ordre que les numéros de bit. Ainsi, la Position réelle apparaît avant le Couple réel dans la structure d'ensemble de données réelles. Ce mécanisme permet de modifier le contenu de l'ensemble de données réelles à tout moment pendant le fonctionnement du dispositif d'entraînement. La valeur de cet élément est transférée vers l'attribut d'Objet Axe de mouvement "Ensemble de données de réaction".

Tableau 9 – Ensemble de données réelles

Bit	Élément de données réelles produit	Type de données
0	Position réelle	DINT
1	Vitesse réelle	REAL
2	Accélération réelle	REAL
3	Couple réel	REAL
4	Courant réel	REAL
5	Tension réelle	REAL
6	Fréquence réelle	REAL
7	(Réservé)	

Les éléments de données réelles ci-dessus sont mis en correspondance avec les attributs correspondants de l'Objet Axe de mouvement (voir Tableau 10). Les unités des éléments de données réelles correspondent à celles définies pour l'attribut d'Objet Axe de mouvement associé.

Tableau 10 – Mise en correspondance de l'élément de données réelles avec l'attribut d'Objet Axe de mouvement

Bit	Élément de données réelles	Attribut d'Objet Axe de mouvement
0	Position réelle	Retour en position
1	Vitesse réelle	Retour en vitesse
2	Accélération réelle	Retour en accélération
3	Couple réel	Référence de couple
4	Courant réel	Référence de courant Iq
5	Tension réelle	Tension de sortie
6	Fréquence réelle	Fréquence de sortie
7	(Réservé)	

- Ensemble de données d'état: Cet octet de matriciel contient des indicateurs qui déterminent le contenu de l'ensemble de données d'état du bloc de données d'instance de la connexion Dispositif - Contrôleur dans l'actualisation suivante (voir Tableau 11). Les données d'état apparaissent dans le même ordre que les numéros de bit. Ainsi, le Statut de défaut d'axe apparaît avant, par exemple, les données Statut d'alarme d'axe dans la structure Ensemble de données d'état. Plusieurs attributs comprenant un élément de données d'état sélectionné sont transmis dans l'ordre de haut en bas. Ainsi, le Code de défaut d'axe est transmis avant le Code de défaut d'axe Mfg. La consultation de la spécification de l'attribut d'Objet Axe de mouvement correspondante fournit les définitions de chacun de ces éléments de données d'état (voir 7.3). La valeur de cet élément est transférée vers l'attribut d'Objet Axe de mouvement "Ensemble de données d'état".

Tableau 11 – Configuration de l'ensemble de données d'état

Bit	Élément de données d'état produit	Type de données
0	Code de défaut d'initialisation Code de défaut d'initialisation – Mfg Code de défaut d'axe Code de défaut d'axe – Mfg Code d'alarme d'axe Code d'alarme d'axe – Mfg Code d'inhibition de démarrage Code d'inhibition de démarrage – Mfg Datation de défaut d'axe	USINT USINT USINT USINT USINT USINT USINT USINT LINT
1	Statut de défaut d'axe Statut de défaut d'axe – Mfg	LWORD LWORD
2	Statut d'alarme d'axe Statut d'alarme d'axe – Mfg	LWORD LWORD
3	Statut d'inhibition de démarrage Statut d'inhibition de démarrage – Mfg	WORD WORD
4	Statut d'axe Statut d'axe – Mfg Statut d'E/S de l'axe Statut d'E/S de l'axe – Mfg	DWORD DWORD DWORD DWORD
5	(Réservé)	
6	(Réservé)	
7	(Spécifique au fournisseur)	

- Commande d'interpolation: L'octet contient les informations nécessaires au contrôle de l'algorithme d'interpolation fine et permet de déterminer le temps cible des données de consigne de la structure de commande d'entraînement (voir Tableau 12). La valeur de cet élément est transférée vers l'attribut d'Objet Axe de mouvement du même nom.

Tableau 12 – Commande d'interpolation

Bit	Définition	Syntaxe
0 à 1	Temps de consigne cible	0 = Immédiat 1 = Extrapoler (+1 période d'actualisation) 2 = Interpoler (+2 périodes d'actualisation)
3 à 5	(Réservé)	
6 à 7	(Spécifique au fournisseur)	

Temps de consigne cible – Cet entier de 2 bits définit un temps spécifique par rapport à la datation de connexion pour lequel les données de consigne sont ciblées, comme défini par le planificateur de mouvement du contrôleur. Le temps de consigne cible absolu est la somme de la datation du contrôleur depuis le contrôleur et le produit (Temps de consigne cible x Période d'actualisation du contrôleur).

- Un Temps de consigne cible nul implique que les données de consigne sont ciblées pour le début du cycle d'actualisation en cours et, par conséquent, nécessite d'être immédiatement appliqué à la structure de commande. Dans ce cas, il n'est pas utile de procéder à une interpolation fine. Cette situation peut se produire lorsque la période d'actualisation du contrôleur est plus courte que celle du dispositif ou que le mouvement contrôlé de l'axe est constant pendant la durée de la période d'actualisation du contrôleur.
- Un temps de consigne cible de 1 implique que la cible des données de consigne est l'événement de temporisation Actualisation de connexion suivant. Dans ce cas, l'interpolateur de consigne fonctionne principalement comme un extrapolateur qui estime la valeur suivante des données de consigne en fonction de la trajectoire présente de l'axe. Il s'agit d'un paramètre classique lorsque la période d'actualisation du contrôleur est comparable à celle du dispositif ou que le mouvement contrôlé de l'axe est relativement constant pendant la durée de la période d'actualisation du contrôleur.
- Un temps de consigne cible de 2 implique que la cible des données de consigne est composée de deux événements de temporisation Actualisation de connexion issus de la datation de connexion. Dans ce cas, l'interpolateur de consigne peut calculer une trajectoire lissée en fonction de la dynamique courante du moteur pour atteindre la valeur Données de consigne au moment ciblé. Il s'agit de la véritable interpolation fine, qui s'applique lorsque la période d'actualisation du contrôleur est bien plus longue que celle du dispositif.
- Données de consigne cycliques: Les données de consigne cycliques contiennent des données à priorité élevée devant être appliquées à l'axe d'entraînement associé lors de l'actualisation suivante du dispositif d'entraînement. Ce bloc est composé d'éléments de données de consigne appliqués en référence aux algorithmes de contrôle du dispositif d'entraînement. Il est déterminé de manière explicite par l'élément Ensemble de données de consigne dans l'En-tête de données de consigne cycliques.

6.4.3.3.4 Bloc de données d'écriture cycliques

Le bloc de données d'écriture cycliques peut être utilisé pour mettre à jour de manière synchrone un ou plusieurs paramètres de configuration d'Objet Axe de mouvement ciblés dans le dispositif d'entraînement. Ce mécanisme peut être utilisé conjointement avec un programme Bloc de fonctions pour mettre en œuvre des algorithmes sophistiqués de contrôle de boucle externe, de séquençement de gain et de limitation dynamique. A l'inverse des demandes de service Définition d'attribut d'axe du canal de service, dont le traitement peut durer plusieurs cycles d'actualisation de dispositif d'entraînement, le mécanisme de données d'écriture cycliques garantit l'application du paramètre ciblé à la prochaine actualisation de dispositif d'entraînement disponible.

Le bloc de données d'écriture cycliques est uniquement pris en charge dans le format Connexion variable (voir Figure 21).

Bloc de données d'écriture cycliques	
ID de bloc d'écriture cyclique	ID de bloc de lecture cyclique
Données d'écriture cycliques	

Figure 21 – Bloc de données d'écriture cycliques

L'en-tête associé de ce bloc contient des éléments essentiels relatifs au contenu du bloc de données d'écriture cycliques.

- ID de bloc d'écriture cyclique: Cet ID de 16 bits détermine la structure prédéfinie du bloc d'écriture cyclique à appliquer aux données d'écriture cycliques pour cette actualisation. Les structures de bloc d'écriture cyclique sont définies à l'aide du service Liste de définition des données d'écriture cycliques. Une réponse correcte à ce service inclut un nouvel ID de bloc d'écriture cyclique qui peut être utilisé dans l'actualisation de connexion suivante pour transmettre les données cycliques dans ce format. L'ID de bloc d'écriture cyclique précédent et sa structure associée doivent être conservés tant que le contrôleur ne demande pas la transmission du nouvel ID de bloc d'écriture cyclique, l'ancien ID et la définition de structure n'étant alors plus obligatoires.
- ID de bloc de lecture cyclique: Cet ID de 16 bits détermine la structure prédéfinie du bloc de lecture cyclique à appliquer aux données de lecture cycliques pour l'actualisation de la connexion Dispositif - Contrôleur suivante. Les structures de bloc de lecture cyclique sont définies à l'aide du service Définition du bloc de données de lecture cycliques. Une réponse correcte à ce service inclut un nouvel ID de bloc de lecture cyclique qui peut être utilisé dans l'actualisation de la connexion suivante pour permettre au dispositif d'entraînement d'utiliser le nouveau format de données de lecture cycliques pour l'actualisation de la connexion Dispositif - Contrôleur disponible suivante.
- Données d'écriture cycliques: Les données d'écriture cycliques contiennent des données à priorité élevée devant être appliquées à l'axe d'entraînement associé lors de l'actualisation suivante du dispositif d'entraînement. Ce bloc est composé d'éléments de données paramétriques appliqués aux attributs d'Objet Axe de mouvement utilisés par les algorithmes de contrôle. Le contenu des données d'écriture cycliques est déterminé de manière explicite par la structure identifiée par l'ID de bloc d'écriture cyclique se trouvant dans l'en-tête de données d'écriture cycliques.

6.4.3.4 Bloc de données d'événement

Le bloc de données d'événement permet d'acheminer des informations concernant le canal d'événements. En particulier, le bloc de données d'événement de la connexion Contrôleur - Dispositif permet de contrôler l'armement des fonctions de vérification des événements, et d'acquitter la réception des notifications d'événement provenant du dispositif d'entraînement envoyées par l'intermédiaire du bloc de données d'événement de la connexion Dispositif - Contrôleur.

Le format du bloc de données d'événement de la connexion Contrôleur - Dispositif est représenté à la Figure 22.

Bloc de données d'événement			
Commande de vérification d'événement			
Acquittement d'événement ID1	Acquittement d'événement État 1	Acquittement d'événement ID2	Acquittement d'événement État 2
...			

Figure 22 – Bloc de données d'événement

- Commande de vérification d'événement: Ce mot de 32 bits est copié dans l'attribut d'Objet Axe de mouvement du même nom utilisé pour activer différentes entrées du dispositif d'entraînement (les entrées de marqueur et d'enregistrement, par exemple) afin de générer des événements. Lorsque ces événements se produisent, le dispositif d'entraînement capture l'heure et la position exacte de l'axe lors du déroulement de l'événement. Les 4 derniers bits de la commande de vérification d'événement représentent

une valeur binaire qui spécifie le nombre d'événements actifs, et qui est littéralement le nombre d'ID d'acquiescement d'événement figurant dans ce bloc de données d'événement. Une définition exhaustive de l'attribut Commande de vérification d'événement est donnée en 7.3.5.2.1.

Le mécanisme de commande d'événement fonctionne comme suit:

- 1) Le contrôleur définit le bit de vérification d'événement approprié pour rechercher une condition d'erreur spécifique. Plusieurs bits du mot Commande de vérification d'événement peuvent être définis à un instant donné.
- 2) Le dispositif d'entraînement détecte que le bit de demande Commande de vérification d'événement est défini, puis exécute l'action de vérification d'événement demandée.
- 3) Le dispositif d'entraînement définit le bit Statut de vérification d'événement correspondant pour confirmer qu'il procède à présent à la vérification active de la condition d'événement spécifiée.
- 4) Le contrôleur détermine que le bit Statut de vérification d'événement provenant du dispositif d'entraînement est défini, indiquant que le déclencheur d'événements est "armé" au niveau du dispositif d'entraînement. En commençant par cette actualisation, le contrôleur peut traiter toutes les notifications d'événement provenant du dispositif d'entraînement qui correspondent à la condition d'événement spécifiée.
- 5) Le dispositif d'entraînement détecte la condition d'événement spécifiée, incrémente l'ID d'événement, puis envoie au contrôleur un bloc de données de notification d'événement avec cet ID. Si le bit Réarmement automatique associé est vide dans le mot Commande de vérification d'événement, le dispositif d'entraînement interrompt la vérification de l'événement spécifié. Toutefois, le bit Statut de vérification d'événement est toujours défini tant que le bit Demande de vérification d'événement n'est pas effacé par le contrôleur. Les notifications des autres événements peuvent être envoyées dans la même actualisation.
- 6) Le contrôleur traite les notifications d'événement, envoie l'acquiescement d'événement au dispositif d'entraînement et, si le réarmement automatique n'est pas activé, supprime le bit Vérification d'événement associé pour l'actualisation de connexion Contrôleur - Dispositif suivante.
- 7) Le dispositif d'entraînement détecte que le bit Demande de vérification d'événement est annulé et efface le bit Statut de vérification d'événement correspondant avant l'actualisation de connexion Dispositif - Contrôleur suivante.
- 8) Le contrôleur est en mesure de définir de nouveau le bit Vérification d'événement dès qu'il constate que le bit Statut de vérification d'événement associé a de nouveau été défini. Par conséquent, la durée minimale entre la réception de la notification d'événement par le contrôleur et le réarmement du dispositif d'entraînement pour l'événement suivant est une période d'actualisation du contrôleur.

Dans le cas d'un événement d'enregistrement où l'événement de réarmement automatique est demandé, la séquence de gestion d'événement se présente comme suit:

- 1) Le contrôleur définit le bit Vérification d'événement approprié pour rechercher une condition d'événement spécifique à déclencher, et définit également le bit Réarmement automatique pour cette condition d'événement.
- 2) Le dispositif d'entraînement détecte que le bit de demande Commande de vérification d'événement est défini, puis exécute l'action de vérification d'événement demandée.
- 3) Le dispositif d'entraînement définit les bits Statut de vérification d'événement et Réarmement automatique correspondants pour confirmer qu'il procède à présent à la vérification active de la condition d'événement spécifiée.
- 4) Le contrôleur détermine que le bit Statut de vérification d'événement provenant du dispositif d'entraînement est défini, indiquant que le déclencheur d'événements est "armé" au niveau du dispositif d'entraînement. En commençant par cette actualisation, le contrôleur peut traiter toutes les notifications d'événement

provenant du dispositif d'entraînement qui correspondent à la condition d'événement spécifiée.

- 5) Le dispositif d'entraînement détecte la condition d'événement spécifiée, incrémente l'ID d'événement, puis envoie au contrôleur un bloc de données de notification d'événement avec cet ID. Etant donné que le bit Réarmement automatique est défini, le dispositif d'entraînement poursuit la vérification de l'événement spécifié. Le bit Statut de vérification d'événement est toujours défini tant que le bit Demande de vérification d'événement n'est pas effacé par le contrôleur. Si le dispositif d'entraînement détecte une autre condition d'événement spécifiée avant la transmission du premier événement au contrôleur, il incrémente de nouveau l'ID d'événement, puis ajoute une autre notification d'événement avec cet ID au bloc de données d'événement du contrôleur. Etant donné que le bit Réarmement automatique est défini, le dispositif d'entraînement poursuit la vérification de l'événement spécifié.
 - 6) Le contrôleur traite la notification d'événement, envoie l'acquiescement d'événement au dispositif d'entraînement mais, étant donné que le bit Réarmement automatique est activé, laisse le bit Vérification d'événement associé défini pour l'actualisation de connexion Contrôleur - Dispositif suivante.
 - 7) Le dispositif d'entraînement reçoit l'acquiescement d'événement de la part du contrôleur. Il peut désormais envoyer une autre notification d'événement si l'événement spécifié s'est de nouveau produit. Sinon, passer à l'étape suivante.
 - 8) Le dispositif d'entraînement continue la vérification de la condition d'événement spécifiée et envoie un bloc de données de notification d'événement au contrôleur à chaque fois que la condition d'événement se produit. Le bit Statut de vérification d'événement est toujours défini tant que le bit Demande de vérification d'événement n'est pas effacé par le contrôleur. Avec la fonction de réarmement automatique, la vérification d'événement est activée en permanence, ce qui assure de ne manquer aucun événement d'enregistrement pendant le délai normal d'un cycle de réarmement du mécanisme de vérification d'événement. La fonction de réarmement automatique présente l'inconvénient de pouvoir générer une multitude d'événements dont le traitement ne présente aucun intérêt, en fait, plusieurs événements spécifiés par période d'actualisation du contrôleur. Par conséquent, ces événements doivent être placés dans la mémoire tampon du contrôleur dans un attribut de matrice d'événement ou filtrés en fonction des données d'événement. Cette dernière possibilité porte sur la manière dont la fonction Enregistrement en mosaïque est mise en œuvre. Cette fonction vérifie tous les événements d'enregistrement provenant du dispositif d'entraînement pour voir si la Position d'événement donne une valeur de position absolue dans les limites de la fenêtre d'enregistrement configurée. Si la position d'enregistrement calculée n'entre pas dans la fenêtre, l'événement est ignoré. Si la position d'enregistrement calculée entre dans la fenêtre, elle est stockée dans le contrôleur avec la datation d'enregistrement.
- ID d'acquiescement d'événement: Numéro de transaction affecté à cet événement par la notification d'événement d'origine. Un nouvel ID d'événement est affecté à chaque événement par incrémentation de l'ID d'événement en cours stocké dans le dispositif d'entraînement. Grâce à l'ID d'événement, le dispositif d'entraînement est en mesure de mettre en correspondance l'acquiescement d'événement et la notification d'événement appropriée pour compléter la transaction de données d'événement.
 - Statut d'acquiescement d'événement: Valeur énumérée indiquant la réponse du contrôleur à l'événement. Une valeur nulle indique que le traitement de l'événement a abouti. Une valeur non nulle indique qu'une erreur s'est produite dans le traitement de l'événement et que l'événement doit être de nouveau envoyé.

6.4.3.5 Bloc de données de service

Le bloc de données de service permet d'envoyer une demande de service par instance au dispositif d'entraînement dans une actualisation donnée. La demande de service requiert une réponse de service spécifique de la part du dispositif d'entraînement, indiquant un succès ou une erreur. Dans certains cas, le service de réponse contient les données demandées. Dans

tous les cas, les données de demande de service sont conservées dans la structure de données de connexion Contrôleur - Dispositif tant que le contrôleur ne reçoit pas la réponse de service associée de la part du dispositif d'entraînement.

Chaque demande de service est représentée par un bloc de données organisé comme indiqué à la Figure 23.

NOTE Par nature, les 4 premiers octets du bloc de données de service ne suivent pas le format de messagerie CIP standard traditionnel. La raison principale est que cette structure de connexion est une connexion E-S CIP implicite et pas une connexion de messagerie explicite. Toutefois, dans le cas d'un format de connexion fixe, les données de demande spécifique au service définies ci-dessous sont envoyées par l'intermédiaire d'une connexion de messagerie explicite et suivent les règles CIP en matière de format de demande de service explicite.

Bloc de données de service			
ID de transaction	Code de service	-	-
Données de demande spécifique au service			

Figure 23 – Bloc de données de service

- ID de transaction: Numéro de transaction affecté à cette demande de service par le contrôleur. Un nouvel ID de transaction est affecté à chaque demande de service par incrémentation de l'ID de transaction en cours stocké dans le contrôleur. Grâce à cet ID de transaction, le contrôleur est en mesure d'associer la réponse de service à la demande de service appropriée et de procéder à la transaction de service.
- Code de service: Identificateur qui détermine la demande de service spécifique à l'objet suivante. La liste des codes de service pris en charge est présentée dans le paragraphe Services spécifiques à l'objet de la présente partie de la série CEI 61800-7 (voir 7.5.1). Les services communs CIP ne s'appliquent pas au bloc de données de service.
- Données de demande spécifique au service: Le format et la syntaxe des données de demande spécifique au service dépendent du code de service spécifié. Cela est vrai, que les données de demande spécifique au service soient transmises dans le cadre de la connexion Contrôleur - Dispositif ou d'une connexion de messagerie explicite.

6.4.4 Connexion Dispositif - Contrôleur

6.4.4.1 Généralités

A l'instar de la structure de données de connexion Contrôleur - Dispositif décrite ci-dessus, la connexion Dispositif - Contrôleur CIP Motion est organisée comme indiqué à la Figure 24.

Format de connexion Contrôleur - Dispositif CIP Motion	
En-tête de connexion	
Blocs de données d'instance	

Figure 24 – Format de connexion Contrôleur - Dispositif CIP Motion

6.4.4.2 En-tête de connexion Dispositif - Contrôleur

6.4.4.2.1 Structure générale

L'en-tête de connexion Dispositif - Contrôleur contient les informations essentielles de configuration de l'axe nécessaires à l'analyse du bloc de données de connexion Dispositif - Contrôleur. La partie fixe de l'en-tête de connexion est définie comme illustré à la Figure 25.

En-tête de connexion			
Format de connexion	Révision de format	ID d'actualisation	Statut de nœud

Figure 25 – En-tête de connexion

- Format de connexion: Identique à la définition Contrôleur - Dispositif, sauf que la valeur requise pour le type de connexion est 3 (type de connexion Dispositif - Contrôleur fixe) ou 7 (type de connexion Dispositif – Contrôleur fixe).

- 0 = Réservé
- 1 = Réservé
- 2 = Connexion Contrôleur - Dispositif fixe
- 3 = Connexion Dispositif - Contrôleur fixe
- 4 = Réservé
- 5 = Réservé
- 6 = Connexion Contrôleur - Dispositif variable
- 7 = Connexion Dispositif - Contrôleur variable
- 8 à 15 = Réservés.

- Révision de format: Le numéro de révision de format est 1 pour tous les dispositifs qui utilisent la présente édition de la CEI 61800-7-202. Cette valeur doit être incrémentée de 1 à chaque révision du format de connexion Dispositif - Contrôleur ayant un impact sur l'interface. La révision de format permet aux contrôleurs les plus récents de prendre en charge les formats de connexion générés par des dispositifs d'entraînement plus anciens. Il permet également aux anciens contrôleurs de reconnaître un format de connexion récent provenant d'un dispositif d'entraînement qu'il ne peut pas prendre en charge et de générer une erreur appropriée à cette application.
- ID d'actualisation: Il convient de mettre en correspondance l'ID d'actualisation de la connexion Dispositif - Contrôleur avec l'ID d'actualisation de la connexion Contrôleur - Dispositif précédent, et doit donc être incrémenté à chaque période d'actualisation. En cas de perte ou de retard du paquet Contrôleur - Dispositif associé, l'ID d'actualisation Dispositif - Contrôleur doit être incrémenté comme si ce paquet était arrivé à temps. Cela permet au système de commande CIP Motion de passer sur le paquet Contrôleur - Dispositif perdu ou manqué et de maintenir la synchronisation avec les ID d'actualisation correspondants.

L'ID d'actualisation s'apparente au nombre de séquences de messages CIP et permet au contrôleur de déterminer si la mémoire tampon de connexion contient de nouvelles données. Si l'ID d'actualisation n'a pas été modifié, le contrôleur tente de passer sur l'actualisation manquée par extrapolation en fonction de la précédente trajectoire, tant que la nouvelle actualisation n'est pas arrivée. Si l'axe n'est pas synchronisé ou qu'il ne prend pas en charge les services de synchronisation, les données de datation ne sont pas incluses ni valides, aucune tentative de passage par extrapolation n'ayant donc lieu. Dans ce cas, l'ID d'actualisation est également le seul moyen dont dispose le contrôleur pour détecter de nouvelles données de connexion.
- Statut de nœud: Contient les bits utilisés pour indiquer le statut du nœud de communication du dispositif d'entraînement associé. La valeur de cet élément est déduite de l'attribut de classe d'Objet Axe de mouvement du même nom. Voir 7.2.2.4 (Objet Axe de mouvement) pour plus de détails.

6.4.4.2.2 En-tête de connexion fixe

Si le format de connexion est une connexion Dispositif - Contrôleur fixe, l'en-tête ci-dessus est immédiatement suivi du bloc de données d'instance.

6.4.4.2.3 En-tête de connexion variable

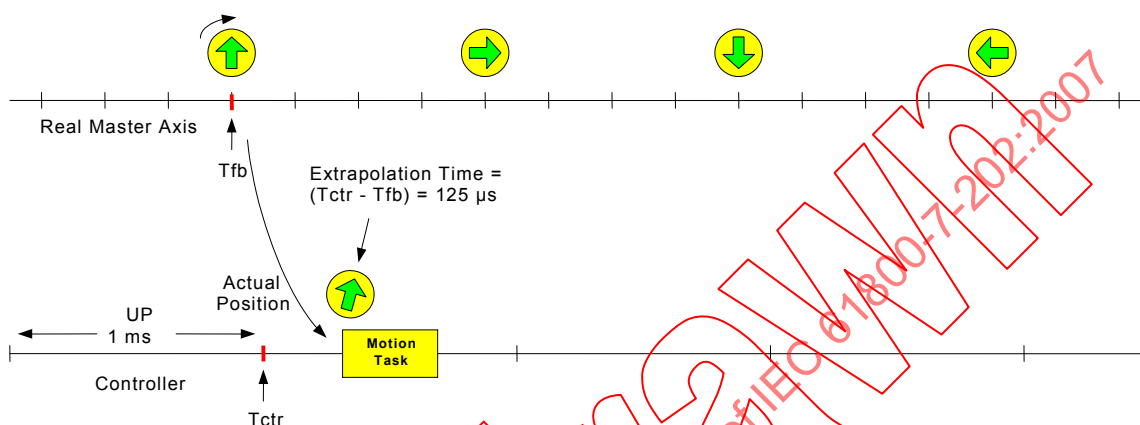
Si le format de connexion est une connexion Dispositif - Contrôleur variable, l'en-tête de connexion contient des champs supplémentaires relatifs à l'adressage et à la datation d'entraînement multi-axe (voir Figure 26).

En-tête de connexion			
<i>Format de connexion</i>	<i>Révision de format</i>	<i>ID d'actualisation</i>	<i>Contrôle de nœud</i>
Comptage d'instances	Alarmes de nœud	Défauts de nœud	Ensemble de données temporelles
Période d'actualisation du dispositif			
Décalage temporel du dispositif			
Datation du dispositif			
Datation reçue des données			
Datation de transmission des données			
Variation de synchronisation			

Figure 26 – En-tête de connexion

- Comptage d'instances: Identique à la définition Contrôleur - Dispositif.
- Alarmes de nœud: Contient les bits utilisés pour indiquer les conditions d'alarme du nœud de communication du dispositif d'entraînement associé. La valeur de cet élément est déduite de l'attribut de classe d'Objet Axe de mouvement du même nom. Voir 7.2.2.6 (Objet Axe de mouvement) pour plus de détails.
- Défauts de nœud: Contient les bits utilisés pour indiquer les conditions de défaut du nœud de communication du dispositif d'entraînement associé. La valeur de cet élément est déduite de l'attribut de classe d'Objet Axe de mouvement du même nom. Voir 7.2.2.5 (Objet Axe de mouvement) pour plus de détails.
- Ensemble de données temporelles: Identique à la définition Contrôleur - Dispositif.
- Période d'actualisation du dispositif: Cet élément représente la période en cours entre les calculs de commande d'entraînement qui appliquent les données de consigne. Cette valeur permet au contrôleur de configurer le comportement des blocs d'interpolation fine appliqués aux données de consigne. En règle générale, si la période d'actualisation du contrôleur est beaucoup plus longue que la période d'actualisation du dispositif, l'interpolation fine est appliquée. Si la période d'actualisation du contrôleur est comparable à la période d'actualisation du dispositif, l'interpolation fine n'est pas nécessaire et seule l'extrapolation est requise pour procéder aux corrections de datation.
- Décalage temporel du dispositif: Cet élément représente la valeur de décalage du temps système de 64 bits associée à la datation du dispositif qui suit. La valeur Décalage temporel du dispositif permet au contrôleur de déterminer si le temps système défini dans le dispositif d'entraînement est étalé par rapport à celui du contrôleur. En principe, le temps système entre le dispositif d'entraînement et le contrôleur sont proches, quel que soit le temps considéré. Mais, conformément au protocole de la CEI 61588:2004, l'horloge maîtresse du système peut corriger toutes les secondes la référence du temps système, parfois d'une durée significative (une heure, par exemple). En effet, même le maître lui-même peut changer si une horloge maîtresse de qualité supérieure est détectée par le système. Ces modifications brusques apportées à la référence du temps système se propagent vers les différents dispositifs présents sur le réseau, de sorte que le contrôleur et le dispositif d'entraînement fonctionnent avec des valeurs de temps système étalées. La valeur Décalage temporel du dispositif peut être utilisée conjointement avec la valeur Décalage du temps système du contrôleur, afin de déterminer si le temps système entre le dispositif d'entraînement et le contrôleur est étalé et, si c'est le cas, pour compenser cet étalement.
- Datation du dispositif: Cette valeur de datation représente la valeur de temps système de 64 bits, en nanosecondes, lorsque l'événement de temporisation d'actualisation du dispositif d'entraînement associé aux données réelles de la structure de connexion s'est produit (lorsque les données réelles ont été capturées, par exemple). Calculée par le dispositif d'entraînement, il s'agit de la somme de la valeur d'horloge locale du dispositif d'entraînement en cas d'événement de temporisation d'actualisation et de la valeur Décalage du temps système du dispositif d'entraînement. Avec la datation du dispositif, le contrôleur dispose de toutes les informations nécessaires à la correction des valeurs de données réelles pour connaître les différences entre la temporisation d'actualisation du dispositif d'entraînement et celle du contrôleur lorsque la période d'actualisation du contrôleur n'est pas un entier multiple de la période d'actualisation du dispositif ou que les

actualisations de dispositif d'entraînement présentent un déphasage par rapport au contrôleur. Ce modèle de temporisation suppose d'enregistrer la datation du dispositif au début de la période d'actualisation du dispositif d'entraînement, qui est également le moment où la réaction a été capturée pour la dernière fois. Si la datation du dispositif ne correspond pas à la datation d'actualisation locale du contrôleur, ce dernier extrapole la valeur instantanée de données de réponse en fonction de la trajectoire de manière à correspondre à la datation du contrôleur. Le diagramme de temporisation de la Figure 27 illustre comment le contrôleur ajuste les données de position de l'axe provenant du dispositif d'entraînement en fonction des datations relatives entre le dispositif d'entraînement et le contrôleur.



Légende

Anglais	Français
Real master axis	Axe du maître réel
Extrapolation Time	Temps d'extrapolation
Actual Position	Position réelle
Controller	Contrôleur
Motion Task	Tâche de mouvement

Figure 27 – Ajustement des données de position réelles en fonction de la datation du dispositif

- Datation reçue des données: Cet élément représente la valeur de temps système de 64 bits à l'arrivée des données de connexion Contrôleur - Dispositif au dispositif d'entraînement et de leur traitement. La datation est exprimée en nanosecondes. Cette valeur, combinée à la datation de transmission de données correspondante du contrôleur, peut être utilisée par le contrôleur pour générer des statistiques de livraison de données de connexion Contrôleur - Dispositif.
- Datation de transmission des données: Cet élément représente la valeur de temps système de 64 bits au moment de la transmission au contrôleur des données de connexion Dispositif - Contrôleur. La datation est exprimée en nanosecondes. Cette valeur, combinée à la datation reçue de données correspondante du contrôleur, peut être utilisée par le contrôleur pour générer des statistiques de livraison de données de connexion Dispositif - Contrôleur.
- Variation de synchronisation: Cet élément représente la variation statistique en cours de la valeur Décalage du temps système par rapport à la valeur Décalage de temps système moyenne.

6.4.4.3 Blocs de données d'instance

6.4.4.3.1 Structure générale

L'En-tête de connexion est suivi d'un ou de plusieurs Blocs de données d'instance, comme le montre le Comptage d'instances ci-dessus. Le Bloc de données d'instance s'apparente beaucoup à celui de la connexion Contrôleur - Dispositif, sa structure de base étant présentée à la Figure 28.

Bloc de données d'instance
En-tête de données d'instance
Bloc de données cycliques
Bloc de données d'événement
Bloc de données de service

Figure 28 – Bloc de données d'instance

6.4.4.3.2 En-tête de données d'instance

L'En-tête de données d'instance contient les informations essentielles de configuration de l'axe nécessaires à l'analyse et à l'application des données contenues dans les trois canaux de données (voir Figure 29). Cet en-tête est uniquement inclus dans le format Connexion variable afin de s'adapter aux applications d'entraînement multi-axe. Les informations qu'il contient peuvent être utilisées par l'interface de communication du dispositif d'entraînement pour copier les blocs de données individuels dans des mémoires fixes séparées pour le traitement.

S'il est configuré pour un format Connexion fixe, seul le Bloc de données cycliques d'une seule instance d'axe est pris en charge. Les informations relatives au numéro d'instance ou à la taille du bloc ne sont donc pas utiles. Par conséquent, l'En-tête de données d'instance n'est pas inclus dans la structure de connexion.

En-tête de données d'instance			
Numéro d'instance		Taille de bloc d'instance	Taille de bloc cyclique
Taille du bloc réel cyclique	Taille de bloc de lecture cyclique	Taille de bloc d'événement	Taille de bloc de service

Figure 29 – En-tête de données d'instance

- Numéro d'instance: Identique à la définition Contrôleur - Dispositif.
- Taille de bloc d'instance: Identique à la définition Contrôleur - Dispositif.
- Taille de bloc cyclique: Identique à la définition Contrôleur - Dispositif.
- Taille de bloc réel cyclique: Identique à la définition Contrôleur - Dispositif.
- Taille de bloc de lecture cyclique: Identique à la définition Contrôleur - Dispositif.
- Taille de bloc d'événement: Identique à la définition Contrôleur - Dispositif.
- Taille de bloc de service: Identique à la définition Contrôleur - Dispositif.

6.4.4.3.3 Bloc de données cycliques

L'en-tête de données cycliques, placé en haut du bloc de données cycliques de la connexion Dispositif - Contrôleur, est toujours inclus, quel que soit le format de connexion. Cet en-tête contient les éléments essentiels relatifs au contenu du bloc de données cycliques et au contexte des données dans le bloc concernant le dispositif d'entraînement (voir Figure 30). Pour la plupart, ces éléments sont établis par les éléments correspondant du précédent bloc de données cycliques de la connexion Contrôleur - Dispositif, et en sont donc des copies directes. Par conséquent, le contenu du bloc de données cycliques de la connexion Dispositif - Contrôleur est en dernier lieu déterminé par le contrôleur.

Bloc de données cycliques			
Mode de commande	Configuration de réaction	Réponse d'axe	Statut de réponse
-	Ensemble de données réelles	Ensemble de données d'état	État de l'axe
(Réservé)			
Données réelles cycliques			
Données de statut cycliques			

Figure 30 – Bloc de données cycliques

- Mode de commande: Identique à la définition Contrôleur - Dispositif.
- Configuration de réaction: Identique à la définition Contrôleur - Dispositif.
- Réponse d'axe: La réponse d'axe à 8 bits est une valeur énumérée utilisée pour établir une liaison avec l'élément Commande d'axe correspondant de la connexion Contrôleur - Dispositif afin d'initier directement le fonctionnement du dispositif d'entraînement n'impliquant pas de demande de service CIP. Les codes d'acquiescement valides correspondent aux codes de demande correspondants de l'élément Commande d'axe (voir le Tableau 13).

Tableau 13 – Réponse d'axe

Code d'acquiescement	Réponse d'axe
0	Pas d'acquiescement
1	Activer l'acquiescement
2	Désactiver l'acquiescement
3	Acquiescement d'interruption
4	Acquiescement de réinitialisation suite à l'interruption
5	Acquiescement d'abandon
6	Acquiescement de réinitialisation suite à un défaut
7 à 127	(Réservé)
128 à 255	(Spécifique au fournisseur)

Le mécanisme Commande d'axe/Réponse d'axe permettant d'initier les modifications d'état est décrit de manière exhaustive dans le paragraphe Commande d'état de l'Objet Axe de mouvement (voir 7.6.1.2).

- Statut de réponse: En présence d'un Code d'acquiescement non nul dans la Réponse d'axe, une valeur Statut de réponse est également fournie pour indiquer le succès ou l'échec de l'opération Commande d'axe. Un Statut de réponse nul (0) indique le succès, une valeur non nulle indiquant une erreur. Les valeurs Statut de réponse sont conformes à la spécification CIP des codes Statut général (voir CEI 61158-6-2).
- Ensemble de données réelles: Identique à la définition Contrôleur - Dispositif.
- Ensemble de données d'état: Identique à la définition Contrôleur - Dispositif.
- État de l'axe: Cet élément de données contient la valeur énumérée État de l'axe indiquant l'état en cours de l'axe de ce dispositif d'entraînement en fonction du modèle d'état de l'Objet Axe de mouvement.
- Données d'état/réelles cycliques: Les données réelles/d'état cycliques contiennent des données à priorité élevée associées à l'axe du dispositif d'entraînement lors de l'actualisation suivante du dispositif d'entraînement. Ce bloc est composé des éléments de données réelles et des éléments de données d'état utilisés par le contrôleur, comme le déterminent de manière explicite les éléments Ensemble de données réelles et Ensemble

de données d'état de l'En-tête de données cycliques. Voir les définitions Contrôleur - Dispositif de 6.4.3 pour plus de détails de la structure Données réelles/d'état cycliques.

6.4.4.3.4 Bloc de données de lecture cycliques

Le bloc de données de lecture cycliques peut être utilisé pour actualiser de manière synchrone un ou plusieurs attributs d'Objet Axe de commande de mouvement ciblés dans le contrôleur en fonction de la valeur en cours des attributs associés dans le dispositif d'entraînement. Ce mécanisme peut par exemple être utilisé conjointement avec un programme de bloc de fonction conforme à la CEI 61131-3 afin de mettre en œuvre un contrôle de boucle externe sophistiqué en fonction d'un large éventail de signaux de commande d'entraînement disponibles. A l'inverse des demandes de service Extraction d'attribut d'axe du canal de service, dont le traitement peut durer plusieurs cycles d'actualisation de dispositif d'entraînement, le mécanisme de données de lecture cycliques garantit l'actualisation du paramètre ciblé à chaque cycle de connexion.

Le bloc de données de lecture cycliques est uniquement pris en charge dans le format Connexion variable (voir Figure 31).

Bloc de données de lecture cycliques	
ID de bloc de lecture cyclique	
Données de lecture cycliques	

Figure 31 – Bloc de données de lecture cycliques

L'en-tête associé de ce bloc contient des éléments essentiels relatifs au contenu du bloc de données d'écriture cycliques.

- ID de bloc de lecture cyclique: Cet ID de 16 bits détermine la structure prédéfinie du bloc de lecture cyclique à appliquer aux données de lecture cycliques pour cette actualisation. Les structures de bloc de lecture cyclique sont définies à l'aide du service Définition de la liste de données de lecture cycliques. Une réponse correcte à ce service inclut un nouvel ID de bloc de lecture cyclique qui peut être utilisé dans l'actualisation de connexion suivante pour transmettre les données cycliques dans ce format.
- Données de lecture cycliques: Les Données de lecture cycliques contiennent des données à priorité élevée devant être appliquées à l'instance d'axe de contrôleur associée. Ce bloc est composé d'un signal et d'éléments de données d'état à adapter et appliquer aux attributs d'Objet Axe de mouvement correspondants. Le contenu des données de lecture cycliques est déterminé de manière explicite par la structure identifiée par l'ID de bloc de lecture cyclique se trouvant dans l'en-tête de données de lecture cycliques.

6.4.4.4 Bloc de données d'événement

Le bloc de données d'événement permet d'envoyer plusieurs notifications d'événement au contrôleur dans le cadre d'une actualisation donnée. Chaque notification d'événement implique un acquittement d'événement spécifique indiquant le succès ou une erreur. Ces données de notification d'événement restent dans la structure de données de connexion Contrôleur - Dispositif tant que le dispositif d'entraînement n'a pas reçu l'acquiescement d'événement correspondant de la part du contrôleur.

Le format du bloc de données d'événement de la connexion Dispositif - Contrôleur est représenté à la Figure 32.

Bloc de données d'événement			
Statut de vérification d'événement			
ID d'événement	Statut d'événement	Type d'événement	-
Position d'événement			
Datation de l'événement			
...			

Figure 32 – Bloc de données d'événement

- Statut de vérification d'événement: Ce mot de 32 bits est un paramètre matriciel permettant d'indiquer si le dispositif d'entraînement recherche actuellement les événements en fonction des différentes entrées du dispositif d'entraînement (les entrées de marqueur, d'origine et d'enregistrement, par exemple). La vérification d'événement est initiée lorsque le bit Commande de vérification d'événement correspondant est défini dans la connexion Contrôleur - Dispositif. Lorsqu'un événement se produit, le dispositif d'entraînement capture l'heure et la position exacte de l'axe, puis transmet ces informations au contrôleur dans des blocs de données d'événement. Mais pour que le contrôleur puisse traiter les données d'événement, le bit de Statut de vérification d'événement correspondant doit être défini. Pour plus de détails sur la manière d'utiliser ce mot dans les opérations d'événement, voir la séquence de mécanisme d'événement en 6.4.3.4. Une définition exhaustive de l'attribut Statut de vérification d'événement est donnée en 7.3.5.2.2.
- ID d'événement: Numéro de transaction affecté à cet événement par la notification d'événement d'origine. Un ID d'événement est affecté à chaque événement par incrémentation de l'ID d'événement en cours stocké dans le dispositif d'entraînement. Grâce à l'ID d'événement, le dispositif d'entraînement est en mesure de mettre en correspondance l'acquiescement d'événement et la notification d'événement appropriée pour compléter la transaction de données d'événement.
- Statut d'événement: Valeur énumérée indiquant la réponse du contrôleur à l'événement. Une valeur nulle indique que le traitement de l'événement a abouti. Une valeur non nulle indique qu'une erreur s'est produite dans le traitement de l'événement et que l'événement doit être de nouveau envoyé.
- Type d'événement: Cette valeur énumérée décrit le type d'événement qui s'est produit. Des types d'événement valides sont spécifiés dans le Tableau 14.

Tableau 14 – Type d'événement

Type d'événement	Description de l'événement
0	Front positif d'enregistrement 1
1	Front négatif d'enregistrement 1
2	Front positif d'enregistrement 2
3	Front négatif d'enregistrement 2
4	Front positif du marqueur
5	Front négatif du marqueur
6	Front positif de l'interrupteur d'origine
7	Front négatif de l'interrupteur d'origine
8	Marqueur - Interrupteur d'origine + +
9	Marqueur - Interrupteur d'origine + -
10	Marqueur - Interrupteur d'origine - +
11	Marqueur - Interrupteur d'origine - -
12 à 127	(Réservé)
128 à 255	(Spécifique au fournisseur)

- Position d'événement: Représentation entière de 32 bits de la position de l'axe lorsque l'événement désigné s'est produit.
- Datation de l'événement: Cet élément représente la valeur de temps système de 64 bits lorsque l'événement spécifié s'est produit. La datation est exprimée en nanosecondes. S'il est accompagné de la valeur Décalage du temps système du dispositif d'entraînement, qui fait partie intégrante de l'en-tête Connexion Dispositif - Contrôleur, le contrôleur dispose de toutes les informations nécessaires pour calculer le temps système absolu lorsque l'événement s'est produit.

6.4.4.5 Bloc de données de service

Le bloc de données de service permet d'envoyer une réponse de service par instance au contrôleur dans une actualisation donnée. Chaque demande de service requiert une réponse de service spécifique de la part du dispositif d'entraînement, indiquant un succès ou une erreur. Dans certains cas, le service de réponse contient les données demandées. Dans tous les cas, les données de réponse de service restent dans la structure de données de connexion Dispositif - Contrôleur tant que le dispositif d'entraînement n'a pas vu que la demande de service associée a été supprimée du bloc de données d'instance de connexion Contrôleur - Dispositif (Taille de bloc de service = 0) ou qu'une nouvelle demande de service n'a pas été émise par le contrôleur (ID de transaction incrémenté).

Chaque réponse de service est représentée par un bloc de données organisé comme indiqué à la Figure 33.

NOTE A l'instar de la structure de demande, la structure de la réponse de service ne suit pas le format de messagerie standard CIP traditionnel. La raison principale est que cette structure de connexion est une connexion E-S CIP implicite et pas une connexion de messagerie explicite. Toutefois, dans le cas d'un format de connexion fixe, les données de demande spécifique au service définies ci-dessous sont envoyées par l'intermédiaire d'une connexion de messagerie explicite et suivent les règles CIP en matière de format de demande de service explicite.

Bloc de données de service			
ID de transaction	Code de service	Statut général	Statut étendu
Données de réponse spécifique au service			

Figure 33 – Bloc de données de service

- ID de transaction: Numéro de transaction affecté à cette réponse de service déduite de l'ID de transaction de la demande d'origine. Un nouvel ID de transaction est affecté à chaque demande de service par incrémentation de l'ID de transaction en cours stocké dans le contrôleur. Grâce à l'ID de transaction de la réponse, le contrôleur est en mesure d'associer la réponse de service à la demande de service appropriée.
- Code de service: Identificateur qui détermine la réponse de service spécifique qui suit, dont il convient qu'elle corresponde au code de service de la demande de service d'origine. Une liste des codes de service valides pour l'Objet Axe de mouvement est donnée dans le paragraphe Connexion Contrôleur - Dispositif.
- Statut général: La valeur Statut général indique le succès ou l'échec de la demande de service. Un Statut général nul (0) indique le succès, une valeur non nulle indiquant une erreur. Les valeurs Statut général sont conformes à la spécification CIP des codes de statut général (voir CEI 61158-6-2).
- Statut général étendu: Le statut général étendu offre une méthode de définition des codes d'erreur spécifiques au fournisseur ou au service. Ces codes ne font actuellement l'objet d'aucune définition standard.
- Données de réponse spécifique au service: Le format et la syntaxe des données de réponse spécifiques au service dépendent du code de service spécifié.

6.4.5 Format de connexion de dispositif fixe

En spécifiant un format de connexion fixe, la connexion E-S CIP Motion peut être réduite à une taille aisément applicable aux réseaux CIP basse performance (DeviceNet et ControlNet, par exemple). Dans le contexte d'un réseau DeviceNet ou ControlNet, les fonctions suivantes ont été retirées de la structure de connexion pour prendre en charge les exigences d'une taille de connexion fixe et d'une bande passante de réseau limitée.

- Datation
- Défauts/Alarmes de nœud
- Une instance uniquement
- Taille de bloc dynamique

- Bloc de données de lecture/écriture cycliques
- Bloc de données d'événement
- Bloc de données de service

Avec le format de connexion fixe, les demandes de service adressées à l'Objet Axe de mouvement sont prises en charge uniquement en tant que Messagerie explicite.

La Figure 34 et la Figure 35 présentent un exemple de format de connexion fixe utilisé dans une simple application de dispositif d'entraînement à vitesse variable impliquant une seule commande de vitesse et donnant la vitesse réelle. Dans ce cas, la taille de connexion a été réduite à 16 octets, taille bien adaptée aux réseaux basse performances (DeviceNet, par exemple).

Structure de connexion Contrôleur - Dispositif			
Format de connexion	Révision de format	ID d'actualisation	Contrôle de nœud
Mode de commande	Configuration de réaction	Commande d'axe	-
Ensemble de données de consigne	Ensemble de données réelles	Ensemble de données d'état	-
Vitesse de consigne			

Figure 34 – Format de connexion Contrôleur – Dispositif fixe (taille fixe = 16 octets)

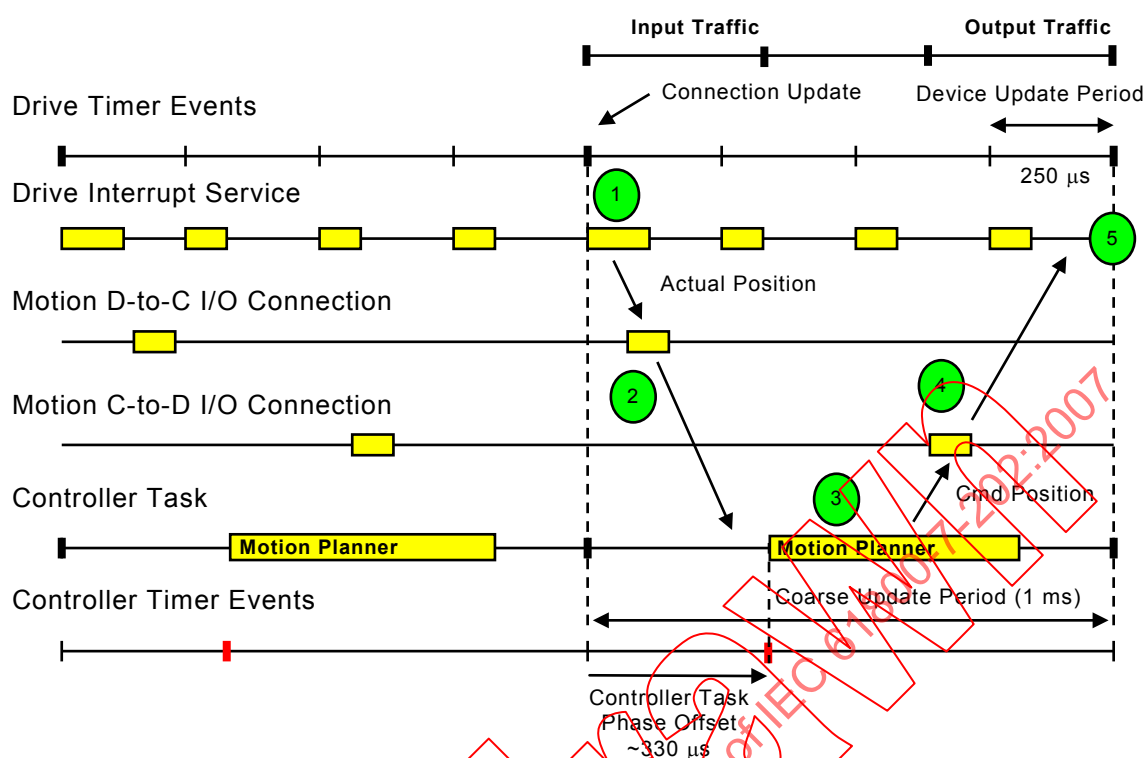
Structure de connexion Dispositif - Contrôleur			
Format de connexion	Révision de format	ID d'actualisation	Statut de nœud
Mode de commande	Configuration de réaction	Réponse d'axe	Statut de réponse
-	Ensemble de données réelles	Ensemble de données d'état	État de l'axe
Vitesse réelle			

Figure 35 – Format de connexion Dispositif – Contrôleur fixe (taille fixe = 16 octets)

6.4.6 Temporisations de la connexion CIP Motion

6.4.6.1 Généralités

Le modèle de temporisation général de l'échange de données de connexion E-S CIP Motion est décrit dans ce paragraphe. L'échange de données entre le dispositif d'entraînement et le contrôleur est assuré par le contrôleur avec un paquet de données Dispositif - Contrôleur envoyé par paquet de données Contrôleur - Dispositif reçu. Les paquets de connexion Contrôleur - Dispositif sont envoyés périodiquement en fonction de la période d'actualisation du contrôleur configurée. La période d'actualisation du dispositif, c'est-à-dire la période d'actualisation à laquelle le dispositif d'entraînement exécute ses calculs de commande, est en général plus rapide que la période d'actualisation du contrôleur. Le modèle de temporisation de connexion CIP Motion de base est illustré à la Figure 36.



Légende

Anglais	Français
Drive Timer Events	Événements de temporisateur du dispositif d'entraînement
Input Traffic	Trafic en entrée
Output Traffic	Trafic en sortie
Connection Update	Actualisation de connexion
Device Update Period	Période d'actualisation du dispositif
Drive Interrupt Service	Service d'interruption de dispositif d'entraînement
Motion D-to-C I/O Connection	Connexion E-S D-C CIP Motion
Motion C-to-D I/O Connection	Connexion E-S C-D CIP Motion
Controller Task	Tâche de contrôleur
Controller Timer Events	Événements de temporisateur du contrôleur
Actual Position	Position réelle
Cmd Position	Position de consigne
Motion Planner	Planificateur de mouvement
Coarse Update Period	Période d'actualisation brute
Phase Offset	Décalage de phase

Figure 36 – Modèle de temporisation de la connexion E-S CIP Motion

La séquence de connexion CIP Motion de base est composée de cinq étapes de base illustrées à la Figure 36 et décrites ci-après:

- 1) Un Service d'interruption de dispositif d'entraînement commence en réponse à un Événement de temporisateur d'entraînement périodique, le dispositif d'entraînement déterminant qu'une transmission Actualisation brute au contrôleur est requise. Le dispositif d'entraînement calcule la valeur Position réelle (en supposant le Mode d'asservissement de position de cet exemple) en fonction de la valeur de compte de réaction capturée avec l'événement du temporisateur de service d'interruption.

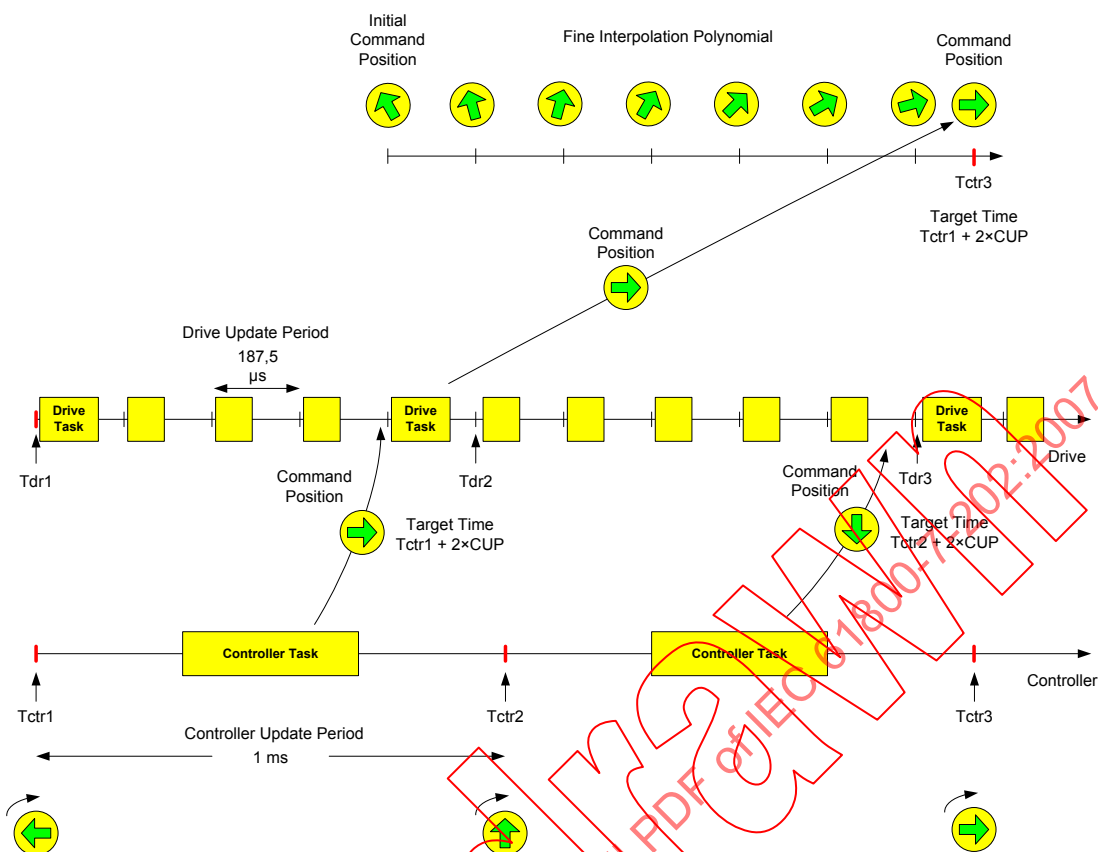
- 2) Une fois chargées les données d'entrée de position réelle dans la structure de connexion Dispositif - Contrôleur, le dispositif d'entraînement transmet les données au contrôleur. Le paquet Données d'entrée traverse le réseau et arrive au contrôleur avant le début de la tâche de contrôleur retardée par phase.
- 3) Après une durée de déphasage prédéterminée à partir du début du dernier événement de temporisateur du contrôleur, qui se situe en général au tiers de la période d'actualisation brute environ, la tâche du contrôleur commence. La tâche commence par la lecture des données d'entrée de position réelle du dispositif d'entraînement. Ensuite, le contrôleur lance un planificateur de mouvement pour calculer une nouvelle valeur Position de consigne à renvoyer au dispositif d'entraînement. Si les systèmes d'engrenage ou de came sont actifs, il peut être nécessaire d'utiliser la position réelle d'un axe maître en entrée pour calculer la position de consigne d'un ou de plusieurs axes esclaves.
- 4) Une fois chargées les données de sortie de position de consigne dans la structure de connexion Contrôleur - Dispositif, le contrôleur transmet les données au dispositif d'entraînement. Le paquet Données de sortie traverse le réseau et arrive au dispositif d'entraînement avant le début de la période d'actualisation brute suivante.
- 5) Le service d'interruption de dispositif d'entraînement suivant commence, et le dispositif d'entraînement détermine qu'il a reçu de nouvelles données de sortie de la part du contrôleur. Ensuite, le dispositif d'entraînement lit les données de position de consigne et les applique avec la datation à son système d'interpolation/extrapolation fine.

Les autres modèles de temporisation pouvant éventuellement reposer sur la configuration du contrôleur, tous les modèles de temporisation CIP Motion commencent par les données réelles envoyées par le dispositif d'entraînement au contrôleur au début de la période d'actualisation du contrôleur, et se terminent par les données de consigne arrivant avant la période d'actualisation du contrôleur suivante. Toutefois, il n'est pas obligatoire que les données de consigne envoyées au dispositif d'entraînement dans une période d'actualisation donnée soient le résultat des calculs exécutés pendant cette période.

La plupart des protocoles de commande de mouvement requièrent que la période d'actualisation du contrôleur soit un entier multiple de la période d'actualisation du dispositif. Mais étant donné que le paquet de connexion E-S CIP Motion contient une datation, il n'est pas utile que la période d'actualisation du contrôleur entretienne une relation fixe avec la période d'actualisation du dispositif d'entraînement.

6.4.6.2 Temporisation de connexion Contrôleur - Dispositif

L'échange de données de connexion E-S CIP Motion est initié par le contrôleur via le paquet de connexion Contrôleur - Dispositif. L'introduction d'informations de datation et de décalage de temps, ainsi que des données de consigne, dans ce paquet permet de dégager des exigences de temporisation rigoureuses imposées par d'autres protocoles de réseau de commande de mouvement. La Figure 37 illustre l'application des données de consigne et des datations délivrées par la connexion Contrôleur - Dispositif à l'axe d'entraînement lorsqu'une interpolation fine est obligatoire.



Légende

Anglais	Français
Initial Command Position	Position de consigne initiale
Fine Interpolation Polynomial	Polynôme d'interpolation fine
Command Position	Position de consigne
Target Time	Temps cible
Drive Update Period	Période d'actualisation du dispositif d'entraînement
Drive	Dispositif d'entraînement
Controller Update Period	Période d'actualisation du contrôleur
Controller	Contrôleur

Figure 37 – Temporisation de connexion Contrôleur - Dispositif avec interpolation fine

Les étapes suivantes décrivent en détail la manière dont les données de connexion sont transférées du contrôleur au dispositif d'entraînement pour l'interpolation fine lors d'un cycle de connexion classique, dans le cas général d'une période d'actualisation du contrôleur (CUP) qui n'est pas un entier multiple de la période d'actualisation du dispositif.

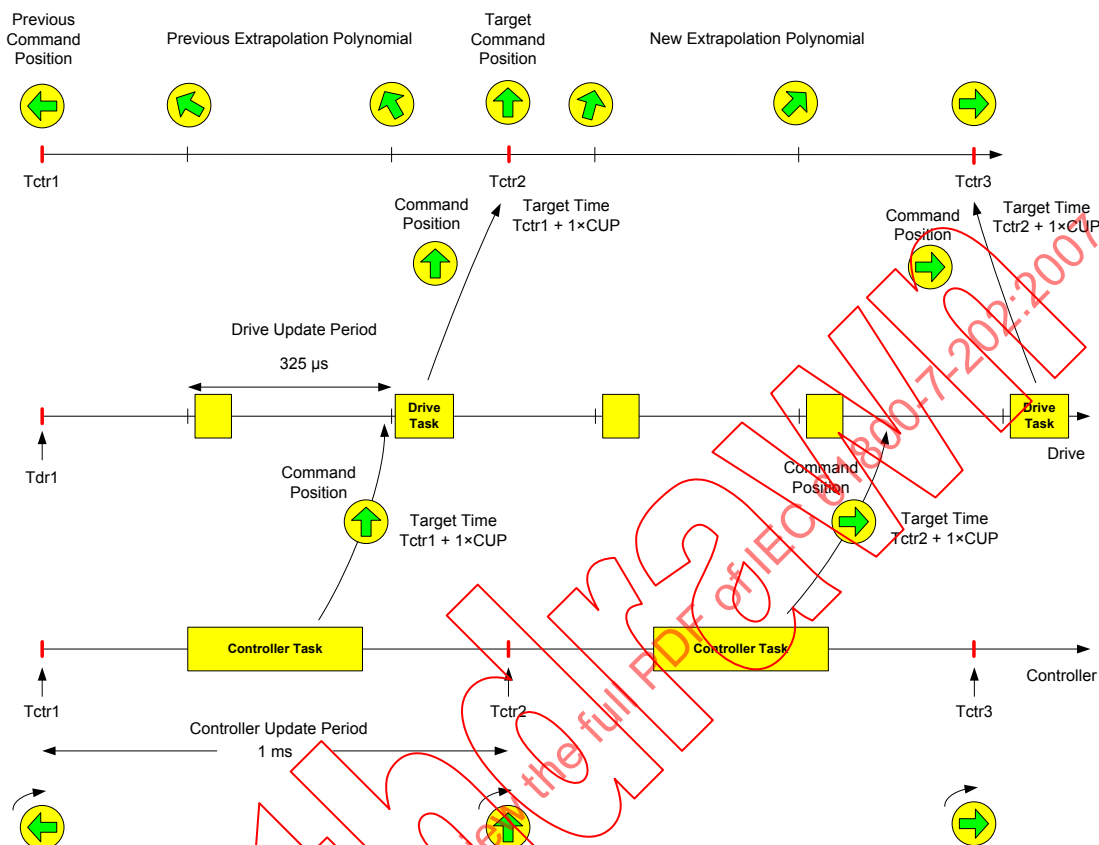
- 1) Transmission du contrôleur: dans le cadre de la tâche de contrôle, le contrôleur lance la transmission d'un paquet de connexion Contrôleur - Dispositif avec de nouvelles données de consigne au dispositif d'entraînement ciblé, avec un ID d'actualisation incrémenté et de nouvelles datations de système de contrôleur et décalage de temps système du contrôleur faisant référence au temps système au début de la période d'actualisation du contrôleur en cours. Le bloc de données d'instance de l'axe ciblé contient également le temps de consigne cible qui est, dans cet exemple, établie à +2 afin de tenir compte du délai de transport de la période d'actualisation du contrôleur et du délai de la période d'actualisation du contrôleur pour l'interpolation fine.
- 2) Vérification de l'ID d'actualisation de consigne: Le dispositif d'entraînement procède à une tâche périodique de dispositif d'entraînement qui vérifie les nouvelles données de

paquet de connexion Contrôleur - Dispositif de chaque période d'actualisation du dispositif. Cette opération peut être facilitée par une vérification d'ID d'actualisation modifié. Si la tâche de dispositif d'entraînement détecte de nouvelles données, il s'agit alors d'un cycle d'actualisation des données de consigne, lesquelles doivent être traitées conformément à la séquence suivante.

- 3) Vérification de fonctionnement synchrone: Ensuite, le dispositif d'entraînement vérifie s'il est synchronisé. Si ce n'est pas le cas, passer à l'étape Appliquer les données de consigne, puisqu'il n'est pas utile de procéder à une vérification de décalage de temps système ou à une vérification d'actualisation retardée. Éviter ces vérifications permet de contrôler le dispositif d'entraînement lors du démarrage ou que le dispositif d'entraînement ne dispose d'aucun service de synchronisation temporelle.
- 4) Vérification de décalage de temps système: Un fonctionnement synchrone implique une mise en correspondance du temps système du dispositif d'entraînement et du contrôleur, de manière à rendre significatives les datations échangées entre les dispositifs. Étant donné que le temps système peut être ajusté à tout instant par l'horloge maîtresse et que les ajustements peuvent se propager vers les différents dispositifs du réseau à différents moments, il est possible que le temps système entre le contrôleur et le dispositif d'entraînement soit étalé. Cet étalement doit être détecté, auquel cas, le dispositif d'entraînement doit ajuster la datation du contrôleur afin de le compenser. Voir 6.4.6.6 (Compensation du décalage du temps système) pour plus de détails sur cet algorithme.
- 5) Vérification d'actualisation retardée: En supposant le caractère synchrone du fonctionnement, le dispositif d'entraînement calcule la différence entre la datation d'actualisation de dispositif d'entraînement en cours et la datation du contrôleur transmise dans le paquet de connexion Contrôleur - Dispositif (ajusté, si nécessaire). Si la différence est supérieure à (Limite supérieure du délai d'actualisation du contrôleur x Période d'actualisation du contrôleur), le dispositif d'entraînement émet un Défaut d'actualisation du contrôleur. Si la différence de temps a dépassé deux fois la période d'actualisation de la connexion, le polynôme d'interpolation fine en cours est effectivement devenu un polynôme d'extrapolation, permettant au dispositif d'entraînement de passer à la condition de données retardées tant que les nouvelles données ne sont pas arrivées.
- 6) Appliquer les données de consigne: En supposant le caractère synchrone du fonctionnement et un temps de consigne cible de +2 pour l'interpolation fine, le dispositif d'entraînement calcule les coefficients du polynôme d'interpolation fine en fonction de la référence de consigne appliquée au temps cible, qui est la somme de la datation du contrôleur (ajusté) (CUP), T_{ctrl1} , et du produit du temps de consigne cible et de la période d'actualisation du contrôleur ou, dans ce cas, $2 \times CUP$. Si le temps cible est inférieur au temps système en cours dans le dispositif d'entraînement, de nouveaux coefficients par rapport au polynôme sont calculés en fonction de ces données de consigne, afin d'améliorer l'exactitude des calculs d'extrapolation. En général, à chaque retard des données de consigne, les données représentées toujours les données de consigne les plus récentes disponibles et doivent être appliquées dès que possible. En cas de fonctionnement asynchrone, les données de consigne sont immédiatement appliquées à la structure de commande d'entraînement.
- 7) Planifier l'actualisation Dispositif - Contrôleur suivante: En supposant le caractère synchrone de ce fonctionnement, le dispositif d'entraînement planifie l'heure d'envoi du paquet de connexion Dispositif - Contrôleur suivant au contrôleur. L'heure de transmission cible est la somme de la datation du contrôleur (ajustée), T_{ctrl1} , et de la période d'actualisation du contrôleur (CUP). Étant donné que, dans ce cas général, la période d'actualisation du contrôleur n'est pas un entier multiple de la période d'actualisation du dispositif, l'heure de transmission cible doit se présenter sous la forme d'une fenêtre de temporisation, définie comme étant la Fenêtre d'actualisation réelle, qui s'étend sur 1 période d'actualisation du dispositif et se termine à l'heure calculée de l'actualisation du contrôleur suivante. Dans le cas d'un fonctionnement asynchrone, le dispositif d'entraînement transmet simplement le paquet de connexion Dispositif - Contrôleur suivant.
- 8) Suivre la séquence de temporisation Dispositif - Contrôleur de 6.4.6.3.

Si le temps de consigne cible est défini sur +1, le polynôme calculé à l'étape 6 n'est pas appliqué pour les besoins de l'interpolation fine, mais plutôt pour ceux de l'extrapolation. Le polynôme d'extrapolation permet au dispositif d'entraînement de calculer une valeur de

données de consigne exacte lorsqu'il procède à ses calculs de commande en fonction de la trajectoire de l'axe précédente. La Figure 38 illustre ce modèle de temporisation dans le cas général où la période d'actualisation du contrôleur (CUP) n'est pas un entier multiple de la période d'actualisation du dispositif d'entraînement.



Légende

Anglais	Français
Previous Command Position	Position de consigne précédente
Previous extrapolation Polynomial	Polynôme d'extrapolation précédent
Target Command Position	Position de consigne cible
New Extrapolation Polynomial	Nouveau polynôme d'extrapolation
Command Position	Position de consigne
Target Time	Temps cible
Drive Update Period	Période d'actualisation du dispositif d'entraînement
Drive	Dispositif d'entraînement
Controller Update Period	Période d'actualisation du contrôleur
Controller	Contrôleur
Controller task	Tâche du contrôleur

Figure 38 – Temporisation de connexion Contrôleur – Dispositif avec périodes d'actualisation inégales

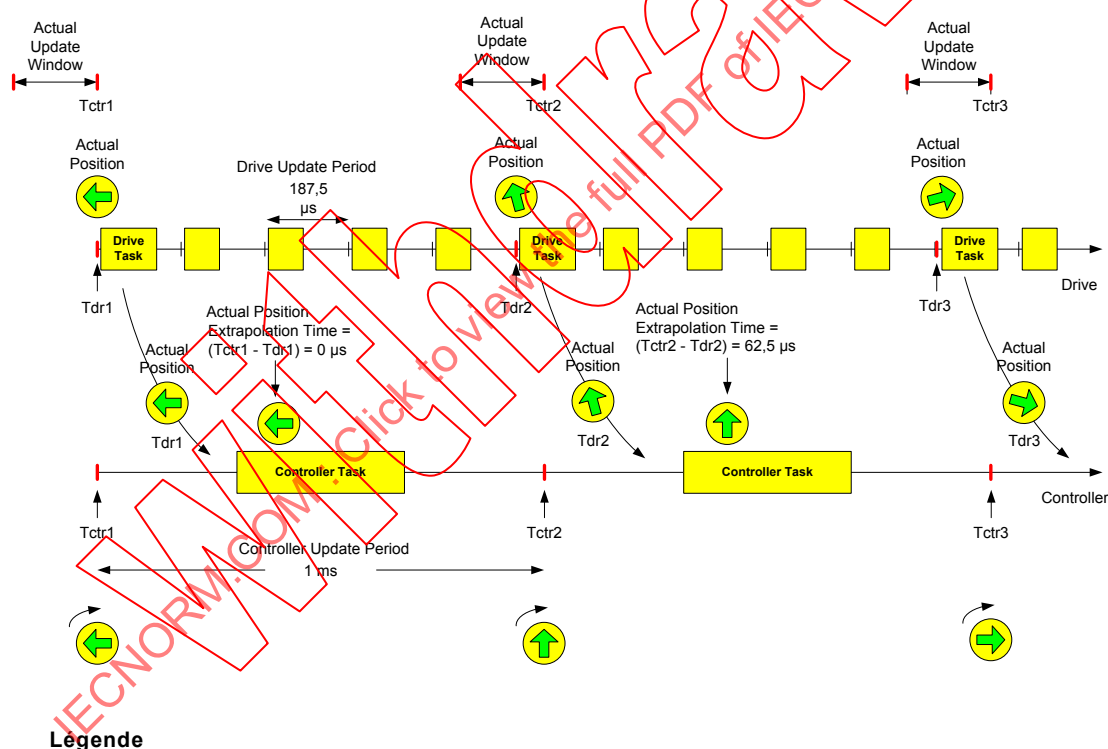
Dans l'exemple ci-dessus, la période d'actualisation du contrôleur ne contient pas beaucoup de périodes d'actualisation du dispositif. Si c'est le cas, l'interpolation fine n'est pas essentielle pour les performances du dispositif d'entraînement, et les données de consigne peuvent être appliquées plus directement à la structure de commande du dispositif d'entraînement sans délai supplémentaire requis pour la prise en charge de l'interpolation.

fine. Toutefois, l'extrapolation présente l'inconvénient de révéler plus directement les erreurs d'extrapolation aux données de consigne résultant d'un mouvement plus grossier que dans le cas d'une interpolation fine.

Toutes les données cycliques associées au paquet de connexion Contrôleur - Dispositif doivent être appliquées dans l'actualisation de la commande Tâche du dispositif d'entraînement pour utiliser le plus tôt possible les dernières données de consigne, en calculant de nouveaux coefficients de polynôme d'interpolation/extrapolation.

6.4.6.3 Temporisation de connexion Dispositif - Contrôleur

En mode synchrone, la connexion Dispositif - Contrôleur CIP Motion inclut une datation du dispositif (et un décalage temporel du dispositif) avec les données réelles, permettant au contrôleur de déterminer la position de l'axe du dispositif d'entraînement au moment de l'actualisation de la tâche du contrôleur. La datation permet au dispositif d'échantillonner la réaction et de calculer les valeurs de données réelles en fonction de sa propre période d'actualisation du dispositif qui, à l'inverse des autres protocoles de réseau de commande de mouvement, n'a pas besoin d'être strictement liée à la période d'actualisation du contrôleur. La Figure 39 illustre l'utilisation des données réelles et des datations délivrées par la connexion Dispositif - Contrôleur pour ajuster la position réelle de l'axe du dispositif d'entraînement, par exemple, à la base de temps du contrôleur.



Légende

Anglais	Français
Actual Update Window	Fenêtre d'actualisation réelle
Actual Position	Position réelle
Drive Update Period	Période d'actualisation du dispositif d'entraînement
Actual Position Extrapolation Time	Temps d'extrapolation de la position réelle
Controller task	Tâche du contrôleur
Controller update period	Période d'actualisation du contrôleur
Controller	Contrôleur

Figure 39 – Utilisation de la datation pour ajuster la position réelle à la base de temps du contrôleur

Les étapes suivantes décrivent en détail la manière dont les données de connexion sont transférées du dispositif d'entraînement au contrôleur lors d'un cycle de connexion classique, dans le cas général d'une période d'actualisation du contrôleur (CUP) qui n'est pas un entier multiple de la période d'actualisation du dispositif. Cette séquence d'étapes commence où la description Contrôleur - Dispositif s'arrête (voir 6.4.6.2).

- 1) Vérification de fenêtre d'actualisation réelle: Si l'axe est synchronisé, le dispositif d'entraînement compare la datation de la tâche de dispositif d'entraînement en cours à la Fenêtre d'actualisation réelle déterminée lors de la dernière actualisation des données de consigne. La fenêtre d'actualisation réelle dure 1 période d'actualisation du dispositif et se termine à l'heure calculée de l'actualisation du contrôleur suivante. Si la datation de la tâche de dispositif d'entraînement se trouve dans la fenêtre temporelle, il s'agit d'un cycle d'actualisation des données réel. Si la datation précède la fenêtre, le dispositif d'entraînement doit attendre une tâche de dispositif d'entraînement suivante pour envoyer les données réelles au contrôleur (cela évite les durées excessives entre la capture de réaction et le début de la tâche de contrôleur suivante). Si l'axe n'est pas synchronisé et qu'une actualisation de consigne vient d'être reçue par l'intermédiaire de la connexion Contrôleur - Dispositif, il s'agit également d'un cycle d'actualisation réel, et le dispositif d'entraînement doit donc passer à l'étape Transmission de dispositif d'entraînement.
- 2) Transmission de dispositif d'entraînement: S'il s'agit d'un cycle d'actualisation réel, le dispositif d'entraînement envoie le paquet de connexion Dispositif - Contrôleur au contrôleur avec les dernières données réelles à partir de cette tâche de dispositif d'entraînement, y compris, s'il est synchronisé, la datation de dispositif en cours (et le décalage temporel du dispositif), ainsi qu'un ID d'actualisation incrémenté. S'il n'est pas synchronisé, envoyer simplement l'ID d'actualisation du paquet de connexion Contrôleur - Dispositif précédent. Toutes les autres données envoyées au contrôleur dans ce paquet peuvent être déduites de la tâche de dispositif d'entraînement précédente. Cela permet de déclencher la transmission de dispositif d'entraînement au dernier point d'exécution de la tâche de dispositif d'entraînement.
- 3) Vérification de l'ID d'actualisation réel: Dans la tâche de contrôleur suivante, le contrôleur vérifie les nouvelles données provenant du dispositif d'entraînement en recherchant un ID d'actualisation modifié. La procédure ci-dessous est exécutée, que l'ID d'actualisation ait été modifié ou non. L'ID d'actualisation est le seul moyen de détecter les nouvelles données réelles lorsque le dispositif d'entraînement n'est pas synchronisé.
- 4) Vérification d'indicateur de synchronisation: Le dispositif d'entraînement vérifie le bit Mode sync de l'octet Statut de nœud de dispositif d'entraînement, afin de déterminer si l'axe d'entraînement est synchronisé. Si ce n'est pas le cas, passer à l'étape Appliquer les données réelles pour éviter la vérification d'actualisation retardée et la correction de datation. Éviter les étapes suivantes permet au dispositif d'entraînement de fonctionner lors du démarrage ou même si le dispositif d'entraînement ne dispose d'aucun service de synchronisation temporelle.
- 5) Vérification de décalage de temps système: Un fonctionnement synchrone implique une mise en correspondance du temps système du dispositif d'entraînement et du contrôleur, de manière à rendre significatives les datations échangées entre les dispositifs. Étant donné que le temps système peut être ajusté à tout instant par l'horloge maîtresse et que les ajustements peuvent se propager vers les différents dispositifs du réseau à différents moments, il est possible que le temps système entre le contrôleur et le dispositif d'entraînement soit étalé. Cet étalement doit être détecté, auquel cas, le dispositif d'entraînement doit ajuster la datation du dispositif afin de le compenser. Voir 6.4.6.6 (Compensation du décalage du temps système) pour plus de détails sur cet algorithme.
- 6) Vérification d'actualisation retardée: Le contrôleur calcule la différence entre la datation de la période d'actualisation de connexion en cours et la datation du paquet de connexion Dispositif - Contrôleur. Si la différence est supérieure à (Tolérance d'actualisation manquée x Période d'actualisation), le contrôleur émet un Défaut de synchronisation de contrôleur.

- 7) Correction de datation: Si la différence de temps préalablement calculée est non nulle, extrapoler la valeur de données réelles en fonction de la précédente trajectoire réelle de l'axe pour s'aligner sur la datation du contrôleur. Cette correction est nécessaire, le planificateur de mouvement supposant que les données d'entrée réelles sont horodatées de manière implicite au début de la période d'actualisation du contrôleur.
- 8) Appliquer les données réelles: Le contrôleur applique les données réelles en entrée au planificateur de mouvement, qui calcule les nouvelles données de référence de consigne pour l'actualisation Contrôleur - Dispositif suivante.

6.4.6.4 Indépendance de la période d'actualisation du dispositif

Le diagramme de temporisation de la Figure 40 illustre comment les deux axes d'entraînement peuvent être très coordonnés malgré différentes périodes d'actualisation du dispositif, et malgré une période d'actualisation du contrôleur associée qui n'est pas un entier multiple de la période d'actualisation du dispositif.

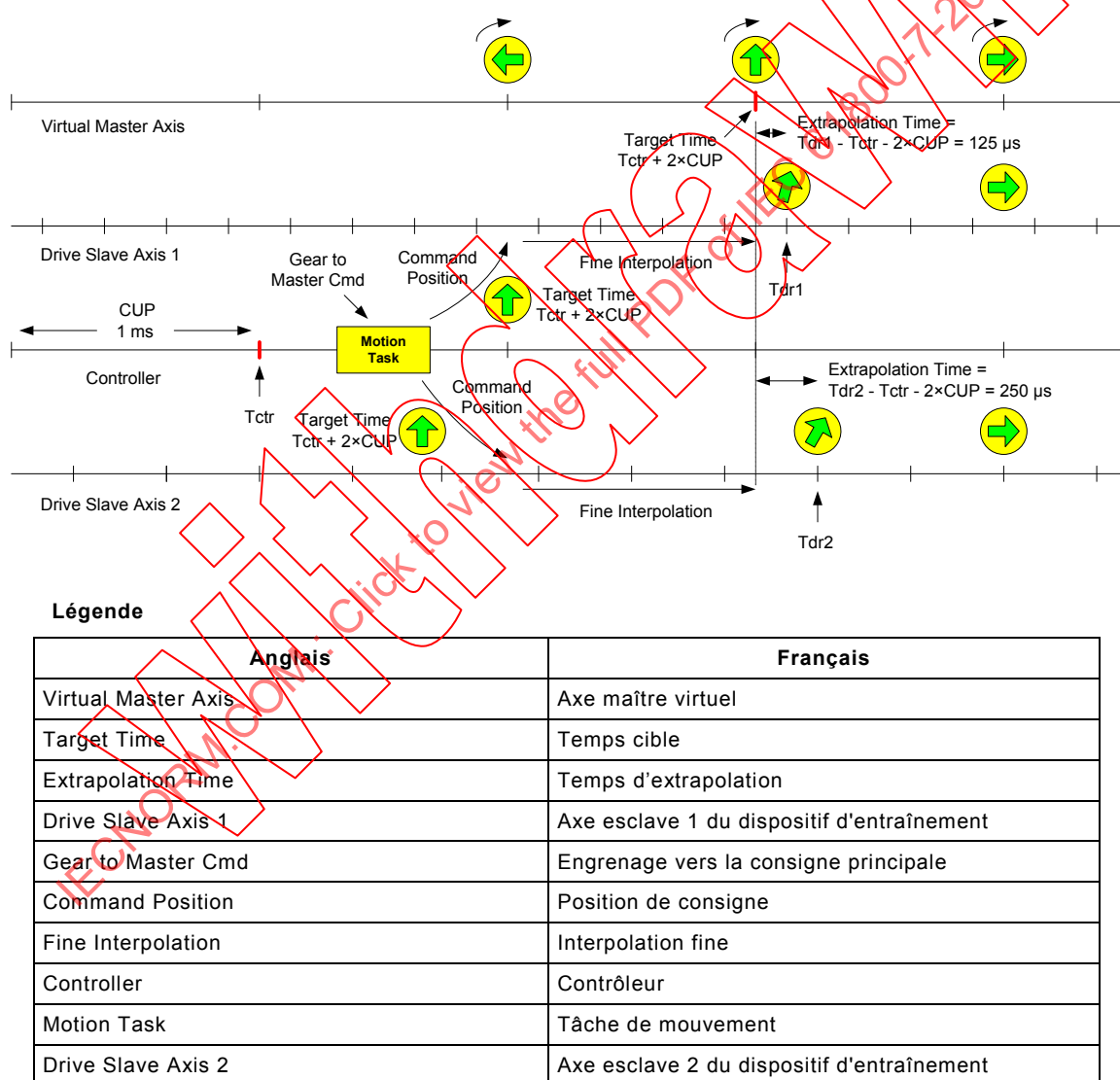


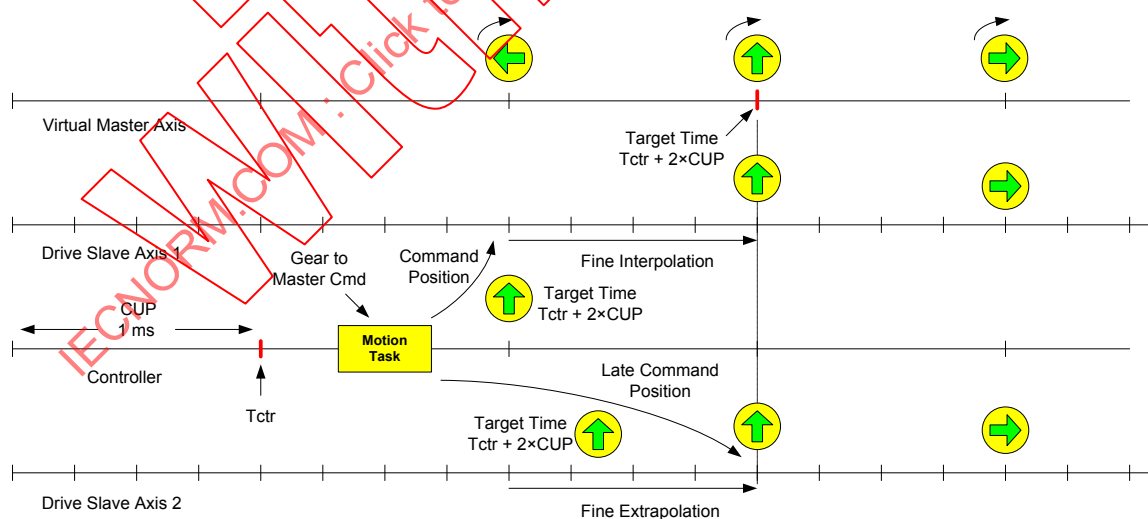
Figure 40 – Coordination des deux dispositifs d'entraînement avec différentes périodes d'actualisation

Dans le diagramme de temporisation ci-dessus, la tâche du planificateur de mouvement du contrôleur envoie des positions de consigne et des datations identiques aux deux axes d'entraînement esclaves qui, lorsqu'ils sont synchronisés avec le temps système, fonctionnent à différentes vitesses d'actualisation de dispositif d'entraînement. Lorsque les données de position de consigne arrivent au niveau des deux dispositifs d'entraînement, elles utilisent la

datation du contrôleur, le temps de consigne cible et la période d'actualisation du contrôleur pour calculer de nouveaux coefficients par rapport au polynôme d'interpolation, en s'appuyant sur la contrainte selon laquelle la valeur polynômiale égale en temps à (Datation du contrôleur + Temps de consigne cible x Période d'actualisation du contrôleur) est la valeur Position de consigne spécifiée. Compte tenu de l'absence de dépendance quant à la vitesse d'actualisation du dispositif d'entraînement, les coefficients polynômiaux calculés par chaque dispositif d'entraînement sont identiques. Aucune actualisation du dispositif d'entraînement ne coïncidant avec ce temps cible, les dispositifs d'entraînement utilisent le polynôme d'interpolation fine pour calculer la référence de position de consigne de chaque actualisation de dispositif d'entraînement tant qu'une nouvelle position de consigne n'a pas été reçue du contrôleur. Si une nouvelle position de consigne n'arrive pas bien après le temps cible, le dispositif d'entraînement continue à utiliser la même équation polynômiale pour "extrapoler" la position de consigne pour les actualisations de dispositif d'entraînement suivantes (voir Figure 40). Cette extrapolation se poursuit tant que de nouvelles données arrivent et que de nouveaux coefficients peuvent être calculés. De cette manière, qu'il s'agisse d'interpolation ou d'extrapolation, chaque axe esclave fonctionne de manière harmonieuse et les deux axes restent verrouillés en phase avec l'axe maître.

6.4.6.5 Indépendance de latence de transmission

La coordination précise de plusieurs axes de dispositif d'entraînement CIP Motion peut être maintenue même si les paquets de connexion Dispositif - Contrôleur impliquent des délais importants lors du passage dans le réseau CIP. Dans la Figure 41, le paquet pour l'axe de dispositif d'entraînement esclave 2 a subi un retard important lors de la transmission. En conséquence, la position de consigne de cet axe doit être extrapolée à partir du dernier polynôme d'interpolation fine. Cela permet à l'axe de se déplacer en douceur à travers une perturbation de latence de transmission. Lorsque les nouvelles données de consigne arrivent, la nouvelle valeur de consigne ne peut pas accepter la valeur extrapolée en raison d'une erreur d'extrapolation. Cette erreur peut perturber le profil de mouvement. L'amplitude de l'erreur d'extrapolation dépend des éléments dynamiques du profil de mouvement et de la vitesse d'actualisation du contrôleur. Dans la plupart des applications réelles, des latences de transmission durant plusieurs périodes d'actualisation peuvent se produire sans perturbation notable du profil de mouvement associé.



Légende

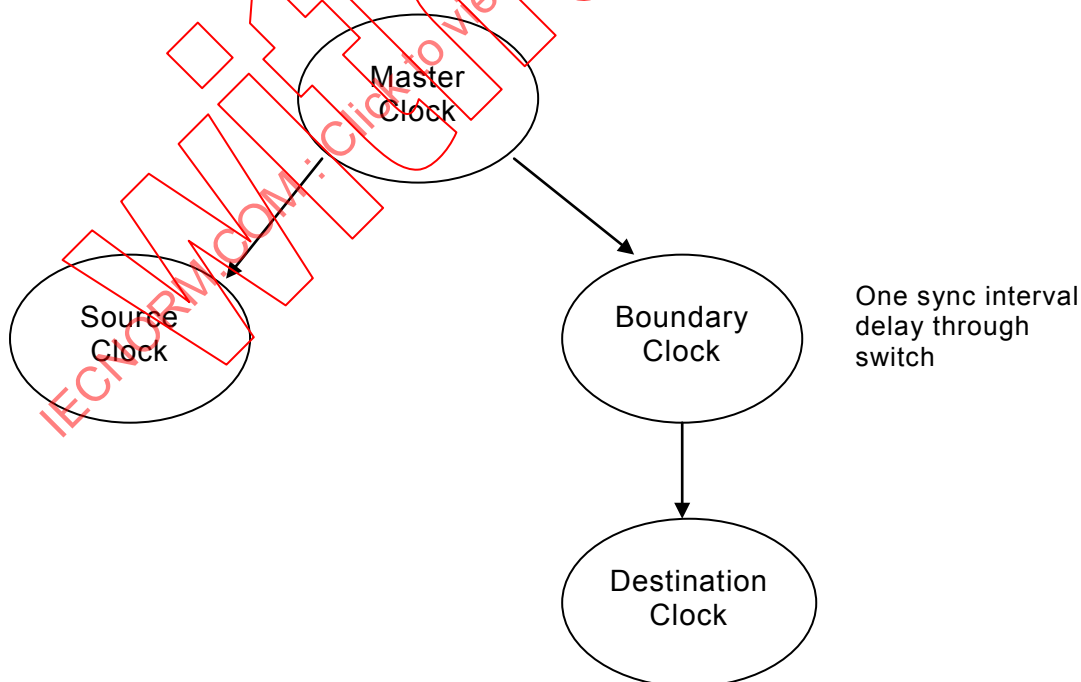
Anglais	Français
Virtual Master Axis	Axe maître virtuel
Target Time	Temps cible
Drive Slave Axis 1	Axe esclave 1 du dispositif d'entraînement
Gear to Master Cmd	Engrenage vers la consigne principale

Anglais	Français
Command Position	Position de consigne
Fine Interpolation	Interpolation fine
Controller	Contrôleur
Motion Task	Tâche de mouvement
Late Command Position	Position de consigne retardée
Drive Slave Axis 2	Axe esclave 2 du dispositif d'entraînement
Fine Extrapolation	Extrapolation fine

Figure 41 – Coordination de plusieurs axes de dispositif d'entraînement en cas de retard des paquets de connexion Contrôleur - Dispositif

6.4.6.6 Compensation de décalage de temps système

CIP Motion repose sur CIP Sync et son protocole de synchronisation CEI 61588:2004 sous-jacent, qui définit un mécanisme de distribution et de synchronisation du temps des dispositifs présents sur le réseau. Malheureusement, la CEI 61588:2004 ne garantit pas la vitesse de réponse de tous les dispositifs du système à une variation brusque de la datation du maître, et ne prévoit aucune disposition quant au traitement de cette condition. Ce changement de datation peut être provoqué, par exemple, par un utilisateur qui ajuste manuellement l'horloge maîtresse en raison d'une dérive correcte de l'horloge ou par modification de l'horloge maîtresse. Conformément à la CEI 61588:2004, les variations brusques de l'heure de l'horloge maîtresse doivent se propager dans le système temporel réparti, la durée de propagation de ces variations brusques dans le système entre l'horloge maîtresse et l'horloge esclave la plus éloignée étant fonction de l'intervalle de paquet de synchronisation (2 s par défaut) et du nombre de sauts (1 saut par horloge limite CEI 61588:2004, c'est-à-dire un interrupteur mettant en œuvre le protocole CEI 61588:2004, par exemple). La Figure 42 illustre cette démonstration.



Légende

Anglais	Français
Master Clock	Horloge maîtresse

Anglais	Français
Source Clock	Horloge source
Boundary clock	Horloge limite
Destination clock	Horloge de destination
One sync interval delay through switch	Un retard d'intervalle de synchronisation par l'interrupteur

Figure 42 – Propagation d'une variation brusque dans le temps

Dans la Figure 42, l'horloge source reçoit un ajustement du temps système sur un intervalle de synchronisation avant l'horloge de destination. Par conséquent, le temps système de l'horloge source est étalé par rapport à celui de l'horloge de destination pour un intervalle de paquet de synchronisation complet.

La variabilité de la propagation de cet ajustement chronologique dans le système crée un problème de commande de mouvement lié à la datation, en ce sens que la datation du dispositif source (le contrôleur, par exemple) peut ne pas s'appliquer lorsqu'il est reçu par le dispositif de destination (un dispositif d'entraînement, par exemple), car le temps système des dispositifs source et de destination est étalé. Cet effet est illustré en détail dans le Tableau 15.

Tableau 15 – Propagation d'une variation brusque dans le temps (exemple 1)

Intervalle de synchronisation temporelle	Horloge maîtresse	Temps source (contrôleur)	Temps de destination (dispositif d'entraînement)
1	100	100	100
2	200 + 1 000	200	200
3	1 300	300 + 1 000	300
4	1 400	1 400	400 + 1 000
5	1 500	1 500	1 500

Aux intervalles de synchronisation temporelle 1 et 2, l'horloge source de datation du contrôleur et l'horloge de destination de datation du dispositif d'entraînement présentent la même notion du temps (c'est-à-dire le temps système). A l'intervalle de synchronisation 2, une variation brusque du temps se produit au niveau de l'horloge maîtresse et se propage vers le contrôleur dans l'intervalle de synchronisation 3, affectant les datations de l'horloge source envoyées au dispositif d'entraînement. Ce n'est qu'à l'intervalle de synchronisation 4 que le dispositif d'entraînement détecte la variation brusque. Toutes les données de datation envoyées du contrôleur au dispositif d'entraînement entre les intervalles de synchronisation 3 et 4 ne sont pas corrélées avec la notion de temps du dispositif d'entraînement.

CIP Motion et CIP Sync étendent le protocole CEI 61588:2004 pour traiter cette condition. Pour ce faire, un algorithme d'exécution est défini dans le dispositif de destination de la datation, qui peut détecter cette condition de pas de temps et ajuster la valeur de la datation reçue de manière précise. Deux conditions sont possibles:

- 1) Le dispositif source a détecté une variation brusque dans le temps, à l'inverse du dispositif de destination;
- 2) Le dispositif de destination a détecté une variation brusque dans le temps, à l'inverse du dispositif source.

La composante essentielle de l'algorithme est la valeur Décalage de temps système définie comme faisant partie intégrante de l'Objet Synchronisation temporelle. Cette valeur de décalage est ajoutée à l'heure de l'horloge locale afin de générer le temps système, c'est-à-dire,

$$\text{Temps système} = \text{Horloge locale} + \text{Décalage de temps système}$$

Si le système est synchronisé, à un moment donné dans le temps, les valeurs Horloge locale et Décalage de temps système peuvent différer d'un dispositif à l'autre, mais il convient que la valeur Temps système soit identique dans la précision de la mise en œuvre du protocole CEI 61588:2004.

Une variation brusque dans le temps est indiquée par une modification de la valeur Décalage de temps système des dispositifs source ou de destination. Le décalage de temps système de la source est envoyé au dispositif de destination en tant que décalage, le temps système de la source étant envoyé en tant que datation de données. Le dispositif de destination compare le décalage à celui déjà reçu afin de déterminer si une variation brusque s'est produite, puis ajuste la valeur de datation en conséquence.

L'algorithme est le suivant:

$$\text{Timestamp}_{\text{comp}} = \text{Timestamp}_{\text{Rec}} + ((\text{Offset}_{\text{dest}} - \text{Offset}_{\text{dest}(\text{last})}) - (\text{Offset}_{\text{src}} - \text{Offset}_{\text{src}(\text{last})}))$$

If $|((\text{Offset}_{\text{dest}} - \text{Offset}_{\text{dest}(\text{last})}) - (\text{Offset}_{\text{src}} - \text{Offset}_{\text{src}(\text{last})}))| \leq \text{Sync Threshold}$

$$\text{Offset}_{\text{dest}(\text{last})} = \text{Offset}_{\text{dest}}$$

$$\text{Offset}_{\text{src}(\text{last})} = \text{Offset}_{\text{src}}$$

où:

TimestampRec = datation reçue.

Offsetdest = valeur en cours du décalage de temps système de la destination.

Offsetdest(last) = valeur précédente du décalage de temps système de la destination.

Offsetsrc = valeur reçue du décalage de temps système de la source.

Offsetsrc(last) = valeur précédente du décalage de temps système de la source.

Sync Threshold = nombre qui définit la tolérance de synchronisation de l'application du dispositif. Pour un dispositif d'entraînement vectoriel CIP Motion classique, il convient que ce nombre soit de l'ordre de 1 µs.

Le Tableau 16 donne un exemple de fonctionnement de la Compensation du décalage du temps système. Pour la clarté de l'illustration, un paquet est supposé être envoyé à l'intervalle de synchronisation sans retard de propagation.

Tableau 16 – Propagation d'une variation brusque dans le temps (exemple 2)

Intervalle de synchronisation temporelle	Horloge maîtresse	Source de la datation (contrôleur)					Destination de la datation (dispositif d'entraînement)				
		Horloge locale	Décalage	Dernier décalage	Heure locale	Datation	Horloge locale	Décalage	Dernier décalage	Heure locale	Datation ajusté
1	100	0	100	100	100	100	0	100	100	100	100
2	200 + 1 000	100	100	100	200	100	100	100	100	200	200
3	300	200	100 + 1 000	100	1 300	1 300	200	100	100	300	300
4	1 400	300	1 100	100	1 400	1 400	300	100 + 1 000	100	1 400	1 400
5	1 500	400	1 100	1 100	1 500	1 500	400	1 100	1 100	1 500	1 500

A l'intervalle de synchronisation 2, une variation brusque de 1 000 se produit dans l'horloge maîtresse, et est propagée vers l'horloge source de la datation (le contrôleur, par exemple)

dans l'intervalle de synchronisation 3. La datation reçue par l'horloge de destination (le dispositif d'entraînement, dans cet exemple) est compensée de 1 000 conformément à l'algorithme ci-dessus, donnant lieu à une valeur Datation de contrôleur ajustée cohérente avec l'heure locale du dispositif d'entraînement, même s'il n'a pas encore détecté la variation brusque. Dans l'intervalle de synchronisation 4, le dispositif d'entraînement détecte la variation brusque, mais ne procède à aucun ajustement de la datation car, dans la formule de compensation ci-dessus, les termes de décalage delta source et de destination s'annulent, et les valeurs Dernier décalage sont actualisées comme indiqué dans l'intervalle de synchronisation 5, plus aucune compensation de décalage n'étant obligatoire.

6.5 Procédure de démarrage du dispositif d'entraînement

6.5.1 Généralités

Ce paragraphe présente le processus de démarrage d'un dispositif d'entraînement CIP Motion. Pour des raisons de clarté, les messages exacts et leurs formats ne sont pas présentés. Il s'agit de présenter les étapes et les échanges mis en place entre le contrôleur et le dispositif d'entraînement afin de configurer correctement le dispositif d'entraînement CIP Motion.

Le démarrage du dispositif d'entraînement implique trois processus distincts initiés par le contrôleur CIP et auxquels tous les dispositifs d'entraînement présents sur le réseau répondent. Il s'agit de:

- 1) la création d'une connexion E-S Motion;
- 2) la configuration d'un objet de mouvement;
- 3) la synchronisation temporelle.

Ces processus peuvent se dérouler en parallèle ou consécutivement, selon la mise en œuvre particulière. La configuration du dispositif d'entraînement peut avoir lieu par l'intermédiaire d'une messagerie explicite connectée ou non connectée. Le processus de démarrage est présenté dans les paragraphes ci-dessous.

6.5.2 Création d'une connexion E-S Motion

Les connexions E-S Motion sont créées sur EtherNet/IP à l'aide des services CIP communs Forward_Open ou Large_Forward_Open (voir CEI 61158-5-2 et CEI 61158-6-2). Il convient que ces services spécifient la taille maximale des structures de données de connexion Contrôleur - Dispositif d'entraînement et Dispositif - Contrôleur appliquées à l'Objet Axe de mouvement. La configuration des chemins de connexion E-S générés et utilisés est décrite ci-dessus.

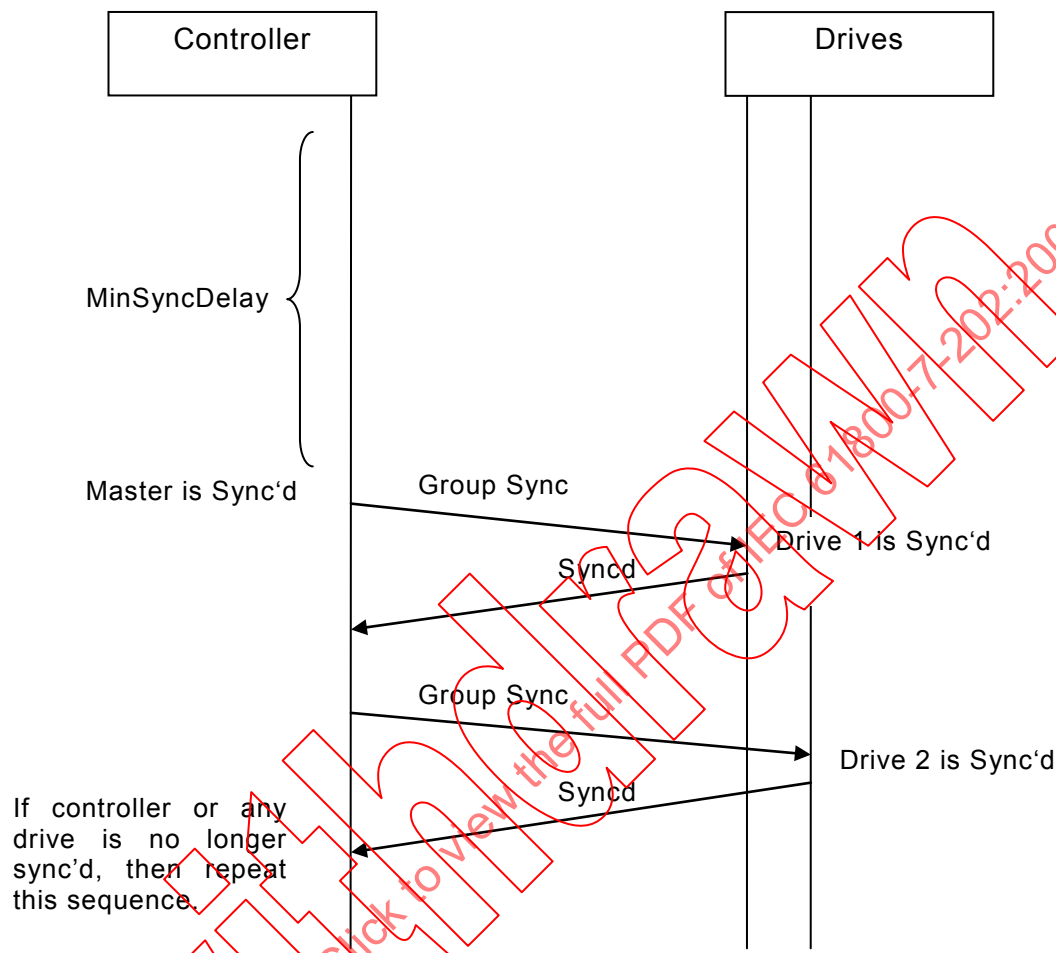
6.5.3 Configuration de l'Objet Axe de mouvement

La configuration de l'Objet Axe de mouvement est assurée à l'aide du service Set_Attribute_List commun ou du service Set_Axis_Attribute_List spécifique à l'objet par l'Objet Axe de mouvement et prise en charge par la connexion E-S CIP Motion.

6.5.4 Synchronisation temporelle

Tous les nœuds de dispositif d'entraînement CIP Motion qui prennent en charge la synchronisation temporelle doivent être synchronisés avec un contrôleur CIP Motion dans le cadre d'un groupe, appelé Groupe Motion. Un service Group_Sync, contenant la période d'actualisation et le décalage de temps système du contrôleur, est envoyé à tous les dispositifs de ce groupe. Cette procédure a pour objet de garantir la synchronisation des deux dispositifs ou, en d'autres termes, que leur notion du temps est identique. Pour ce faire, il s'agit de garantir qu'aucune variation brusque de la valeur Décalage de temps système du dispositif d'entraînement ne s'est produite dernièrement, et que les valeurs Décalage de temps système du contrôleur et du dispositif d'entraînement reposent sur une référence de temps système cohérente entre le contrôleur et le dispositif d'entraînement.

La Figure 43 illustre comment le contrôleur attend d'abord la synchronisation de son horloge locale à l'horloge Temps système du maître, puis attend une période minimale pour s'assurer qu'aucune variation brusque ne s'est produite par rapport au décalage de temps système. Les variations brusques sont des variations par rapport au décalage de temps système plus importantes qu'un seuil prédéfini par l'attribut Seuil de synchronisation.



Légende

Anglais	Français
Controller	Contrôleur
Drives	Dispositifs d'entraînement
Master is Sync'd	Le maître est synchronisé
Group Sync	Synchronisation de groupe
Sync'd	Synchronisé
Drive 1 is Sync'd	Le dispositif d'entraînement 1 est synchronisé
If controller or any drive is no longer sync'd, then repeat this sequence	Si le contrôleur ou un dispositif d'entraînement n'est plus synchronisé, répéter cette séquence.
Drive 2 is Sync'd	Le dispositif d'entraînement 2 est synchronisé

Figure 43 – Synchronisation de groupe de dispositifs d'entraînement CIP Motion

Si aucun décalage de temps système ne s'est produit pendant cette période, le contrôleur est considéré comme étant synchronisé, et lance une série de services de synchronisation de groupe vers chacun des dispositifs d'entraînement CIP Motion. Chaque dispositif

d'entraînement renvoie une réponse indiquant s'il est également synchronisé selon les mêmes critères que ceux du contrôleur. Si le contrôleur et tous les dispositifs d'entraînement indiquent qu'ils sont synchronisés, le Groupe Motion est considéré comme étant synchronisé.

Deux paramètres particuliers sont utilisés par un dispositif CIP Motion pour déterminer si le dispositif est synchronisé au groupe.

1. Sync Update Delay
2. Sync Threshold

Sync Update Delay est calculé par l'Objet Axe de mouvement comme étant la durée minimale d'attente après la dernière modification du décalage de temps système supérieure à Sync Threshold, avant que le dispositif ne soit considéré comme étant synchronisé au groupe. Cette valeur repose sur le produit du nombre de sauts de réseau qui sépare le dispositif d'entraînement de l'horloge grand-maître et l'intervalle de synchronisation temporelle. L'attribut Sync Update Delay est en général de l'ordre de quelques secondes.

Sync Threshold est un attribut de classe de l'Objet Axe de mouvement dont la valeur est utilisée pour déterminer si l'horloge locale du dispositif est synchronisée en fréquence avec l'horloge de temps système du maître. Il dépend de la mise en œuvre de la CEI 61588:2004, mais est en général de l'ordre de quelques microsecondes dans le cas d'une mise en œuvre d'horloge assistée par matériel. Les changements de temps de décalage supérieurs à Sync Threshold sont considérés comme étant des variations brusques par rapport au décalage de temps système. Les variations de temps de décalage inférieures à Sync Threshold sont considérées comme des corrections normales du temps système visant à compenser la dérive d'horloge et la variation de latence du réseau.

Lorsque le contrôleur et tous les dispositifs d'entraînement du Groupe Motion sont synchronisés au groupe, le contrôleur peut définir le bit Commande synchrone de l'en-tête Connexion Contrôleur - Dispositif du paquet cyclique suivant envoyé aux dispositifs d'entraînement, indiquant que le dispositif d'entraînement peut planifier son actualisation de connexion Dispositif - Contrôleur en fonction de la datation du contrôleur et de la période d'actualisation du contrôleur contenues dans les données de connexion. Le dispositif d'entraînement indique qu'il envoie à présent les données de connexion planifiées en fonction de la période d'actualisation du contrôleur en définissant le bit Mode synchrone de l'en-tête Connexion Dispositif - Contrôleur du paquet cyclique suivant envoyé au contrôleur. L'échange de données entre le contrôleur et le dispositif d'entraînement se poursuit donc conformément au modèle de temporisation CIP Motion.

6.6 Visualisation du dispositif

Le dispositif d'entraînement CIP Motion requiert des composants de visualisation permettant de diagnostiquer rapidement son état. Il s'agit de composants tels que des DEL bicolores et des affichages alphanumériques à plusieurs caractères. Le Tableau 17 définit les exigences relatives à ces composants de visualisation et leurs libellés appropriés lorsqu'ils sont utilisés par le dispositif d'entraînement. S'ils sont appliqués à des réseaux CIP différents, le comportement DEL d'état du réseau et d'état du module doit être conforme au comportement DEL du réseau CIP associé auquel le dispositif est connecté. La mise en correspondance des états de la DEL de statut du module du dispositif avec les états de l'Objet Axe de mouvement est définie de manière exhaustive dans la section Objet Axe de mouvement de Visualisation.

Tableau 17 – Composants de visualisation CIP Motion

Composant de visualisation	Besoin dans la mise en œuvre	Libellé associé
DEL d'état du réseau	Obligatoire	"État du réseau" "ER"
DEL de statut du module	Obligatoire	"État du module" "EM"
DEL de statut de l'axe	Conditionnel ^a	"État de l'axe" "XS" "Axe # État" "XS#" ^b
Affichage à sept segments	Facultatif	N/A

Composant de visualisation	Besoin dans la mise en œuvre	Libellé associé
Affichage alphanumérique à plusieurs caractères	Facultatif	N/A
^a Il est possible de déroger à cette exigence en utilisant un affichage alphanumérique à plusieurs caractères. ^b Des dispositifs d'entraînement multi-axe ont besoin de plusieurs DEL d'état de dispositif d'entraînement (une par axe de dispositif d'entraînement).		

6.7 Quality of Service (QoS) Ethernet

Lorsqu'un dispositif d'entraînement CIP Motion prenant en charge les services de synchronisation temporelle est ciblé pour fonctionner sur EtherNet/IP, le protocole Ethernet QoS (ou Quality of Service) standard doit être utilisé pour garantir que les paquets CEI 61588:2004 CIP Sync et les paquets de connexion E-S CIP Motion sont délivrés de manière opportune sur le réseau. De manière plus spécifique, le protocole QoS sur Ethernet standard utilise le protocole de paquet de trame marqué IEEE 802.1Q prenant en charge un champ QoS à trois bits. De plus, les paquets CIP Motion ont la priorité CIP urgente. Voir la CEI 61158-6-2 (Machine de protocole de mise en correspondance EtherNet/IP) pour la spécification de priorité des messages de commande de mouvement en temps réel.

7 Objet Axe de mouvement

7.1 Considérations générales

7.1.1 Généralités

Ce paragraphe définit le comportement d'un objet CIP appelé Objet Axe de mouvement. Cet objet est le principal composant fonctionnel des profils de dispositif CIP Motion définis dans le cadre des extensions CIP Motion vers CIP Common (le profil de dispositif d'entraînement CIP Motion, par exemple). Les instances de cet objet sont obligatoires pour prendre en charge la commande de mouvement en cas de connexion à un contrôleur conforme à CIP Motion par l'intermédiaire d'un réseau CIP. Les réseaux CIP applicables sont DeviceNet ControlNet et EtherNet/IP.

7.1.2 Présentation de l'objet

L'Objet Axe de mouvement est l'un des deux principaux composants d'objet nécessaires à la prise en charge de la commande de mouvement dans l'architecture de commande CIP Motion. Ces composants d'objet sont illustrés à la Figure 44.

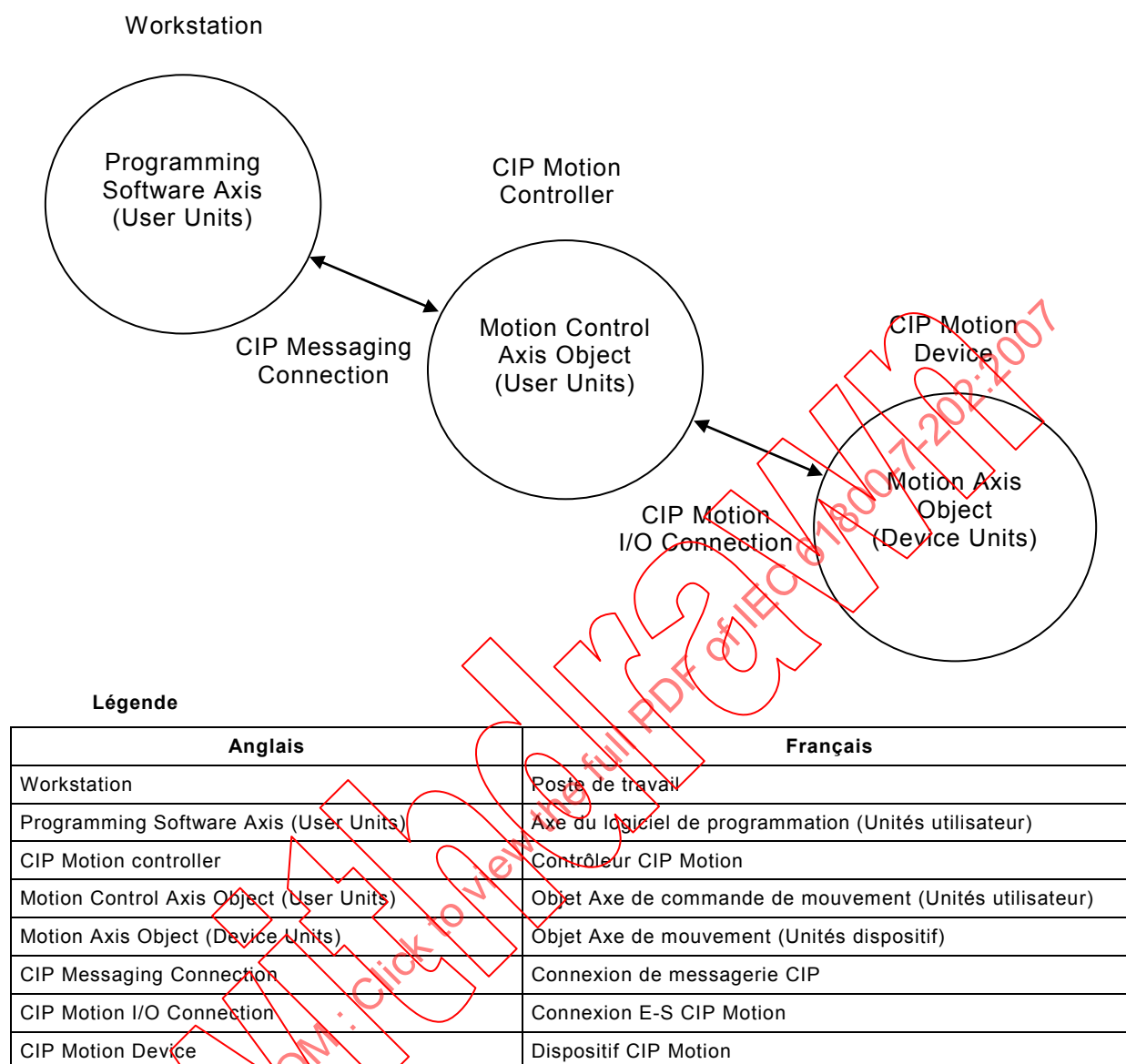


Figure 44 – Composants d'objet de l'architecture de mouvement CIP

7.1.3 Abstraction de l'Objet Axe de mouvement

L'Objet Axe de mouvement est une abstraction qui peut être appliquée à des composants mobiles associés à un système de commande de mouvement. Un composant mobile peut donc être représenté par une instance Objet Axe de mouvement, parfois simplement appelée "Axe". En règle générale, une instance Objet Axe de mouvement est composée d'un ou de plusieurs des éléments intégraux suivants:

- 1) Actionneur de mouvement (moteurs rotatifs ou linéaires, actionneurs hydrauliques et pneumatiques, par exemple);
- 2) Capteur de mouvement (codeurs, résolveurs, transducteurs ultrasoniques de débattement, par exemple);
- 3) Commande d'actionneur (commande de moteur vectorielle, à fréquence et pas à pas, commande de robinet hydraulique, par exemple);
- 4) L'amplificateur qui fournit la puissance à l'actionneur (structure de puissance d'excitation des moteurs, par exemple).

Les attributs d'instance sont inclus dans l'Objet Axe de mouvement pour chacun de ces éléments intégraux afin de décrire le comportement de l'axe. Si un Objet Axe de mouvement est doté d'un élément de commande d'actionneur, les attributs de commande sont inclus dans l'objet afin de contrôler les éléments dynamiques de l'actionneur. L'objet contient également des attributs qui décrivent les éléments dynamiques de l'actionneur, qu'ils soient estimés ou réels.

Le mode d'application le plus fréquent d'un Objet Axe de mouvement étant un dispositif "d'entraînement" tel que décrit par le profil de dispositif d'entraînement CIP Motion, l'Objet Axe de mouvement peut être appliqué à d'autres dispositifs de mouvement (un dispositif de réaction à une seule voie ou à plusieurs voies, par exemple). Dans le cas d'un dispositif de réaction à plusieurs voies, chaque voie est associée à un axe et, à ce titre, une instance de l'Objet Axe de mouvement lui est affectée. Dans le cas d'un dispositif d'entraînement à plusieurs amplificateurs, chaque amplificateur est considéré comme un axe, et une instance de l'Objet Axe de mouvement lui est affectée. Pour les instances d'axe d'entraînement, l'Objet Axe de mouvement peut représenter un entraînement asservi à commande vectorielle en boucle fermée complexe ou un simple dispositif d'entraînement Volts/Hertz à boucle ouverte (ou VFD). Les dispositifs d'entraînement sont souvent dotés de plusieurs voies de réaction qui peuvent être dédiées à des instances d'amplificateur de dispositif d'entraînement ou utilisées comme sources de réaction principales indépendantes. Dans ce dernier cas, la voie de réaction principale est considérée comme un axe, et une instance de l'Objet Axe de mouvement lui est affectée.

7.1.4 Objet Axe de commande de mouvement

A chaque instance Objet Axe de mouvement sur le dispositif dans le système de commande de mouvement correspond un Objet Axe de commande de mouvement sur le contrôleur. Les valeurs d'attribut contenues dans l'Objet Axe de mouvement comportent des attributs dans l'Objet Axe de commande de mouvement, la seule différence étant les unités dans lesquelles sont exprimés ces attributs. Dans l'Objet Axe de mouvement, les attributs sont exprimés en unités de dispositif fixes (nombres de réactions ou r/min, par exemple) telles que définies dans l'Objet Axe de mouvement. Dans l'Objet Axe de commande de mouvement, les attributs sont exprimés en unités définies par l'utilisateur (pouces ou mètres, par exemple). Étant donné que le logiciel de programmation interagit avec le dispositif par l'intermédiaire de l'Objet Axe de commande de mouvement, la programmation de la configuration, du diagnostic et du mouvement est assurée avec les unités préférentielles de l'utilisateur. Par conséquent, plutôt que de programmer l'application de commande de mouvement en nombres de réaction ou en r/min de l'arbre de moteur, par exemple, l'utilisateur peut programmer en centimètres, en pieds, en produits par minute ou en toute autre unité qui fait sens pour l'application spécifique.

Lors du transfert des valeurs d'attribut de configuration ou de commande vers l'Objet Axe de mouvement via la connexion E-S CIP Motion, ces valeurs d'attribut sont ajustées par l'Objet Axe de commande de mouvement dans les valeurs de l'unité de dispositif fixe requises par l'Objet Axe de mouvement. De la même manière, lorsque des attributs de signal sont transférés vers l'Objet Axe de commande de mouvement via la connexion E-S CIP Motion, ces valeurs d'unité de réaction sont ajustées par l'Objet Axe de commande de mouvement et stockées dans l'objet des unités utilisateur.

7.1.5 Classification de commande d'entraînement

En fonction des variations en Mode de commande et Méthode de commande figurant en 4.2, il est possible de déduire certains codes de commande de dispositif de base autour desquels la plupart des attributs de l'Objet Axe de mouvement peut être organisé. Les codes de commande de dispositif peuvent être désignés par un identifiant à une seule lettre, permettant de déterminer les attributs d'objet obligatoires pour la mise en œuvre d'un dispositif CIP Motion donné. La liste des codes de commande de dispositif est présentée ci-dessous:

Modes de commande:

P – Asservissement de position

V – Commande de vitesse

T – Asservissement de couple

N – Pas d'asservissement (Réaction uniquement)

Méthodes de commande:

F – Dispositifs d'entraînement à variation de fréquence (boucle ouverte V/Hz ou VFD)

C – Dispositifs d'entraînement vectoriels en boucle fermée

Les combinaisons de ces lettres peuvent être utilisées pour désigner une classe particulière de dispositifs CIP Motion afin d'identifier les attributs applicables. Par exemple, "F" indique un axe d'entraînement à fréquence variable, "V" un axe d'entraînement contrôlé par la vitesse (boucle ouverte ou fermée) et "N" un axe doté de la fonction de réaction uniquement.

7.1.6 Obligatoire par rapport à Facultatif dans une mise en œuvre

Dans les paragraphes qui suivent, les attributs et services sont définis comme étant obligatoires ou facultatifs dans la mise en œuvre de l'Objet Axe de mouvement. Les attributs et services obligatoires doivent être pris en charge dans la mise en œuvre de l'objet. Les attributs et services facultatifs peuvent être pris en charge ou non dans la mise en œuvre, et sont laissés à la discrétion du fournisseur.

La détermination du caractère obligatoire ou facultatif d'un attribut ou service donné dépend souvent du code de commande du dispositif défini ci-dessus. Si un attribut ou service est marqué comme étant obligatoire pour un code de commande de dispositif donné, la mise en œuvre d'un dispositif doit le prendre en charge s'il est censé fonctionner dans ce mode. Par exemple, un attribut marqué comme étant obligatoire pour le code de commande de dispositif "V" doit être pris en charge par un dispositif CIP Motion prenant en charge le mode de commande de Vitesse.

Dans certains cas, un attribut ou service peut même ne pas être applicable à un code de commande de dispositif. Cette situation se produit lorsque l'attribut est défini comme n'étant ni obligatoire ni facultatif.

Légende du Tableau 18: (voir 7.1.5 et 7.1.6 pour plus de détails).

(O) – Bit ou énumération obligatoire – doit être pris en charge dans la mise en œuvre.

(F) – Bit ou énumération facultatif – la prise en charge est laissée à la discrétion du fournisseur.

Set* – Indique que l'attribut est en principe défini par le bloc de données de connexion CIP Motion et pas par un service Set.

Tableau 18 – Mise en œuvre d'attribut d'instance par rapport au Code de commande de dispositif

Attribut d'instance			Mise en œuvre par code de commande de dispositif					
ID attribut	Règle d'accès	Nom de l'attribut	N	F	Ctrl vecteur C-PI			Mise en œuvre conditionnelle
					V	P	T	
80	Set*	Mode de commande	-	O	O	O	O	
81	Set	Méthode de commande	-	O	O	O	O	
1310	Get	Numéro de lot du moteur	-	F	F	F	F	
1311	Get	Numéro de série du moteur	-	F	F	F	F	
1312	Get	Code de date du moteur	-	F	F	F	F	
1313	Set	Source de données du moteur	-	O	O	O	O	
1314	Set	Code de dispositif du moteur	-	O	O	O	O	
1315	Set	Type de moteur	-	O	O	O	O	
1316	Set	Unité moteur	-	F	F	F	F	
1317	Set	Polarité du moteur	-	O	O	O	O	
1318	Set	Tension assignée du moteur	-	O	O	O	O	
1319	Set	Courant continu assigné du moteur	-	O	O	O	O	
1320	Set	Courant de crête assigné du moteur	-	C	C	C	C	O – PM; F - IM
1321	Set	Puissance de sortie assignée du moteur	-	C	C	C	C	O – PM; F - IM
1322	Set	Limite de surcharge du moteur	-	F	F	F	F	
1323	Set	Thermocontacteur interne du moteur	-	F	F	F	F	
1324	Set	Temp. d'enroulement max du moteur	-	F	F	F	F	
1325	Set	Cap. Therm. Enroul-Amb. du moteur	-	F	F	F	F	
1326	Set	Rés. Therm. Enroul-Amb. du moteur	-	F	F	F	F	
1327	Set	Résistance du moteur à aimant permanent	-	O	O	O	O	Moteur à aimant permanent uniquement
1328	Set	Inductance du moteur à aimant permanent	-	O	O	O	O	Moteur à aimant permanent uniquement
1329	Set	Pôles du moteur rotatif	-	O	O	O	O	Moteur rotatif uniquement
1330	Set	Inertie du moteur rotatif	-	F	F	F	F	Moteur rotatif uniquement
1331	Set	Vitesse assignée du moteur rotatif	-	O	O	O	O	Moteur rotatif uniquement
1332	Set	Vitesse max du moteur rotatif	-	F	F	F	F	Moteur rotatif uniquement
1333	Set	Coef. d'amortisseur du moteur rotatif	-	F	F	F	F	Moteur rotatif uniquement
1334	Set	Pas polaire du moteur linéaire	-	O	O	O	O	Moteur linéaire uniquement
1335	Set	Vitesse assignée du moteur linéaire	-	O	O	O	O	Moteur linéaire uniquement
1336	Set	Masse en mouvement du moteur linéaire	-	F	F	F	F	Moteur linéaire uniquement
1337	Set	Vitesse max du moteur linéaire	-	F	F	F	F	Moteur linéaire uniquement
1338	Set	Coef. d'amortisseur du moteur linéaire	-	F	F	F	F	Moteur linéaire uniquement
1339	Set	Couple assigné du moteur à aimant permanent	-	O	O	O	O	Moteur à aimant permanent rotatif uniquement
1340	Set	Constante de couple du moteur à aimant permanent	-	O	O	O	O	Moteur à aimant permanent rotatif uniquement
1341	Set	Constante de tension du moteur à aimant permanent rotatif	-	O	O	O	O	Moteur à aimant permanent rotatif uniquement
1342	Set	Force assignée du moteur à aimant permanent	-	O	O	O	O	Moteur à aimant permanent linéaire uniquement
1343	Set	Const. de force du moteur à aimant permanent	-	O	O	O	O	Moteur à aimant permanent linéaire uniquement

Attribut d'instance			Mise en œuvre par code de commande de dispositif					
ID attribut	Règle d'accès	Nom de l'attribut	N	F	Ctrl vecteur C-PI			Mise en œuvre conditionnelle
				V	P	V	T	
1344	Set	Constante de tension du moteur à aimant permanent linéaire	-	O	O	O	O	Moteur à aimant permanent linéaire uniquement
1345	Set	Fréquence assignée du moteur à induction	-	O	O	O	O	Moteur à induction uniquement
1346	Set	Courant de flux du moteur à induction	-	O	O	O	O	Moteur à induction uniquement
1347	Set	Résistance statorique du moteur à induction	-	O	O	O	O	Moteur à induction uniquement
1348	Set	Fuite statorique X du moteur à induction	-	O	O	O	O	Moteur à induction uniquement
1349	Set	Aimantation X du moteur à induction	-	O	O	O	O	Moteur à induction uniquement
1350	Set	Résistance rotorique du moteur à induction	-	O	O	O	O	Moteur à induction uniquement
1351	Set	Fuite rotorique X du moteur à induction	-	O	O	O	O	Moteur à induction uniquement
1400 + o	Get	Référence n de réaction	-	F	F	F	F	
1401 + o	Get	Numéro de série n de réaction	-	F	F	F	F	
1402 + o	Get	Position n de réaction	O	-	O	O	F	
1403 + o	Get	Vitesse n de réaction	O	-	O	O	F	
1404 + o	Get	Accélération n de réaction	O	-	O	O	F	
90	Set*	Ensemble de données de réaction	O	O	O	O	O	
82	Set*	Configuration de réaction	O	O	O	O	O	
83	Set*	Sélection de réaction principale	O	-	O	O	O	
84	Set	Rapport de 1/2 nombre de réactions	-	-	F	-	-	
1411 + o	Set	Unité n de réaction	F	-	F	F	F	
1412 + o	Set	Sélection de port n de réaction	F	-	F	F	F	
1413 + o	Set	Type n de réaction	O	-	O	O	O	
1414 + o	Set	Polarité n de réaction	F	-	F	F	F	
1415 + o	Set	Mode n de réaction	O	-	O	O	O	
1416 + o	Set	Résolution de cycle n de réaction	O	-	O	O	O	TT, SC, HI, E21, RS
1417 + o	Set	Interpolation de cycle n de réaction	O	-	O	O	O	TT, SC, HI, E21, RS
1420 + o	Set	Longueur de données n de réaction	O	-	O	O	O	TP, SS, HI, E21, E22
1421 + o	Set	Code de données n de réaction	F	-	F	F	F	TP, SS
1422 + o	Set	Rapport trans. du résolveur n de réaction	F	-	F	F	F	RS
1423 + o	Set	Tension d'excitation du résolveur n de réaction	F	-	F	F	F	RS
1424 + o	Set	Fréquence d'excitation du résolveur n de réaction	F	-	F	F	F	RS
1425 + o	Set	Équilibre du câble du résolveur n de réaction	F	-	F	F	F	RS
1426 + o	Set	Type LDT n de réaction	O	-	O	O	O	LT
1427 + o	Set	Recirculations LDT n de réaction	O	-	O	O	O	LT
1428 + o	Set	Décalage de commutation n de réaction	O	-	O	O	O	HS, TP, HI, E21, E22, SS, RS (moteur à aimant permanent uniquement)
1434 + o	Set	Bande passante de filtre de vitesse n de réaction	F	-	F	F	F	
1435 + o	Set	Bande passante de filtre d'accélération n de réaction	F	-	F	F	F	
256	Set*	Commande de vérification d'événement	F	-	F	-	-	
257	Get	Statut de vérification d'événement	F	-	F	-	-	
258	Get	Position du front positif 1 de l'enregistrement	O	-	O	-	-	
259	Get	Position du front négatif 1 de l'enregistrement	O	-	O	-	-	
260	Get	Position du front positif 2 de l'enregistrement	F	-	F	-	-	
261	Get	Position du front négatif 2 de	F	-	F	-	-	

Attribut d'instance			Mise en œuvre par code de commande de dispositif					
ID attribut	Règle d'accès	Nom de l'attribut	N	F	Ctrl vecteur C-PI			Mise en œuvre conditionnelle
				V	P	V	T	
		l'enregistrement						
262	Get	Durée du front positif 1 de l'enregistrement	F	-	F	-	-	
263	Get	Durée du front négatif 1 de l'enregistrement	O	-	O	-	-	
264	Get	Durée du front positif 2 de l'enregistrement	O	-	O	-	-	
265	Get	Durée du front négatif 2 de l'enregistrement	F	-	F	-	-	
266	Get	Position d'événement d'origine	O	-	O	-	-	
268	Get	Heure d'événement d'origine	F	-	F	-	-	
93	Get	Temps de consigne cible	-	-	F	-	-	
286	Set*	Consigne de position du contrôleur	-	-	O	-	-	
287	Set*	Consigne de vitesse du contrôleur	-	-	F	O	-	
288	Set*	Consigne d'accélération du contrôleur	-	-	F	F	F	
289	Set*	Consigne de couple du contrôleur	-	-	F	F	F	
290	Get	Position de consigne interpolée	-	-	F	-	-	
291	Get	Vitesse de consigne interpolée	-	-	F	F	-	
292	Get	Accélération de consigne interpolée	-	-	F	F	F	
91	Set*	Ensemble de données de consigne	-	-	O	O	O	
92	Set*	Commande d'interpolation	-	-	O	O	-	
302	Set	Limite de vitesse - Positive	-	-	F	O	-	
303	Set	Limite de vitesse - Négative	-	-	F	O	-	
304	Set	Limite d'accélération	-	-	F	O	-	
305	Set	Limite de décélération	-	-	F	O	-	
306	Set	Commande de limite d'à-coups	-	-	F	F	-	
307	Set	Activation de l'amorçage instantané	-	-	F	F	-	
308	Set	Vitesse de saut 1	-	F	-	-	-	
309	Set	Vitesse de saut 2	-	F	-	-	-	
310	Set	Vitesse de saut 3	-	F	-	-	-	
311	Set	Bande de vitesse de saut	-	F	-	-	-	
420	Get	Consigne de position	-	-	O	-	-	
421	Set*	Compensation de position	-	-	O	-	-	
422	Get	Référence de position	-	-	O	-	-	
423	Get	Consigne d'action anticipatrice de vitesse	-	-	O	-	-	
424	Get	Retour en position	O	-	O	O	F	
426	Get	Erreur de position	-	-	O	-	-	
427	Get	Sortie de l'intégrateur de position	-	-	O	-	-	
428	Get	Sortie de la boucle de position	-	-	O	-	-	
430	Set	Kvff	-	-	O	-	-	
431	Set	Kpp	-	-	O	-	-	
432	Set	Kpi	-	-	O	-	-	
433	Set	Tolérance de verrouillage de position	-	-	O	-	-	
434	Set	Tolérance d'erreur de position	-	-	O	-	-	
435	Set	Temps de tolérance d'erreur de position	-	-	F	-	-	
439	Set	Commande d'intégrateur de position	-	-	O	-	-	
440	Set	Préchargement d'intégrateur de position	-	-	O	-	-	
450	Get	Commande de consigne	-	O	O	O	-	
451	Set*	Compensation de vitesse	-	O	O	O	-	
452	Get	Consigne d'anticipation d'accélération	-	-	O	O	-	
453	Get	Référence de vitesse	-	O	O	O	-	
454	Get	Retour en vitesse	O	-	O	O	F	
455	Get	Erreur de vitesse	-	-	O	O	-	
456	Get	Sortie d'intégrateur de vitesse	-	-	O	O	-	
457	Get	Sortie de boucle de vitesse	-	-	O	O	-	
460	Set	Kaff	-	-	O	-	-	
461	Set	Kvp	-	-	O	O	-	
462	Set	Kvi	-	-	O	O	-	
464	Set	Kdr	-	O	F	O	-	
465	Set	Tolérance d'erreur de vitesse	-	-	F	F	-	

Attribut d'instance			Mise en œuvre par code de commande de dispositif					
ID attribut	Règle d'accès	Nom de l'attribut	N	F	Ctrl vecteur C-PI			Mise en œuvre conditionnelle
				V	P	V	T	
466	Set	Temps de tolérance d'erreur de vitesse	-	-	F	F	-	
467	Set	Commande d'intégrateur de vitesse	-	-	O	O	-	
468	Set	Préchargement d'intégrateur de vitesse	-	-	O	O	-	
469	Set	Bande passante du filtre passe-bas de vitesse.	-	-	F	F	-	
470	Set	Seuil de vitesse	-	F	O	O	-	
471	Set	Tolérance de verrouillage de vitesse	-	F	O	O	-	
472	Set	Fenêtre de gel de vitesse	-	O	O	O	-	
480	Get	Consigne d'accélération	-	-	F	F	F	
481	Set*	Compensation d'accélération	-	-	F	F	F	
482	Get	Référence d'accélération	-	-	F	F	F	
483	Get	Retour en accélération	O	-	O	O	F	
490	Get	Consigne de couple	-	-	O	O	O	
491	Set*	Compensation de couple	-	-	O	O	O	
492	Get	Référence de couple	-	-	O	O	O	
493	Get	Référence de couple filtrée	-	-	O	O	O	
494	Get	Référence de couple limitée	-	-	O	O	O	
496	Set	Kj	-	-	O	O	F	
498	Set	Compensation de frottement	-	-	F	F	F	
502	Set	Bande passante de filtre passe-bas de couple	-	-	F	F	F	
503	Set	Fréquence de rejet de consigne de couple	-	-	F	F	F	
504	Set	Limite de couple - Positive	-	-	O	O	O	
505	Set	Limite de couple - Négative	-	-	O	O	O	
506	Set	Limite de débit de couple	-	-	F	F	F	
507	Set	Seuil de couple	-	-	F	F	F	
508	Set	Limite de sur-couple	-	-	F	F	F	
509	Set	Temps de limite de sur-couple	-	-	F	F	F	
510	Set	Limite de sous-couple	-	-	F	F	F	
511	Set	Temps de limite de sous-couple	-	-	F	F	F	
520	Get	Consigne de courant Iq	-	-	O	O	O	
521	Get	Limite de courant opératoire	-	-	F	F	F	
522	Get	Source de limite de courant	-	-	F	F	F	
523	Get	Angle électrique du moteur	-	-	O	O	O	
524	Get	Référence de courant Iq	-	-	F	F	F	
525	Get	Référence de courant Id	-	-	F	F	F	
527	Get	Erreur de courant Iq	-	-	F	F	F	
528	Get	Erreur de courant Id	-	-	F	F	F	
529	Get	Retour en courant Iq	-	-	F	F	F	
530	Get	Retour en courant Id	-	-	F	F	F	
531	Get	Découplage Iq	-	-	F	F	F	
532	Get	Découplage Id	-	-	F	F	F	
533	Get	Sortie de tension Vq	-	-	F	F	F	
534	Get	Sortie de tension Vd	-	-	F	F	F	
535	Get	Sortie de tension U	-	-	F	F	F	
536	Get	Sortie de tension V	-	-	F	F	F	
537	Get	Sortie de tension W	-	-	F	F	F	
538	Get	Retour en courant U	-	-	F	F	F	
539	Get	Retour en courant V	-	-	F	F	F	
540	Get	Retour en courant W	-	-	F	F	F	
541	Get	Décalage de courant U	-	-	F	F	F	
542	Get	Décalage de courant V	-	-	F	F	F	
543	Get	Décalage de courant W	-	-	F	F	F	
554	Get	Kqp	-	-	F	F	F	
555	Get	Kqi	-	-	F	F	F	
556	Get	Kdp	-	-	F	F	F	
557	Get	Kdi	-	-	F	F	F	
558	Set	Commande de flux ascendant	-	-	F	F	F	Moteur à induction uniquement

Attribut d'instance			Mise en œuvre par code de commande de dispositif					
ID attribut	Règle d'accès	Nom de l'attribut	N	F	Ctrl vecteur C-PI			Mise en œuvre conditionnelle
				V	P	V	T	
559	Set	Temps de flux ascendant	-	-	F	F	F	Moteur à induction uniquement
570	Get	Compensation de glissement	-	O	-	-	-	
575	Set	Méthode de commande de fréquence	-	O	-	-	-	
571	Set	Vitesse de glissement assignée	-	O	-	-	-	
572	Set	Tension maximale	-	O	-	-	-	Moteurs à aimant permanent uniquement
573	Set	Fréquence maximale	-	O	-	-	-	
574	Set	Limite de fréquence	-	O	-	-	-	
575	Set	Tension de rupture	-	O	-	-	-	
576	Set	Fréquence de rupture	-	O	-	-	-	
577	Set	Suralimentation au démarrage	-	O	-	-	-	
578	Set	Suralimentation à l'exécution	-	O	-	-	-	
600	Get	Fréquence de sortie	-	O	-	-	-	
601	Get	Courant de sortie	-	O	O	O	O	
602	Get	Tension de sortie	-	O	O	O	O	
603	Get	Puissance de sortie	-	O	O	O	O	
604	Set	Fréquence MID	-	F	F	F	F	
610	Set	Mode d'arrêt	-	O	O	O	O	
611	Set	Courant d'arrêt	-	-	O	O	O	
612	Set	Limite de temps d'arrêt	-	-	O	O	O	
613	Set	Délai de contact de frein résistif	-	F	F	F	F	
614	Set	Commande de frein mécanique	-	F	F	F	F	
615	Set	Délai de libération de frein mécanique	-	O	O	O	O	
616	Set	Délai d'engagement de frein mécanique	-	O	O	O	O	
620	Get	Tension de bus c.c.	-	O	O	O	O	
621	Get	Tension de bus c.c. - Nominale	-	O	O	O	O	
624	Set	Action du régulateur de bus	-	O	O	O	O	
625	Set	Limite de puissance du régulateur de bus	-	F	F	F	F	
627	Set	Action de perte de puissance	-	F	F	F	F	
628	Set	Seuil de perte de puissance	-	F	F	F	F	
629	Set	Action d'interruption	-	F	F	F	F	
635	Get	Capacité du moteur	-	O	O	O	O	
636	Get	Capacité de l'onduleur	-	O	O	O	O	
637	Get	Capacité du convertisseur	-	F	F	F	F	
638	Get	Capacité du régulateur de bus	-	F	F	F	F	
639	Get	Température ambiante	-	F	F	F	F	
640	Get	Température du dissipateur thermique de l'onduleur	-	F	F	F	F	
641	Get	Température de l'onduleur	-	F	F	F	F	
642	Get	Température du moteur	-	F	F	F	F	
643	Get	Température de réaction 1	-	F	F	F	F	
644	Get	Température de réaction 2	-	F	F	F	F	
645	Get	Limite de surcharge de l'onduleur	-	F	F	F	F	
646	Set	Action de surcharge du moteur	-	F	F	F	F	
647	Set	Action de surcharge de l'onduleur	-	F	F	F	F	
650	Get	État de l'axe	O	O	O	O	O	
651	Get	Statut de l'axe	O	O	O	O	O	
652	Get	Statut de l'axe - Mfg	O	O	O	O	O	
653	Get	Statut d'E-S de l'axe	O	O	O	O	O	
654	Get	Statut d'E-S de l'axe - Mfg	O	O	O	O	O	
94	Set*	Ensemble de données d'état	O	O	O	O	O	
655	Get	Statut d'exception de l'axe	O	O	O	O	O	
656	Get	Statut d'exception de l'axe - Mfg	O	O	O	O	O	
657	Get	Statut de défaut d'axe	O	O	O	O	O	
658	Get	Statut de défaut d'axe - Mfg	O	O	O	O	O	
659	Get	Statut d'alarme d'axe	O	O	O	O	O	
660	Get	Statut d'alarme d'axe - Mfg	O	O	O	O	O	
661	Get	Code de défaut d'axe	O	O	O	O	O	

Attribut d'instance			Mise en œuvre par code de commande de dispositif					
ID attribut	Règle d'accès	Nom de l'attribut	N	F	Ctrl vecteur C-PI			Mise en œuvre conditionnelle
				V	P	V	T	
662	Get	Code de défaut d'axe - Mfg	O	O	O	O	O	
663	Get	Datation de défaut d'axe	O	O	O	O	O	
664	Get	Code d'alarme d'axe	O	O	O	O	O	
665	Get	Code d'alarme d'axe - Mfg	O	O	O	O	O	
666	Get	Sous-code de défaut	F	F	F	F	F	
680	Get	Tolérance de fabrication de surchauffe du moteur	-	F	F	F	F	
681	Get	Tolérance de fabrication de surcharge de T. du moteur	-	F	F	F	F	
682	Get	Tolérance de fabrication de surchauffe de l'onduleur	-	F	F	F	F	
683	Get	Tolérance de fabrication de surcharge de T. de l'onduleur	-	F	F	F	F	
684	Get	Tolérance de fabrication de surchauffe du convertisseur	-	F	F	F	F	
685	Get	Tolérance de fabrication de surcharge de T. du convertisseur	-	F	F	F	F	
686	Get	Tolérance de fabrication de surchauffe du régulateur de bus	-	F	F	F	F	
687	Get	Tolérance de fabrication de surcharge de T. du régulateur de bus	-	F	F	F	F	
688	Get	Tolérance de fabrication de surtension du bus	-	F	F	F	F	
689	Get	Tolérance de fabrication de sous-tension du bus	-	F	F	F	F	
695	Set	Limite utilisateur de survitesse du moteur	-	F	F	F	F	
696	Set	Limite utilisateur de surchauffe du moteur	-	F	F	F	F	
697	Set	Limite utilisateur de surcharge de T. du moteur	-	F	F	F	F	
698	Set	Limite utilisateur de surchauffe de l'onduleur	-	F	F	F	F	
699	Set	Limite utilisateur de surcharge de T. de l'onduleur	-	F	F	F	F	
700	Set	Limite utilisateur de surchauffe du convertisseur	-	F	F	F	F	
701	Set	Limite utilisateur de surcharge de T. du convertisseur	-	F	F	F	F	
702	Set	Limite utilisateur de surchauffe du régulateur de bus	-	F	F	F	F	
703	Set	Limite utilisateur de surcharge de T. du régulateur de bus	-	F	F	F	F	
704	Get	Limite utilisateur de surtension du bus	-	F	F	F	F	
705	Get	Limite utilisateur de sous-tension du bus	-	F	F	F	F	
667	Set	Action d'exception	O	O	O	O	O	
668	Set	Action d'exception - Mfg	O	O	O	O	O	
669	Get	Statut de défaut d'initialisation	O	O	O	O	O	
670	Get	Statut de défaut d'initialisation - Mfg	O	O	O	O	O	
671	Get	Code de défaut d'initialisation	O	O	O	O	O	
672	Get	Code de défaut d'initialisation - Mfg	O	O	O	O	O	
673	Get	Statut d'inhibition de démarrage	-	O	O	O	O	
674	Get	Statut d'inhibition de démarrage - Mfg	-	O	O	O	O	
675	Get	Code d'inhibition de démarrage	-	O	O	O	O	
676	Get	Code d'inhibition de démarrage Mfg	-	O	O	O	O	
710	Get	Durée de mise sous tension de commande	-	F	F	F	F	
711	Get	Temps d'exécution cumulé	-	F	F	F	F	
712	Get	Utilisation d'énergie cumulée	-	F	F	F	F	
713	Get	Révs de moteur cumulé	-	F	F	F	F	
714	Get	Cycles de puissance principale cumulés	-	F	F	F	F	
715	Get	Cycles de puissance de commande cumulés	-	F	F	F	F	

Attribut d'instance			Mise en œuvre par code de commande de dispositif					
ID attribut	Règle d'accès	Nom de l'attribut	N	F	Ctrl vecteur C-PI			Mise en œuvre conditionnelle
				V	P	V	T	
720	Get	Tension de sortie assignée de l'onduleur	-	O	O	O	O	
721	Get	Courant de sortie assigné de l'onduleur	-	O	O	O	O	
722	Get	Puissance de sortie assignée de l'onduleur	-	O	O	O	O	
723	Get	Courant de sortie assigné du convertisseur	-	F	F	F	F	
724	Get	Puissance de sortie assignée du convertisseur	-	F	F	F	F	
730	Get	Entrées numériques	-	F	F	F	F	
731	Set	Sorties numériques	-	F	F	F	F	
732	Get	Entrée analogique 1	-	F	F	F	F	
733	Get	Entrée analogique 2	-	F	F	F	F	
734	Set	Sortie analogique 1	-	F	F	F	F	
735	Set	Sortie analogique 2	-	F	F	F	F	
750	Set	Commande locale	F	F	F	F	F	
Décalage d'ID d'attribut, o = (n-1)x50								

Décalage d'ID d'attribut, o = (n-1)x50

7.2 Attributs de classe

7.2.1 Généralités

Le tableau des attributs du Tableau 19 s'applique à la classe Objet Axe de mouvement, qui peut être référencée en instance 0. Ces attributs existent avant la création des instances Objet Axe de mouvement. Étant donné que les attributs de classe ne sont pas liés à une instance d'axe particulière du dispositif CIP Motion, ils sont en général utilisés pour adresser le comportement paramétrique qui s'applique à toutes les instances d'axe (le comportement du nœud de communication, par exemple). Au fur et à mesure de leur création, des numéros d'instance consécutifs sont donnés aux instances de cette classe, en commençant par 1.

NOTE 1 Aucun attribut associé à cet objet ne requiert un stockage non volatile. La valeur "No" est donc implicite pour tous les attributs de la colonne NV.

NOTE 2 Les bits spécifiques au fournisseur et les énumérations permettent aux fournisseurs de dispositif d'offrir des fonctions de produit supplémentaires.

Légende du Tableau 19: (voir 7.1.5 et 7.1.6 pour plus de détails).

(O) – Bit ou énumération obligatoire – doit être pris en charge dans la mise en œuvre.

(F) – Bit ou énumération facultatif – la prise en charge est laissée à la discrétion du fournisseur.

Set* – Indique que l'attribut est en principe défini par le bloc de données de connexion CIP Motion et pas par un service Set.

Tableau 19 – Attributs de classe de l'Objet Axe de mouvement

ID attr	Besoin dans la mise en œuvre	Règle d'accès	NV	Nom de l'attribut	Type de données	Description d'attribut	Sémantique des valeurs
1 à 7	Ces attributs de classe sont facultatifs et sont décrits dans la CEI 61158-5-2 et dans la CEI 61158-6-2.						
10	Obligatoire	Set*		Données de connexion utilisées du contrôleur	Struct	Cet attribut contient la structure des données utilisées définie par la connexion C-D CIP Motion.	Voir la sémantique en 7.2.2.1.
11	Obligatoire	Set*		Données de connexion générées du contrôleur	Struct	Cet attribut contient la structure des données générées définie par la connexion D-C CIP Motion.	Voir la sémantique en 7.2.2.2.
14	Obligatoire	Set*		Contrôle de nœud	OCTET	Contient les bits utilisés pour contrôler le comportement du nœud de communication du dispositif associé.	Voir la sémantique en 7.2.2.3.
15	Obligatoire	Get		Statut de nœud	OCTET	Contient les bits utilisés pour indiquer le statut du nœud de communication du dispositif associé.	Voir la sémantique en 7.2.2.4.
16	Obligatoire	Get		Défauts de nœud	USINT	Contient le code numérique de la condition de défaut de nœud. Le code de défaut de nœud est utilisé pour donner des détails de diagnostic et identifier la position de la source de la condition de défaut.	Voir la sémantique en 7.2.2.5.
17	Facultatif	Get		Alarme de nœud	USINT	Contient le code numérique de la condition d'alarme active en cours. Le code d'alarme de nœud est utilisé pour donner des détails de diagnostic et identifier la position de la source de la condition d'alarme.	Voir la sémantique en 7.2.2.6.
18	Obligatoire	Set*		Période d'actualisation du contrôleur	UDINT	Représente la période entre les actualisations du contrôleur, qui est également la période entre les actualisations de connexion C-D CIP Motion.	Nanosecondes
19	Obligatoire	Set*		Décalage temporel du contrôleur	LINT	Cet élément représente la valeur de 64 bits au début de la période d'actualisation du contrôleur associé à la valeur Datation du contrôleur. Le décalage temporel du contrôleur est la valeur ajoutée à l'horloge locale du contrôleur pour générer le temps système.	Nanosecondes

ID attr	Besoin dans la mise en œuvre	Règle d'accès	NV	Nom de l'attribut	Type de données	Description d'attribut	Sémantique des valeurs
20	Obligatoire	Set*		Datation du contrôleur	LINT	Cet élément représente la valeur Temps système de 64 bits au début de la période d'actualisation du contrôleur lorsqu'un événement de temporisation de l'actualisation du contrôleur se produit. Calculé par le contrôleur, il s'agit de la somme de la valeur d'horloge locale du contrôleur en cas d'événement de temporisation d'actualisation et de la valeur Décalage du temps système du contrôleur donnée par le décalage de temps système. La datation du contrôleur est donc directement liée aux données de consigne contenues dans la connexion. Le format de datation est absolu et suit la norme CIP Sync, 0 correspondant au 1 ^{er} janvier 1970.	Nanosecondes (CIP Sync absolu)
21	Facultatif	Set		Limite supérieure du délai d'actualisation du contrôleur	USINT	Représente la limite de délai supérieure d'une actualisation de connexion Contrôleur - Dispositif (C-D). Ce délai est spécifié en unités de période d'actualisation du contrôleur. Le dépassement de cette limite donne lieu à un défaut d'actualisation de commande.	Nombre de périodes d'actualisation du contrôleur
22	Facultatif	Set		Limite inférieure du délai d'actualisation du contrôleur	USINT	Représente la limite de délai supérieure admise d'une actualisation de connexion Contrôleur - Dispositif (C-D). Ce délai est spécifié en unités de période d'actualisation du contrôleur. Le dépassement de cette limite donne lieu à une alarme d'actualisation de commande.	Nombre de périodes d'actualisation du contrôleur
28	Facultatif	Get		Variation de synchronisation	UDINT	Représente la variation statistique en cours de la valeur Décalage du temps système par rapport à la valeur Décalage de temps système moyenne.	Nanosecondes
29	Facultatif	Set		Sync Threshold	UDINT	Détermine le seuil de l'écart observé du temps système sous	Nanosecondes Valeur par défaut: valeur

ID attr	Besoin dans la mise en œuvre	Règle d'accès	NV	Nom de l'attribut	Type de données	Description d'attribut	Sémantique des valeurs
						lequel l'Objet Axe de mouvement est considéré comme étant synchronisé. Le service Synchronisation de groupe l'utilise comme critère de réponse positive.	minimale dépendant du dispositif.
30	Facultatif	Get		Sync Update Delay	REAL	Cette valeur est le produit du nombre de sauts de réseau qui sépare le dispositif d'entraînement de l'horloge grand-maître, déterminé par l'attribut Objet de synchronisation temporelle. Pas supprimés et l'intervalle de synchronisation, déterminé par l'attribut Objet de synchronisation temporelle, Intervalle de synchronisation.	Secondes
31	Facultatif	Set		Ensemble de données temporelles	OCTET	Cet octet de matriciel contient des indicateurs qui déterminent l'utilisation et le format des informations de datation du contrôleur et du dispositif d'entraînement. Cette valeur est déduite de l'élément Configuration temporelle envoyé par le contrôleur dans le cadre d'une connexion Contrôleur - Dispositif (C-D) et devient la valeur Configuration temporelle envoyée par le dispositif dans le cadre d'une connexion Dispositif - Contrôleur (D-C).	Champ de bit: 0 = Période d'actualisation 1 = Datations 2 = Diagnostic de synchronisation 3-7 = (Réservé)

7.2.2 Sémantique

7.2.2.1 Attribut N° 10 – Données de connexion utilisées du contrôleur

Cet attribut contient la structure des données utilisées définie par la connexion C-D CIP Motion. Le contenu de cette structure de données dynamique est actualisé par le contrôleur lors de la période d'actualisation du contrôleur. Cette structure, présentée à la Figure 45, est définie de manière dynamique par les données d'en-tête de connexion. Voir 6.4.3 pour une description détaillée de la structure de données de connexion.

← Mot de 32 bits →
Format de connexion Contrôleur - Dispositif (C-D)
En-tête de connexion
En-tête de données d'instance 1
Bloc de données cycliques
Bloc de données d'écriture cycliques
Bloc de données d'événement
Bloc de données de service
En-tête de données d'instance 2
...

Figure 45 – Format de données de connexion utilisées du contrôleur

7.2.2.2 Attribut N° 11 – Données de connexion générées du contrôleur

Cet attribut contient la structure des données générées définie par la connexion D-C CIP Motion. Cette structure de données dynamique est actualisée par le dispositif de mouvement et transmise au contrôleur lors de la période d'actualisation du contrôleur. Cette structure, présentée à la Figure 46, est définie de manière dynamique par les données d'en-tête de connexion. Voir 6.4.4 pour une description détaillée de la structure de données de connexion.

← Mot de 32 bits →
Format de connexion Dispositif - Contrôleur (D-C)
En-tête de connexion
En-tête de données d'instance 1
Bloc de données cycliques
Bloc de données de lecture cycliques
Bloc de données d'événement
Bloc de données de service
En-tête de données d'instance 2
...

Figure 46 – Format de données de connexion générées du contrôleur

7.2.2.3 Attribut N° 14 – Contrôle de nœud

Contient les bits utilisés pour contrôler le comportement du nœud de communication du dispositif de mouvement associé (voir Tableau 20).

Tableau 20 – Définitions de bit Nœud de contrôle

Bit	Oblig. Facult.	Nom	Description
0	O	Commande à distance	Ce bit est utilisé pour demander que le nœud du dispositif transmette le contrôle du dispositif de mouvement au contrôleur. Si ce bit est vide, le dispositif CIP Motion est sous le contrôle exclusif de l'interface locale. Le contrôleur CIP Motion ne peut pas agir sur le comportement du dispositif sans le rebasculer en mode Commande à distance. En cas de perte de la connexion CIP Motion alors que le dispositif est sous commande à distance, ce dernier peut être contrôlé par l'interface locale.
1	F	Commande de synchronisation	Ce bit est utilisé par le contrôleur pour demander le fonctionnement synchrone du dispositif CIP Motion. Le fonctionnement synchrone se caractérise par le fait que le temporisateur local du dispositif est synchronisé avec le temps système, et que le nœud du dispositif commence à utiliser la datation du contrôleur et le décalage temporel pour traiter les données de connexion et planifier des actualisations de connexion Contrôleur - Dispositif (C-D) ultérieures. Le bit Commande de synchronisation implique que les valeurs Datation du contrôleur et Décalage temporel

Bit	Oblig. Facult.	Nom	Description
			sont valides. Le bit doit rester défini pendant la durée du fonctionnement synchrone. En mode asynchrone, il n'existe aucune exigence de synchronisation ni de datation du temporisateur local, ni de données de temps associées valides.
2	O	Données de contrôleur valides	Ce bit doit être défini pour que le nœud de dispositif traite les blocs de données d'instance. La séquence d'initialisation de la connexion peut présenter une période de temps dans laquelle les données de connexion des blocs de données d'instance ne sont pas synchronisées. Cette condition peut être reconnue par le dispositif par une vérification préalable de ce bit et en détectant qu'il est vide.
3	O	Réinitialisation suite à un défaut de nœud	Ce bit est utilisé pour demander au dispositif CIP Motion de réinitialiser le nœud de communication et de tenter d'annuler le bit Défaut de nœud. Lorsque le contrôleur détecte que le bit Défaut de nœud a été annulé, le bit Réinitialisation suite à un défaut de nœud l'est également. Si le défaut de communication persiste, le bit Défaut de nœuds peut rester défini, auquel cas un cycle d'alimentation ou un processus de reconfiguration complet doit être réalisé.
4 à 5			(Réserve)
6 à 7			(Spécifique au fournisseur)

7.2.2.4 Attribut N° 15 – Statut de nœud

Contient les bits utilisés pour indiquer le statut du nœud de communication du dispositif CIP Motion associé (voir Tableau 21).

Tableau 21 – Définitions du bit Statut du nœud

Bit	Oblig. Facult.	Nom	Description
0	O	Mode distant	Le bit Mode distant est utilisé pour indiquer que le nœud du dispositif a transmis le contrôle total du dispositif de mouvement au contrôleur. Si ce bit est vide, le dispositif ne doit pas agir sur des données contenues dans les voies cycliques ou de service Contrôleur - Dispositif.
1	F	Mode sync	Le bit Mode sync indique si le nœud du dispositif est synchronisé ou non. Le fonctionnement synchrone est défini comme étant l'état de synchronisation du temporisateur local du nœud du dispositif avec le temps système, le nœud du dispositif utilisant les datations de connexion pour traiter les données de connexion. La définition du bit Mode sync implique également que la datation du dispositif est valide. Le bit doit rester défini pendant la durée du fonctionnement synchrone. Si le bit Mode sync est vide, le dispositif est supposé être en mode asynchrone. En mode asynchrone, il n'existe aucune exigence de synchronisation ni de datation du temporisateur local, ni de données de temps associées valides.
2	O	Données valides	Le bit Données valides doit être défini pour que le contrôleur traite les blocs de données d'instance provenant du nœud du dispositif. La séquence d'initialisation de la connexion peut présenter une période de temps dans laquelle les données de connexion des blocs de données d'instance ne sont pas synchronisées. Cette condition peut être reconnue par le contrôleur en procédant à une vérification préalable de ce bit.
3	O	Défaut de nœud	Le bit Défaut de nœud est utilisé pour indiquer que le dispositif a détecté une ou plusieurs conditions de défaut liées au nœud de communication. Les conditions de défaut spécifiques peuvent être déterminées par l'attribut Défaut de nœud. Si ce bit est vide, il n'existe aucune condition de défaut. Le bit Défaut de nœud peut être annulé en définissant le bit Réinitialisation suite à un défaut de nœud du mot Contrôle de nœud. Tous les axes associés sont désactivés en présence d'une condition Défaut de nœud.

Bit	Oblig. Facult.	Nom	Description
4 à 5			(Réservé)
6 à 7			(Spécifique au fournisseur)

7.2.2.5 Attribut N° 16 – Défauts de nœud

Cet attribut est un code de défaut de 8 bits utilisé pour signaler la présence d'une condition de défaut particulière liée au nœud de communication du dispositif CIP Motion (voir Tableau 22).

Tableau 22 – Définitions du code Défauts de nœud

Code	Nom	Description
0	Absence de défaut	Indique l'absence de condition de défaut au niveau du nœud de communication du dispositif.
1	Défaut d'actualisation de commande	Le bit Défaut d'actualisation de commande est utilisé pour indiquer que les actualisations provenant du contrôleur ont beaucoup de retard.
2	Défaut de chien de garde du processeur	Le bit Défaut de chien de garde du processeur indique que le processeur associé au nœud du dispositif a fait l'objet d'une condition de surcharge excessive qui a déclenché le mécanisme de chien de garde du processeur associé.
3	Défaut matériel	Le bit Défaut matériel indique que le support matériel critique (FPGA, ASIC, etc.) associé au nœud du dispositif a fait l'objet d'une condition de défaut.
4	Erreur de format de données	Ce défaut indique qu'une erreur s'est produite dans le format de données entre le contrôleur et le dispositif. Un défaut de correspondance de révision de format peut par exemple être à l'origine de ce défaut.
5 à 127	(Réservé)	
128 à 255	(Spécifique au fournisseur)	

7.2.2.6 Attribut N° 17 – Alarmes de nœud

Cet attribut est un code d'alarme de 8 bits utilisé pour indiquer des conditions d'alarme particulières liées au nœud de communication du dispositif CIP Motion (voir Tableau 23). Les alarmes ne donnent pas lieu à une action directe.

Tableau 23 – Définitions du code Alarmes de nœud

Code	Nom	Description
0	Pas d'alarme	Indique l'absence de condition d'alarme au niveau du nœud de communication du dispositif.
1	Alarme d'actualisation de commande	Le bit Alarme d'actualisation de commande est utilisé pour indiquer que les actualisations provenant du contrôleur ont du retard.
2	Alarme de surcharge du processeur	L'Alarme de surcharge du processeur indique que le processeur associé au dispositif fait l'objet de conditions de surcharge pouvant éventuellement donner lieu à un défaut.
3	Alarme de synchronisation	L'Alarme de synchronisation indique que la Variation de synchronisation a dépassé le seuil de synchronisation alors que le dispositif fonctionnait en Mode sync.
4 à 127	(Réservé)	
128 à 255	(Spécifique au fournisseur)	

7.3 Attributs d'instance

7.3.1 Généralités

Ce paragraphe présente tous les attributs pris en charge d'une instance Objet Axe de mouvement. Compte tenu de leur nombre important, les attributs ont été organisés en catégorie fonctionnelle. Chaque groupe fonctionnel peut être organisé en indiquant d'abord les attributs de statut et de signal de l'objet, suivis des attributs de configuration de l'objet.

En raison du nombre important d'attributs d'instance pris en charge par cet objet, des ID d'attribut de 16 bits sont requis. Saut indication contraire, la valeur par défaut de tous les attributs facultatifs est 0.

NOTE Aucun attribut associé à cet objet ne requiert un stockage non volatile. La valeur "No" est donc implicite pour tous les attributs de la colonne NV.

Légende des tableaux: (voir 7.1.5 et 7.1.6 pour plus de détails).

(O) – Bit ou énumération obligatoire – doit être pris en charge dans la mise en œuvre

(F) – Bit ou énumération facultatif – la prise en charge est laissée à la discrétion du fournisseur

Modes de commande:

P – Asservissement de position

V – Commande de vitesse

T – Asservissement de couple

N – Pas d'asservissement

Méthodes de commande:

F – Dispositifs d'entraînement de fréquence (V/Hz ou VFD)

C – Boucle fermée (commande vectorielle)

Set* – Indique que les attributs sont en principe définis par l'en-tête de connexion CIP Motion et pas par un service Set.

% Unités assignées du dispositif – défini comme étant le pourcentage des caractéristiques assignées continues du dispositif avec un fonctionnement de 100 % à la spécification assignée continue du dispositif. Cette unité peut être appliquée aux attributs liés à la vitesse, au couple, à la force, au courant, à la tension et à la puissance. Les "dispositifs" concernés peuvent être le moteur, l'onduleur, le convertisseur ou le régulateur de bus. Cette unité peut être utilisée indépendamment du fait que la valeur d'attribut représente un niveau instantané ou moyenné dans le temps. L'unité appropriée à la puissance assignée du dispositif est implicite. Comme pour tous les attributs dont l'unité est exprimée en %, une valeur d'attribut de 100 signifie 100 %.

Unités de réaction: En règle générale, les attributs relatifs à la dynamique du mouvement expriment le déplacement en termes de résolution de comptage du dispositif de réaction sélectionné. Étant donné que le dispositif CIP Motion peut utiliser différentes voies de réaction pour différentes boucles de régulation en fonction de la Configuration de réaction, la détermination du dispositif de réaction qui s'applique dépend de la Configuration de réaction.

Le Tableau 24 est un tableau croisé permettant de déterminer les nombres de réactions appropriées ou les unités à appliquer à un attribut en fonction de son type d'unité dynamique (position, vitesse ou accélération), ainsi que la Configuration de réaction. Si une instance d'axe est configurée pour l'absence de réaction, c'est-à-dire un fonctionnement sans capteur (V/Hz, entraînement asservi de vitesse sans capteur ou future prise en charge pas à pas), les

nombre de réaction ne s'appliquent pas, la dynamique de mouvement doit donc être exprimée en termes de déplacement du moteur.

Tableau 24 – Unité dynamique par rapport à la Configuration de réaction

Unités dynamiques	Configuration de réaction			
	Pas de réaction	Réaction 1	Réaction 2	Réaction en double boucle
Unités d'asservissement de position	Nombre de moteurs	Nombre de réactions 1	Nombre de réactions 2	Nombre de réactions 2
Unités de commande de vitesse	Unités moteur	Unités de réaction 1	Unités de réaction 2	Unités de réaction 1
Unités de commande d'accélération	Unités moteur	Unités de réaction 1	Unités de réaction 2	Unités de réaction 1

7.3.2 Attributs de configuration de commande de mouvement

7.3.2.1 Généralités

Le tableau d'attribut ci-dessous du Tableau 25 contient les attributs de configuration de commande de mouvement de base associés à l'instance Objet Axe de mouvement. Ces attributs régissent les aspects du comportement général de l'Objet Axe de mouvement.

Set* → Ces attributs sont en général mis à jour par l'intermédiaire de l'Ensemble de données de consigne cycliques de la connexion Contrôleur - Dispositif (C-D) CIP Motion. S'ils sont inclus sous la forme de données de consigne cycliques, il convient de ne pas actualiser ces attributs via un service Set.

Tableau 25 – Attributs de configuration de commande de mouvement

ID attr	Besoin dans la mise en œuvre	Règle d'accès	NV	Nom de l'attribut	Type de données	Description d'attribut	Sémantique des valeurs
80	Obligatoire - Tous	Set*		Mode de commande	OCTET	Attribut qui détermine le comportement de commande dynamique général de l'instance d'axe de mouvement.	Voir la sémantique en 7.3.2.2.1
81	Obligatoire - Tous	Set		Méthodes de commande	USINT	Code énuméré qui détermine l'algorithme de base de commande de moteur appliqué par le dispositif pour commander le comportement dynamique du moteur.	Voir la sémantique en 7.3.2.2.2

7.3.2.2 Sémantique

7.3.2.2.1 Attribut N° 80 – Mode de commande

L'attribut Mode de commande est composé d'un champ Commande de moteur énuméré à 4 bits réservé pour un développement ultérieur (voir Figure 47).

← OCTET de 8 bits →	
Format du bit Mode de commande	
Bit 4-7: (Réservé)	Bits 0-3: Énumération Commande de moteur

Figure 47 – Champ du bit Mode de commande

L'énumération du champ Commande de moteur détermine le comportement dynamique spécifique du moteur que le dispositif est tenu de commander pour cette instance d'axe. La vue système de ces modes de commande est décrite en détail en 4.2. Le Tableau 26 présente un bref récapitulatif des modes Commande de moteur:

Tableau 26 – Définitions de l'énumération du champ Commande de moteur

Énum.	Oblig. Facult.	Nom	Description
0	O/N	Pas d'asservissement	Aucune commande de moteur n'est fournie dans ce mode, mais l'interface vers un dispositif de réaction spécifique, en tant que source de réaction principale, est possible par l'intermédiaire de l'attribut Configuration de réaction.
1	O/P	Asservissement de position	Le dispositif cherche à commander la position (ou l'orientation) du moteur.
2	O/PV	Commande de vitesse	Le dispositif cherche à commander la vitesse du moteur.
3	F	Commande d'accélération	Le dispositif cherche à commander l'accélération du moteur.
4	O/C	Asservissement de couple	Le dispositif cherche à commander la sortie de couple du moteur.
5	F/C	Asservissement de courant	Le dispositif cherche à commander le couple générant le courant pour le moteur.
6 à 15		(Réservé)	-

7.3.2.2.2 Attribut N° 81 – Mode d'asservissement

L'attribut Méthode de commande est un code énuméré de 8 bits qui détermine l'algorithme de commande de base appliqué par le dispositif pour commander le comportement dynamique du moteur associé à une instance d'axe (voir Tableau 27).

Tableau 27 – Définitions de l'énumération du champ Méthode de commande

Énum.	Oblig. Facult.	Nom	Description
0	O/N	Pas d'asservissement	Aucun asservissement n'est associé à ce mode de commande en l'absence de commande de moteur explicite fournie par le dispositif pour cette instance d'axe.
1	O/F	Contrôle de fréquence	Le contrôle de fréquence est une méthode de commande en "boucle ouverte" qui applique la tension au moteur, en général proportionnellement à la fréquence ou la vitesse commandée. Cette méthode de commande est associée à des dispositifs d'entraînement à fréquence variable (VFD) ou dispositifs d'entraînement Volts/Hertz.
2	O/C	Commande vectorielle PI	La commande vectorielle PI est une méthode de commande en "boucle fermée" qui utilise la réaction réelle ou estimée de la commande PI en boucle fermée des éléments dynamiques du moteur, c'est-à-dire la position, la vitesse, l'accélération et le couple, incluant toujours le contrôle PI indépendant en boucle fermée des composants Iq et Id du vecteur de courant du moteur.
3 à 255		(Réservé)	-

7.3.3 Attributs du moteur

7.3.3.1 Généralités

Les tableaux d'attribut ci-dessous contiennent les attributs de configuration du moteur associés à l'instance Objet Axe de mouvement qui s'appliquent aux différentes technologies de motorisation. Ces technologies incluent les moteurs rotatifs, linéaires, à aimant permanent et à induction triphasés. Les attributs du moteur sont donc organisés en fonction des différents types de moteur. La catégorie Besoin dans la mise en œuvre d'un attribut est fonction du contexte du type de moteur et, par conséquent, du contexte du tableau dans lequel ces attributs apparaissent. Le cas échéant, Standard peut être distingué de Facultatif par les abréviations PM (Aimant permanent - Permanent Magnet) et IM (Moteurs à induction - Induction Motors). Il est sous-entendu que ces attributs de moteur ne sont pas obligatoires en mode Pas d'asservissement (Réaction uniquement).

Ce paragraphe relatif aux attributs du moteur a pour objet de définir l'ensemble minimal d'attributs obligatoires à la prise en charge de l'interchangeabilité du dispositif CIP Motion. Il s'agit de s'assurer que le contrôleur a fourni le nombre suffisant de données paramétriques pour un dispositif conforme à CIP Motion, c'est-à-dire un dispositif d'entraînement, et qu'un moteur donné est commandé de manière efficace.

Pour les moteurs à induction, l'Objet Axe de mouvement s'appuie sur le modèle de motorisation du circuit équivalent phase-neutre recommandé IEEE reposant sur la configuration "Wye" (voir Figure 48). Les valeurs de réactance, X , sont associées à leurs valeurs d'inductance correspondantes, L , par l'équation $X = \omega L$, où ω est la fréquence assignée du moteur. La notation primaire (X_2 , R_2' , par exemple) indique que les valeurs de composant rotorique réelles X_2 et R_2 font référence au côté stator du rapport d'enroulement stator-rotor.

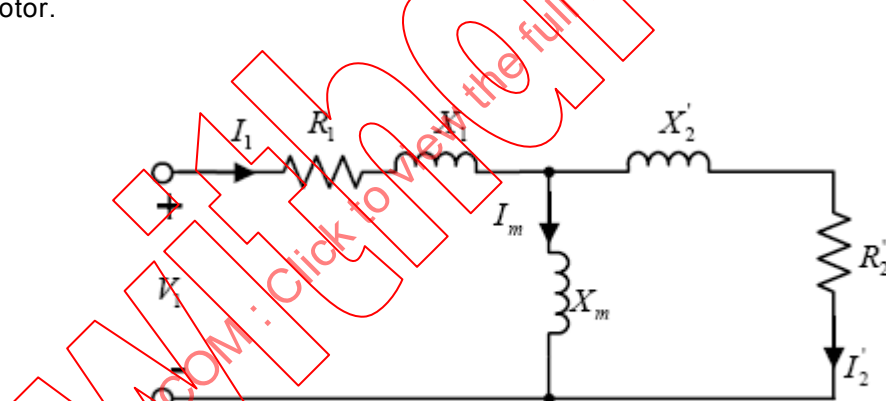


Figure 48 – IEEE par modèle de moteurphasé

Pour les moteurs à aimant permanent, l'Objet Axe de mouvement suppose de définir tous les paramètres moteur dans le contexte d'un modèle de moteur entre phases.

7.3.3.2 Attributs généraux du moteur

Le Tableau 28 et le Tableau 29 contiennent les attributs généraux du moteur qui s'appliquent à toutes les technologies de motorisation.

Tableau 28 – Attributs généraux d'informations sur le moteur

ID attr	Besoin dans la mise en œuvre	Règle d'accès	NV	Nom de l'attribut	Type de données	Description d'attribut	Sémantique des valeurs
1310	Facultatif	Get		Numéro de lot du moteur	CHAÎNE COURTE	Chaîne de 32 caractères qui précise le numéro de lot du moteur. Cet attribut ne s'applique pas lorsque la source de données du moteur est NV du contrôleur.	
1311	Facultatif	Get		Numéro de série du moteur	CHAÎNE COURTE	Chaîne de 16 caractères qui précise le numéro de série du moteur. Cet attribut s'applique uniquement lorsque la source de données du moteur est NV du moteur.	Par exemple: 0012003400560078
1312	Facultatif	Get		Code de date du moteur	CHAÎNE COURTE	Chaîne de 16 caractères qui précise la date de fabrication du moteur. Cet attribut s'applique uniquement lorsque la source de données du moteur est NV du moteur.	

Tableau 29 – Attributs généraux de configuration du moteur

ID attr	Besoin dans la mise en œuvre	Règle d'accès	NV	Nom de l'attribut	Type de données	Description d'attribut	Sémantique des valeurs
1313	Obligatoire	Set		Source de données du moteur	USINT	Énumération qui précise la source des données du moteur pour le dispositif d'entraînement. NV du contrôleur implique que les attributs du moteur sont déduits de la mémoire non volatile du contrôleur lors du processus de configuration du dispositif d'entraînement. NV du dispositif d'entraînement implique que les attributs du moteur sont déduits directement de la mémoire non volatile du dispositif d'entraînement. Dans ce mode, le contrôleur ne doit envoyer aucun paramètre moteur au dispositif d'entraînement. NV moteur implique que les attributs du moteur sont déduits de la mémoire non volatile d'un dispositif de réaction intelligent monté sur le moteur équipé d'une interface série. De même, dans ce mode, le contrôleur ne doit envoyer aucun paramètre moteur au dispositif d'entraînement.	Bits 0-3: Énum 0 = NV contrôleur (O) 1 = NV dispositif d'entraînement (F) 2 = NV moteur (F) 3 à 127 = (réservé) 128 à 255 = (spécifique au fournisseur)

ID attr	Besoin dans la mise en œuvre	Règle d'accès	NV	Nom de l'attribut	Type de données	Description d'attribut	Sémantique des valeurs
1314	Obligatoire	Set		Code de dispositif du moteur	UDINT	Numéro unique affecté à un numéro de lot du moteur. Cette valeur permet de s'assurer que le moteur et les données de configuration du dispositif de réaction provenant du contrôleur correspondent aux données réelles du moteur et de réaction associées au dispositif d'entraînement. Cette comparaison est uniquement valide si la source de données du moteur est NV contrôleur, le moteur étant équipé d'un dispositif de réaction intelligent. Si les codes ne correspondent pas, le dispositif d'entraînement génère un acquittement négatif. Les codes de dispositif du moteur sont affectés par le constructeur du moteur.	
1315	Obligatoire	Set		Type de moteur	USINT	Énumération qui précise le type de moteur.	Énumération: 0 = à aimant permanent rotatif (F) 1 = à induction rotatif (F) 2 = à aimant permanent linéaire (F) 3 = à induction linéaire (F) 4 à 127 = (réservé) 128 à 255 = (spécifique au fournisseur)
1316	Obligatoire	Set		Unité moteur	UINT	Unité de mesure du déplacement du moteur. Également utilisé pour le fonctionnement sans capteur.	Énumération: 0 = Rév 1 = Mètre 2 à 127 = (réservé) 128 à 255 = (spécifique au fournisseur)
1317	Facultatif	Set		Polarité du moteur	USINT	Valeur énumérée utilisée pour établir le sens du mouvement du moteur en réponse à une sortie de dispositif d'entraînement positive. La polarité normale est définie comme étant le sens de mouvement du moteur lorsque les cosses du moteur UVW sont connectées conformément aux spécifications publiées du dispositif d'entraînement. La polarité inverse bascule le phasage UVW vers UWV, de sorte que le moteur se déplace dans le sens opposé en réponse à une sortie de dispositif d'entraînement positive. Cet attribut peut être utilisé pour accorder le sens de déplacement à la définition de déplacement positif de l'utilisateur. Il peut être utilisé conjointement avec le bit Polarité de réaction afin de fournir une réaction négative, si un contrôle de boucle fermée est requis.	Énumération: 0 = Polarité normale 1 = Polarité inverse

ID attr	Besoin dans la mise en œuvre	Règle d'accès	NV	Nom de l'attribut	Type de données	Description d'attribut	Sémantique des valeurs
1318	Obligatoire	Set		Tension assignée du moteur	REAL	Valeur flottante qui spécifie la tension c.a. assignée du moteur. Il s'agit de la tension entre phases appliquée au moteur pour atteindre la vitesse assignée en pleine charge.	Volts (eff)
1319	Obligatoire	Set		Courant assigné continu du moteur	REAL	Valeur flottante qui spécifie le courant assigné continu c.a. du moteur. Il s'agit du courant appliqué au moteur dans les conditions de pleine charge à vitesse et tension assignées.	Ampères (eff)
1320	Obligatoire – PM Facultatif – IM	Set		Courant de crête assigné du moteur	REAL	Valeur flottante qui spécifie le courant assigné de crête ou intermittent du moteur. Le courant assigné de crête du moteur est souvent déterminé par les contraintes thermiques de l'enroulement du stator ou par les limites de saturation du matériau magnétique du moteur à aimant permanent.	Ampères (eff)
1321	Obligatoire – IM Facultatif – PM	Set		Puissance de sortie assignée du moteur	REAL	Valeur flottante qui spécifie la puissance de sortie assignée du moteur. Il s'agit de la puissance de sortie du moteur dans les conditions de pleine charge à courant, vitesse et tension assignés.	Watts
1322	Obligatoire	Set		Limite de surcharge du moteur	REAL	Valeur flottante qui spécifie la limite de surcharge thermique maximale du moteur. Cette valeur est en général de 100 %, correspondant à la puissance dissipée dans le cadre d'un fonctionnement à courant assigné continu du moteur. Elle peut être plus élevée si des options de refroidissement sont appliquées. Pour les moteurs à induction, cet attribut est lié au facteur de service du moteur. Si la valeur de l'attribut Capacité du moteur associée au modèle thermique du moteur dépasse la limite de surcharge du moteur, le dispositif d'entraînement peut éventuellement déclencher une action Surcharge du moteur prédéterminée. La limite de surcharge du moteur peut également être utilisée pour déterminer la limite de capacité thermique absolue du moteur, c'est-à-dire la tolérance de fabrication de surcharge thermique du moteur qui, si elle est dépassée, génère une exception Tolérance de fabrication de surcharge thermique du moteur.	% de la valeur assignée du moteur

ID attr	Besoin dans la mise en œuvre	Règle d'accès	NV	Nom de l'attribut	Type de données	Description d'attribut	Sémantique des valeurs
1323	Facultatif	Set		Thermo-contacteur intégral du moteur	BOOL	Valeur booléenne qui indique si le moteur est doté d'un thermocontacteur intégral.	0 = Non 1 = Oui
1324	Facultatif	Set		Température d'enroulement max. du moteur	REAL	Valeur flottante qui spécifie la température d'enroulement maximale du moteur.	°C
1325	Facultatif	Set		Capacité thermique Enroulement-Ambiant du moteur	REAL	Valeur flottante qui spécifie la capacité thermique enroulement-ambiant.	J/°C
1326	Facultatif	Set		Résistance thermique Enroulement-Ambiant du moteur	REAL	Valeur flottante qui spécifie la résistance thermique enroulement-ambiant.	°C/W

7.3.3.3 Attributs généraux du moteur à aimant permanent

Le Tableau 30 contient les attributs de configuration du moteur qui s'appliquent aux types de moteur à aimant permanent en général.

Tableau 30 – Attributs généraux de configuration du moteur à aimant permanent

ID attr	Besoin dans la mise en œuvre	Règle d'accès	NV	Nom de l'attribut	Type de données	Description d'attribut	Sémantique des valeurs
1327	Obligatoire	Set		Résistance du moteur à aimant permanent	REAL	Valeur flottante qui spécifie la résistance entre phases d'un moteur à aimant permanent.	Ohms
1328	Obligatoire	Set		Inductance du moteur à aimant permanent	REAL	Valeur flottante qui spécifie l'inductance entre phases d'un moteur à aimant permanent.	Henries

7.3.3.4 Attributs généraux du moteur rotatif

Le Tableau 31 contient les attributs de configuration du moteur qui s'appliquent spécifiquement aux types de moteur rotatif.

Tableau 31 – Attributs généraux de configuration du moteur rotatif

ID attr	Besoin dans la mise en œuvre	Règle d'accès	N V	Nom de l'attribut	Type de données	Description d'attribut	Sémantique des valeurs
1329	Obligatoire	Set		Pôles du moteur rotatif	UINT	Entier qui spécifie le nombre de pôles par révolution des moteurs rotatifs. Cette valeur est toujours un nombre pair, les pôles évoluant toujours par paires.	
1330	Facultatif	Set		Inertie du moteur rotatif	REAL	Valeur flottante qui spécifie l'inertie sans charge d'un moteur rotatif.	kg.m ²
1331	Obligatoire	Set		Vitesse assignée du moteur rotatif	REAL	Valeur flottante qui spécifie la vitesse assignée d'un moteur rotatif à tension et courant assignés. Pour les moteurs à induction, cette valeur est souvent considérée comme la vitesse de base. Souvent, la valeur représente également la vitesse à laquelle la puissance de sortie du moteur est optimisée.	r/min
1332	Facultatif	Set		Vitesse max du moteur rotatif	REAL	Valeur flottante qui spécifie la vitesse maximale absolue d'un moteur en unités de r/min. Cette vitesse peut être déterminée par les limites du moteur ou par celles du système mécanique. De manière spécifique, elle peut représenter la vitesse maximale de fonctionnement sûr, la vitesse maximale continue "sans charge", la vitesse continue maximale du codeur ou la vitesse continue maximale de défilement du moteur. Cette valeur peut être utilisée pour déterminer l'exception Survitesse du moteur.	r/min
1333	Facultatif	Set		Coefficient d'amortissement du moteur rotatif	REAL	Valeur flottante qui spécifie l'amortissement ou le frottement visqueux avec un moteur rotatif.	N.m/(rad/s)

7.3.3.5 Attributs généraux du moteur linéaire

Le Tableau 32 contient les attributs de configuration du moteur qui s'appliquent spécifiquement aux types de moteur linéaire.

Tableau 32 – Attributs généraux de configuration du moteur linéaire

ID attr	Besoin dans la mise en œuvre	Règle d'accès	NV	Nom de l'attribut	Type de données	Description d'attribut	Sémantique des valeurs
1334	Obligatoire	Set		Pas polaire du moteur linéaire	REAL	Valeur flottante qui spécifie le pas polaire d'un moteur linéaire en mètres, équivalent à la longueur de cycle électrique.	mètres

ID attr	Besoin dans la mise en œuvre	Règle d'accès	NV	Nom de l'attribut	Type de données	Description d'attribut	Sémantique des valeurs
1335	Obligatoire	Set		Vitesse assignée du moteur linéaire	REAL	Valeur flottante qui spécifie la vitesse assignée d'un moteur linéaire à tension et courant assignés. Pour les moteurs à induction, cette valeur est souvent considérée comme la vitesse de base. Souvent, la valeur représente également la vitesse à laquelle la puissance de sortie assignée du moteur est mesurée.	m/s
1336	Facultatif	Set		Masse en mouvement du moteur linéaire	REAL	Valeur flottante qui spécifie la masse en mouvement sans charge d'un moteur linéaire.	kg
1337	Facultatif	Set		Vitesse max. du moteur linéaire	REAL	Valeur flottante qui spécifie la vitesse maximale absolue d'un moteur linéaire, en m/s. Cette vitesse peut être déterminée par les limites du moteur ou par celles du système mécanique. De manière spécifique elle peut représenter la vitesse maximale de fonctionnement sûr, la vitesse maximale continue "sans charge", la vitesse continue maximale du codeur ou la vitesse continue maximale de défilement du moteur. Cette valeur peut être utilisée pour déterminer l'exception Survitesse du moteur.	m/s
1338	Facultatif	Set		Coefficient d'amortissement du moteur linéaire	REAL	Valeur flottante qui spécifie l'amortissement ou le frottement visqueux avec un moteur linéaire, en unités.	N/(m/s)

7.3.3.6 Attributs du moteur à aimant permanent rotatif

Le Tableau 33 contient les attributs de configuration du moteur qui s'appliquent spécifiquement aux types de moteur rotatif.

Tableau 33 – Attributs de configuration du moteur à aimant permanent rotatif

ID attr	Besoin dans la mise en œuvre	Règle d'accès	NV	Nom de l'attribut	Type de données	Description d'attribut	Sémantique des valeurs
1339	Obligatoire	Set		Couple assigné du moteur à aimant permanent	REAL	Valeur flottante qui spécifie le couple assigné continu d'un moteur à aimant permanent rotatif.	N.m
1340	Obligatoire	Set		Constante de couple du moteur à aimant permanent	REAL	Valeur flottante qui spécifie la constante de couple d'un moteur à aimant permanent rotatif, en newton-mètres par ampère eff.	N.m/ A (eff)
1341	Obligatoire	Set		Constante de tension du moteur à aimant rotatif	REAL	Valeur flottante qui spécifie la constante de tension (ou force contre-électromotrice) d'un moteur à aimant permanent rotatif, en Volts eff entre phases par kRPM.	Volts (eff)/kRPM

7.3.3.7 Attributs du moteur à aimant linéaire

Le Tableau 34 contient les attributs de configuration du moteur qui s'appliquent spécifiquement aux types de moteur à aimant permanent linéaire.

Tableau 34 – Attributs de configuration du moteur à aimant permanent linéaire

ID attr	Besoin dans la mise en œuvre	Règle d'accès	NV	Nom de l'attribut	Type de données	Description d'attribut	Sémantique des valeurs
1342	Obligatoire	Set		Force assignée du moteur	REAL	Valeur flottante qui spécifie la force assignée continue d'un moteur à aimant permanent linéaire.	N
1343	Obligatoire	Set		Constante de force du moteur	REAL	Valeur flottante qui spécifie la constante de force d'un moteur à aimant permanent linéaire, en newton-mètres par ampère eff.	N/A (eff)
1344	Obligatoire	Set		Constante de tension du moteur linéaire	REAL	Valeur flottante qui spécifie la constante de tension (ou force contre-électromotrice) d'un moteur à aimant permanent linéaire, en Volts eff entre phases par m/s.	Volts (eff) / (m/s)

7.3.3.8 Attributs du moteur à induction

Le Tableau 35 contient les attributs de configuration du moteur qui s'appliquent spécifiquement aux types de moteur à induction.

Tableau 35 – Attributs de configuration du moteur à induction

ID attr	Besoin dans la mise en œuvre	Règle d'accès	NV	Nom de l'attribut	Type de données	Description d'attribut	Sémantique des valeurs
1345	Obligatoire	Set		Fréquence assignée du moteur à induction	REAL	Valeur flottante qui spécifie la fréquence assignée d'un moteur à induction.	Hertz
1346	Obligatoire	Set		Courant de flux du moteur à induction	REAL	Référence de courant Id qui nécessite de générer un flux de moteur complet. Cette valeur est estimée par le courant assigné du moteur sans charge communément trouvé dans les fiches techniques du moteur à induction.	Ampères (eff)
1347	Obligatoire	Set		Résistance statorique du moteur à induction	REAL	Valeur flottante qui spécifie la résistance d'enroulement phase-neutre du circuit Y du stator comme indiqué dans R1 dans le modèle de moteur IEEE.	Ohms
1348	Obligatoire	Set		Réactance de fuite statorique du moteur à induction	REAL	Valeur flottante qui spécifie la réactance de fuite phase-neutre du circuit Y de l'enroulement statorique comme indiqué dans X_1 dans le modèle de moteur IEEE.	Ohms
1349	Obligatoire ^a	Set		Réactance d'aimantation du moteur à induction	REAL	Valeur flottante qui spécifie la réactance d'aimantation phase-neutre du circuit Y du moteur, à fréquence assignée, comme indiqué dans X_m dans le modèle de moteur IEEE.	Ohms

ID attr	Besoin dans la mise en œuvre	Règle d'accès	NV	Nom de l'attribut	Type de données	Description d'attribut	Sémantique des valeurs
1350	Obligatoire ^a	Set		Résistance rotorique du moteur à induction	REAL	Valeur flottante qui spécifie la résistance d'enroulement référencée statorique équivalente phase-neutre du rotor comme indiqué dans R_2' dans le modèle de moteur IEEE.	Ohms
1351	Obligatoire	Set		Réactance de fuite rotorique du moteur à induction	REAL	Valeur flottante qui spécifie l'inductance de fuite équivalente statorique phase-neutre du circuit Y de l'enroulement rotorique comme indiqué dans X_2' dans le modèle de moteur IEEE.	Ohms
^a Ces paramètres comportent une composante de température du moteur élevée que certains dispositifs d'entraînement contournent par diverses techniques d'adaptation ou de compensation. Néanmoins, ces paramètres sont classés pour les besoins d'interchangeabilité du dispositif.							

7.3.4 Attributs de réaction

7.3.4.1 Généralités

Le Tableau 36 et le Tableau 37 contiennent tous les attributs relatifs au retour en position associés à une instance Objet Axe de mouvement qui s'appliquent à différentes techniques de dispositif de réaction et d'interface de réaction.

NOTE Ces technologies d'interface de réaction incluent Digital AqB (signaux A quad B numériques), Sinus/Cosinus (signaux A quad B analogiques), Parallèle numérique (interface de bit numérique parallèle), SSI (Interface série synchrone), LDT (Transducteur à déplacement linéaire) et Résolveur. D'autres interfaces de réaction modernes prises en charge sont Hiperface® (Stegmann) et EnDat 2.1® & EnDat 2.2® (Heidenhain).

La catégorie "Besoin dans la mise en œuvre" d'un attribut repose sur le contexte des abréviations Type de réaction pour TT (AqB numérique), TP (Parallèle numérique), SC (Sinus/Cosinus), HI (Hiperface®), E21 (EnDat 2.1®), E22 (EnDat 2.2®), SS (SSI), LT (LDT – Transducteur à déplacement linéaire et RS (Résolveur). L'abréviation HS s'applique à tous les dispositifs qui prennent en charge les capteurs Hall.

Ce paragraphe relatif aux attributs de réaction a pour objet de définir l'ensemble minimal d'attributs obligatoires à la prise en charge de l'interchangeabilité du dispositif CIP Motion. Il s'agit de s'assurer que le contrôleur a fourni le nombre suffisant de données paramétriques pour un dispositif conforme à CIP Motion, permettant d'assurer une interface efficace avec un large éventail de types de dispositif de réaction.

Plusieurs interfaces de dispositif de réaction sont actuellement définies par l'Objet Axe de mouvement pour la réaction de boucle de régulation, à savoir Réaction 1 et Réaction 2. Dans les configurations d'asservissement de réaction double, Réaction 1 est en général associée au dispositif de réaction monté sur le moteur, Réaction 2 étant associée au dispositif de réaction en aval ou monté sur la machine. Pour tous les autres modes d'asservissement, Réaction 1 doit être utilisée pour l'interface de réaction singulière et est toujours requise pour la commutation du moteur à aimant permanent.

Si Pas d'asservissement est sélectionné pour l'instance Objet Axe de mouvement, toutes les voies de réaction non engagées peuvent faire office de source de réaction principale. Étant donné que Réaction 1 est en général affectée à la fonction de commande de l'instance Objet Axe de mouvement principale, Réaction 2, 3, 4, etc. sont des candidats probables aux sources de réaction principales. Pour cette version de l'Objet Axe de mouvement, deux voies de réaction supplémentaires (Réaction 3 et Réaction 4) sont prises en charge à cette fin.

Pour limiter la longueur des tableaux d'attribut de réaction ci-dessous, la lettre "n" dans le nom d'attribut "Réaction n" générique permet de spécifier le numéro de voie de réaction

associée. Les numéros de voie valides des attributs de réaction standard ouverts de l'Objet Axe de mouvement sont 1, 2, 3 et 4. Les ID d'attribut sont affectés en fonction du numéro de voie. La prise en charge des voies d'interface de réaction, 2, 3, 4 est facultative dans la mise en œuvre du dispositif. Toutefois, si le support matériel de l'une de ces voies de réaction est disponible dans un dispositif donné, ces attributs facultatifs sont clairement obligatoires dans la mise en œuvre pour que les voies puissent être utilisées.

Tableau 36 – Attributs généraux d'informations sur la réaction

ID attr	Besoin dans la mise en œuvre	Règle d'accès	N V	Nom de l'attribut	Type de données	Description d'attribut	Sémantique des valeurs
1400 + (n-1)x50	Facultatif - Tous	Get		Référence n de réaction	CHAINE COURTE	Chaîne de 32 caractères qui spécifie la référence du dispositif associé à Réaction n.	
1401 + (n-1)x50	Facultatif - Tous	Get		Numéro de série n de réaction	CHAINE COURTE	Chaîne de 16 caractères qui spécifie le numéro de série du dispositif associé à Réaction n.	Par exemple: 0012003400 560078
1402 + (n-1)x50	Obligatoire - NPVC Facultatif - T	Get		Position de réaction n	DINT	Position réelle de l'axe en fonction de Réaction n.	Nombre de Réactions n
1403 + (n-1)x50	Obligatoire - NPVC Facultatif - T	Get		Vitesse de Réaction n	REAL	Vitesse filtrée réelle de l'axe en fonction de Réaction n.	Unités/s Réaction n
1404 + (n-1)x50	Obligatoire - NPVC Facultatif - T	Get		Accélération de Réaction n	REAL	Accélération filtrée réelle de l'axe en fonction de Réaction n.	Unités/s ² Réaction n

Tableau 37 – Attributs de Configuration de réaction

ID attr	Besoin dans la mise en œuvre	Règle d'accès	N V	Nom de l'attribut	Type de données	Description d'attribut	Sémantique des valeurs
90	Obligatoire - PVT	Set*		Ensemble de données de réaction	OCTET	Champ matriciel qui détermine les valeurs de données de réaction réelles transmises au contrôleur dans le cadre d'une connexion Dispositif - Contrôleur (D-C).	Champ de bit: 0 = Retour en position (R/C) 1 = Retour en vitesse (R/C) 2 = Retour en accélération (O) 3 = Référence de couple (R/T) 4 = Référence de courant Iq (O) 5 = Sortie de tension (O) 6 = Sortie de fréquence (O) 7 = (Réservé)
82	Obligatoire - Tous	Set*		Configuration de réaction	OCTET	Cet attribut détermine la manière dont les différentes voies de réaction disponibles sont utilisées pour mettre en œuvre le Mode de commande sélectionné.	Voir la sémantique en 7.3.4.2
83	Obligatoire - Tous	Set*		Sélection de réaction principale	USINT	Cet attribut détermine la voie logique affectée à cette	Énumération: 0 = (Réservé) 1 = Réaction 1

ID attr	Besoin dans la mise en œuvre	Règle d'accès	N V	Nom de l'attribut	Type de données	Description d'attribut	Sémantique des valeurs
						instance d'axe lorsque la Configuration de réaction est définie sur Réaction principale.	2 = Réaction 2 3 = Réaction 3 4 = Réaction 4
84	Facultatif - PC	Set		Rapport de 1/2 compte de réaction	REAL	Nombre de Réactions 1 par Réaction 2. Cette valeur permet de convertir le nombre de réactions 2 en nombre de réactions 1 en cas de configuration pour le fonctionnement de boucle de position double.	Nombre de réactions 1 par nombre de réactions 2
1411 + (n-1)x50	Facultatif - NC	Set		Unité n de réaction	USINT	Unité de mesure du dispositif de réaction désigné. L'Unité de réaction de réaction 1 doit être évolutive par rapport à l'unité moteur configurée. Si cette dernière est définie sur Rév, l'Unité de réaction 1 doit être définie sur Rév. Si l'Unité moteur est définie sur Mètre, l'Unité de réaction 1 doit être définie sur Mètre ou Pouce, et le dispositif doit gérer la conversion entre les unités de déplacement du moteur linéaire et la réaction.	Énumération: 0 = Rév 1 = Mètre 2 = Pouce 3 à 255 = (réservé)
1412 + (n-1)x50	Facultatif	Set		Sélection de port n de réaction	REAL	Met en correspondance la voie logique "n" avec le port de réaction physique "m". Sélection du port n de réaction par défaut = n.	Énumération: 0 = (Réservé) 1 = Port 1 2 = Port 2 3 = Port 3 4 = Port 4 5 à 255 = Réservé
1413 + (n-1)x50	Obligatoire	Set		Type n de réaction	USINT	Identifie le type de dispositif de réaction connecté à l'interface de réaction associée. La prise en charge du dispositif de types de réaction individuels est laissée à la discrétion du constructeur du dispositif. Toutefois, si un type de réaction particulier est pris en charge, les attributs associés à ce type sont en général obligatoires dans la mise en œuvre.	Énumération: 0 = Absence de réaction 1 = AqB numérique (sans Hall) 2 = AqB numérique (avec Hall) 3 = Parallèle numérique (Absolue) 4 = Sinus/Cosinus (sans Hall) 5 = Sinus/Cosinus (avec Hall) 6 = Hiperface® 7 = EnDat 2.1® 8 = EnDat 2.2® 9 = Résolveur 10 = SSI 11 = LDT 12 à 127 = (réservé) 128 à 255 = (spécifique au fournisseur)
1414 + (n-1)x50	Facultatif	Set		Polarité n de réaction	USINT	Valeur énumérée permettant d'établir le sens de changement du compteur de réactions en réponse à	Énumération: 0 = Polarité normale 1 = Polarité inverse

ID attr	Besoin dans la mise en œuvre	Règle d'accès	N V	Nom de l'attribut	Type de données	Description d'attribut	Sémantique des valeurs
						un mouvement positif du dispositif de réaction associé. La polarité normale est définie comme étant celle donnant lieu à une augmentation des nombres de réaction lorsque le dispositif de réaction est connecté et déplacé dans le sens positif conformément aux spécifications publiées du dispositif d'entraînement. La polarité inverse bascule en interne la polarité de l'accumulateur de réactions de manière à diminuer le nombre lorsque le dispositif de réaction se déplace dans le sens positif. Cet attribut peut être utilisé pour accorder le sens de déplacement à la définition de déplacement positif de l'utilisateur. Il peut être utilisé conjointement avec le bit Polarité du moteur afin de fournir une réaction négative, si cette voie de réaction est utilisée pour le contrôle de boucle fermée.	
1415 + (n-1)x50	Obligatoire	Set		Mode n de réaction	USINT	Détermine la manière dont le dispositif applique la valeur de nombre de réactions. S'il est configuré pour le mode incrémentiel, le dispositif annule l'accumulateur de nombre de réactions à la mise sous tension. S'il est configuré pour le mode Absolu, le dispositif initialise l'accumulateur de nombre de réactions à la mise sous tension et applique la valeur de dispositif de réaction absolue lue dans le dispositif de réaction.	Énumération: 0 = Incrémentiel 1 = Absolu
1416 + (n-1)x50	Obligatoire - TT, SC, HI, E21, RS	Set		Résolution de cycle n de réaction	UDINT	Nombre de cycles de réaction par unité de réaction. Le cycle d'un dispositif AqB numérique est la résolution "en ligne" du codeur, alors qu'il représente la résolution de "cycle"	Cycles de réaction/Unité de réaction

ID attr	Besoin dans la mise en œuvre	Règle d'accès	N V	Nom de l'attribut	Type de données	Description d'attribut	Sémantique des valeurs
						sinusoïdal d'un dispositif de réaction Sin/Cos et le nombre de "pôles" du dispositif de réaction du résolveur. Pour les interfaces de dispositif de réaction en série numériques (SSI, par exemple) ou parallèles absolues, cette valeur représente la résolution "par pas" du dispositif.	
1 417 + (n-1)x50	Obligatoire - TT, SC, HI, E21, RS	Set		Interpolation de cycle n de réaction	UDINT	Nombre de réactions interpolées par cycle de réaction. Pour un dispositif AqB numérique, le matériel d'interface de réaction peut en général prendre en charge les valeurs d'interpolation 1, 2 ou 4. Pour le dispositif de réaction Sin/Cos, Hiperface, Endat 2.1 ou Résolveur le nombre est en général plus important, et est déterminé par la capacité d'interpolation du matériel d'interface du dispositif de réaction. La valeur 1 024 est en général utilisée dans ce cas. Pour les interfaces de dispositif de réaction en série numériques (SSI, par exemple) ou parallèles absolues, cette valeur est toujours 1 étant donné qu'il n'existe aucune chance d'interpolation basée sur le dispositif. La résolution effective du dispositif de réaction dans le nombre de réactions par unité de réaction est le produit de la résolution de cycle de réaction et de l'interpolation de cycle de réaction.	Nombre de réactions/Cycle de réaction
1420 + (n-1)x50	Obligatoire – TP, SS, HI, E21, E22	Set		Longueur de données n de réaction	USINT	Nombre de bits de données de réaction transférés sur la voie d'interface de données en série ou parallèle numérique d'un dispositif de réaction.	Nombre de bits
1421 + (n-1)x50	Facultatif – TP, SS	Set		Code de données n de réaction	USINT	Type de bits de codage de données de réaction utilisé par la voie d'interface de données en série ou parallèle désignée	Énumération: 0 = Binaire 1 = Gray

ID attr	Besoin dans la mise en œuvre	Règle d'accès	N V	Nom de l'attribut	Type de données	Description d'attribut	Sémantique des valeurs
						d'un dispositif de réaction.	
1422 + (n-1)x50	Facultatif - RS	Set		Rapport de transformateur du résolveur de réaction n	REAL	Spécification du rapport de transformateur du dispositif de réaction de résolveur désigné.	
1423 + (n-1)x50	Facultatif - RS	Set		Tension d'excitation du résolveur de réaction n	REAL	Définit la tension d'excitation sinusoïdale appliquée au rotor du dispositif de réaction de résolveur désigné.	Volts (eff)
1424 + (n-1)x50	Facultatif - RS	Set		Fréquence d'excitation du résolveur de réaction n	REAL	Fréquence du signal d'excitation sinusoïdal appliqué au dispositif de réaction de résolveur désigné. La gamme ou les valeurs de fréquences valides de cet attribut dépendent de l'interface matérielle du dispositif spécifique.	Hertz
1425 + (n-1)x50	Facultatif - RS	Set		Équilibre du câble du résolveur de réaction n	REAL	Ajuste l'amplitude relative des signaux Sinus et Cosinus provenant du résolveur afin de compenser l'impact du câble du résolveur.	%
1426 + (n-1)x50	Obligatoire - LT	Set		Type LDT n de réaction	USINT	Détermine le type LDT. Les options sont Start/Stop et MID. Les transducteurs Start/Stop acceptent un signal d'entrée (interrogation) pour lancer le cycle de mesure et répondre par deux impulsions sur la ligne de retour. La temporisation peut reposer sur le front de montée ou de descente. La durée entre les impulsions est proportionnelle à la position. Les transducteurs MID répondent au signal d'interrogation par une seule impulsion longue sur la ligne de retour. La largeur d'impulsion est proportionnelle à la position.	Énumération: 0 = MID 1 = Start/stop (Démarrer/Arrêter) la montée 2 = Start/Stop (Démarrer/Arrêter) la descente
1427 + (n-1)x50	Obligatoire - LT	Set		Recirculations LDT n de réaction	USINT	Détermine le nombre de recirculations associées à un transducteur LDT de type MID. Plusieurs recirculations peuvent être utilisées pour augmenter la résolution du LDT au détriment de	Nombre de recirculations

ID attr	Besoin dans la mise en œuvre	Règle d'accès	N V	Nom de l'attribut	Type de données	Description d'attribut	Sémantique des valeurs
						l'augmentation de la période d'échantillonnage.	
1428 + (n-1)x50	Obligatoire – HS, TP, HI, E21, E22, SS, RS (moteurs à aimant permanent uniquement)	Set		Décalage de commutation n de réaction	REAL	Valeur qui spécifie le décalage de commutation du dispositif de réaction monté sur un moteur à aimant permanent, en unités de degrés électriques. L'attribut spécifie le décalage par rapport à une position de référence de commutation définie en appliquant le courant c.c. à la borne U et hors des bornes V et W court-circuitées du moteur et en permettant au rotor de se déplacer vers sa position nulle magnétique par rapport au stator. Sur un codeur ou résolveur absolu, le décalage est la différence entre la position absolue nulle du dispositif et la position de référence de commutation. Sur un codeur incrémentiel doté de capteurs Hall, le décalage est la différence entre la position correspondante d'une transition de la voie Hall S3 (les voies S1 et S2 étant respectivement hautes et basses) et la position de référence de commutation. Le décalage de commutation s'applique uniquement au moteur monté sur dispositif Réaction 1.	Degrés électriques
1434 + (n-1)x50	Facultatif	Set		Bande passante de filtre de vitesse n de réaction.	REAL	Contrôle la bande passante du filtre passe-bas appliqué au signal de vitesse brut provenant de Réaction n. Une valeur nulle désactive cette fonction.	rad/s
1435 + (n-1)x50	Facultatif	Set		Bande passante de filtre d'accélération n de réaction.	REAL	Contrôle la bande passante du filtre passe-bas appliqué au signal d'accélération brut provenant de Réaction n. Une valeur nulle désactive cette fonction.	rad/s

7.3.4.2 Sémantique: Attribut N° 82 – Configuration de réaction

Cet attribut contient un champ Sélection de réaction énuméré de 4 bits qui détermine la manière dont sont utilisées les différentes voies de réaction logiques afin de mettre en œuvre le Mode de commande sélectionné pour cette instance d'axe (voir Figure 49).

← OCTET de 8 bits →	
Format du bit Configuration de réaction	
Bit 4-7: (Réservé)	Bits 0-3: Énumération Sélection de réaction

Figure 49 – Champ du bit Configuration de réaction

Les énumérations Sélection de réaction (voir Tableau 38) fournissent un support pour la fonctionnalité de commande multi-réaction pour les différents Modes de commande actifs, c'est-à-dire lorsque le dispositif commande activement le moteur en fonction d'une réaction. Dans ces Modes de commande actifs, la voie logique Réaction 1 est supposée être directement liée au moteur, Réaction 2 étant liée en aval de la transmission mécanique. Les signaux de commutation d'un moteur à aimant permanent sont toujours déduits de Réaction 1.

Tableau 38 – Définitions de l'énumération du champ Sélection de réaction

Énum.	Oblig. Facult.	Nom	Description
0	R/NF O/C	Pas de réaction	Pas de réaction est sélectionné si un contrôle de boucle fermée ou de boucle ouverte est choisi. Lors de l'exécution d'un contrôle de boucle ouverte, aucun signal de réaction n'est requis. Dans le contrôle de boucle fermée, le signal de réaction requis est estimé par un algorithme de commande sans capteur reposant sur les signaux de tension et de courant de phase du moteur.
1	O/N	Réaction principale	La Réaction principale attribue une voie de réaction non engagée, telle que spécifiée par l'attribut Sélection de réaction principale, à cette instance d'axe de mouvement, pour faire office de source de "réaction principale" lorsque le dispositif est configuré pour le mode Pas d'asservissement.
2	O/C	Réaction du moteur	Si Réaction du moteur est sélectionné, les signaux de réaction d'accélération, de vitesse et de position de commutation sont tous déduits de Réaction 1 montée sur le moteur.
3	F/C	Réaction de charge	Si Réaction de charge est sélectionné, Réaction 1 montée sur le moteur est uniquement utilisée pour la commutation du moteur à aimant permanent, Réaction 2 en aval étant utilisée pour la position, la vitesse et l'accélération.
4	O/P	Réaction double	Si Réaction double est sélectionné, Réaction 1 montée sur le moteur est utilisée pour la commutation, l'accélération et la vitesse, Réaction 2 en aval étant utilisée exclusivement pour la position.
5 à 255	-	(Réservé)	-

7.3.5 Attributs Capture d'événement

7.3.5.1 Généralités

Le Tableau 39 contient les attributs d'événement associés à une instance Objet Axe de mouvement. Il contient les événements d'enregistrement, de marquage et de retour à la position de référence. Les attributs Capture d'événement sont conçus pour prendre en charge jusqu'à 16 événements actifs éventuels par période d'actualisation du contrôleur. Le format de tous les attributs Datation est le Temps système absolu et suit la spécification CIP Sync, la valeur 0 correspondant au 1^{er} janvier 1970.

L'Objet Axe de mouvement prend en charge deux voies d'entrée d'enregistrement indépendantes par instance d'axe, qui peuvent être déclenchées sur les fronts de montée ou de descente du signal. Si la mise en œuvre matérielle du dispositif le permet, les données d'heure et de position d'événement peuvent être capturées simultanément pour quatre conditions d'événement. Les attributs Capture d'événement prennent également en charge le réarmement automatique pour les événements d'enregistrement. Cela permet au contrôleur de mettre en œuvre des fonctions importantes telles que l'Enregistrement en mosaïque et la Reconnaissance de modèle d'enregistrement.

L'Objet Axe de mouvement prend également en charge les événements Interrupteur d'origine, Marqueur et Marqueur d'interrupteur de la fonctionnalité de retour à la position de référence axe par axe. Les événements Marqueur sont souvent générés par le dispositif de retour en position configuré pour l'instance d'axe associée.

Set* → Ces attributs sont en général mis à jour par l'intermédiaire de l'Ensemble de données de consigne cycliques de la connexion Contrôleur - Dispositif (C-D) CIP Motion. S'ils sont inclus sous la forme de données de consigne cycliques, il convient de ne pas actualiser ces attributs via un service Set.

Tableau 39 – Attributs d'événement

ID attr	Besoin dans la mise en œuvre	Règle d'accès	N V	Nom de l'attribut	Type de données	Description d'attribut	Sémantique des valeurs
256	Facultatif – NP	Set*		Commande de vérification d'événement	DWORD	Cet attribut est transmis au dispositif par le contrôleur dans le cadre d'une connexion Dispositif - Contrôleur (D-C) afin d'armer les différentes entrées de dispositif (les entrées de marqueur, d'interrupteur d'origine et d'enregistrement, par exemple) afin de générer des événements pour le contrôleur. Lorsque ces événements sont activés, le dispositif capture l'heure et la position exacte de l'axe lors du déroulement de l'événement.	Voir la sémantique en 7.3.5.2.1
257	Facultatif – NP	Get		Statut de vérification d'événement	DWORD	Cet attribut est transmis au dispositif par le contrôleur dans le cadre d'une connexion Dispositif - Contrôleur (D-C) afin d'indiquer si le dispositif vérifie réellement les événements en fonction des entrées de dispositif (les entrées de marqueur, d'interrupteur d'origine et d'enregistrement, par exemple). La vérification d'événement est initiée lorsque le bit Commande de vérification d'événement correspondant est défini dans la connexion Contrôleur - Dispositif (C-D).	Voir la sémantique en 7.3.5.2.2
258	Obligatoire – NP	Get		Position du front positif de l'enregistrement 1	DINT	Position de retour verrouillée sur le front de montée de l'Entrée d'enregistrement 1.	Unités d'asservissement de position
259	Obligatoire – NP	Get		Position du front négatif de l'enregistrement 1	DINT	Position de retour verrouillée sur le front de descente de l'Entrée d'enregistrement 1.	Unités d'asservissement de position
260	Facultatif – NP	Get		Position du front positif de l'enregistrement 2	DINT	Position de retour verrouillée sur le front de montée de l'Entrée d'enregistrement 2.	Unités d'asservissement de position
261	Facultatif – NP	Get		Position du front négatif de l'enregistrement 2	DINT	Position de retour verrouillée sur le front de descente de l'Entrée d'enregistrement 2.	Unités d'asservissement de position
262	Facultatif – NP	Get		Durée du front positif de l'enregistrement 1	LINT	Temps système reposant sur le front de montée de l'Entrée d'enregistrement 1.	Nanosecondes (CIP Sync absolu)
263	Obligatoire – NP	Get		Durée du front négatif de l'enregistrement 1	LINT	Temps système reposant sur le front de descente de l'Entrée d'enregistrement 1.	Nanosecondes (CIP Sync absolu)
264	Obligatoire – NP	Get		Durée du front positif de l'enregistrement 2	LINT	Temps système reposant sur le front de montée de l'Entrée d'enregistrement 2.	Nanosecondes (CIP Sync absolu)

ID attr	Besoin dans la mise en œuvre	Règle d'accès	N V	Nom de l'attribut	Type de données	Description d'attribut	Sémantique des valeurs
265	Facultatif – NP	Get		Durée du front négatif de l'enregistrement 2	LINT	Temps système reposant sur le front de descente de l'Entrée d'enregistrement 2.	Nanosecondes (CIP Sync absolu)
266	Obligatoire – NP	Get		Position d'événement d'origine	DINT	Position de retour verrouillée sur l'événement de référence spécifié.	Unités d'asservissement de position
268	Facultatif – NP	Get		Heure d'événement d'origine	LINT	Temps système verrouillé sur l'événement de référence spécifié.	Nanosecondes (CIP Sync absolu)

7.3.5.2 Sémantique

7.3.5.2.1 Attribut N° 256 – Commande de vérification d'événement

Les 28 premiers bits de cet attribut de 32 bits (voir Figure 50) sont utilisés pour activer les différentes entrées du dispositif (les entrées de marqueur, d'interrupteur d'origine et d'enregistrement, par exemple) afin de générer des événements pour le contrôleur. Lorsque ces événements sont activés, le dispositif capture l'heure et la position exacte de l'instance d'axe de mouvement associée. Les 4 bits de poids fort représentent le nombre de messages Acquiescement d'événement présents dans le bloc de données d'événement de la dernière actualisation de la connexion Contrôleur - Dispositif (C-D). Pour une description détaillée du protocole de gestion des événements entre le dispositif d'entraînement et le contrôleur, voir la section Connexion du profil de dispositif CIP Motion approprié (voir 6.4.3.4).

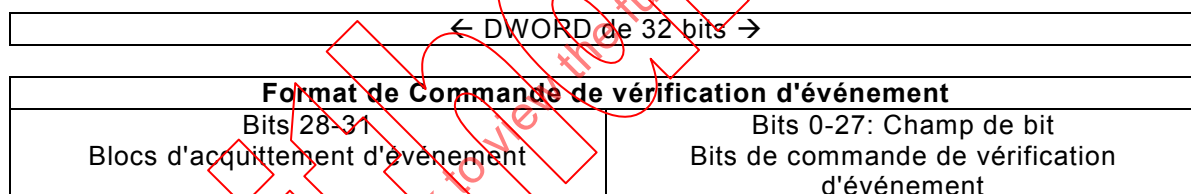


Figure 50 – Champ de mot Commande de vérification d'événement

Pour les événements Marqueur - Interrupteur d'origine spécifiés dans le Tableau 40, le dispositif recherche en premier lieu le niveau de l'entrée d'interrupteur d'origine par rapport à la transition en fonction du symbole + ou –, puis recherche immédiatement la transition de l'entrée de marqueur en fonction du deuxième symbole + ou –.

Tableau 40 – Définitions de bit Commande de vérification d'événement

Bit	Oblig. Facult.	Nom	Description
0	O/P	Front pos Enreg 1	Activer la vérification pour la transition de front positif de l'entrée Enregistrement 1
1	O/P	Front nég Enreg 1	Activer la vérification pour la transition de front négatif de l'entrée Enregistrement 1
2	F/P	Front pos Enreg 2	Activer la vérification pour la transition de front positif de l'entrée Enregistrement 2
3	F/P	Front nég Enreg 2	Activer la vérification pour la transition de front négatif de l'entrée Enregistrement 2
4-7		(Réservé)	-
8	F/P	Réarm auto pos Enreg 1	Activer le mécanisme de réarmement automatique après la transition de front positif de l'entrée Enregistrement 1
9	F/P	Réarm auto nég Enreg 1	Activer le mécanisme de réarmement automatique

Bit	Oblig. Facult.	Nom	Description
			après la transition de front négatif de l'entrée Enregistrement 1
10	F/P	Réarm auto pos Enreg 2	Activer le mécanisme de réarmement automatique après la transition de front positif de l'entrée Enregistrement 2
11	F/P	Réarm auto nég Enreg 2	Activer le mécanisme de réarmement automatique après la transition de front négatif de l'entrée Enregistrement 2
12 à 15		(Réservé)	
16	O/P	Front pos du marqueur	Activer la vérification du front positif de l'entrée de marqueur de la voie de retour en position associée.
17	O/P	Front nég du marqueur	Activer la vérification du front négatif de l'entrée de marqueur de la voie de retour en position associée.
18	O/P	Front pos de l'interrupteur d'origine	Activer la vérification du front positif de l'entrée d'origine associée à cette instance d'axe.
19	O/P	Front nég de l'interrupteur d'origine	Activer la vérification du front négatif de l'entrée d'origine associée à cette instance d'axe.
20	O/P	Marqueur - Interrupteur d'origine + +	Activer la vérification de la séquence d'événements spécifiée en tant que transition de front positif de l'entrée Interrupteur d'origine suivie d'un front positif de l'entrée Marqueur.
21	O/P	Marqueur - Interrupteur d'origine + -	Activer la vérification de la séquence d'événements spécifiée en tant que transition de front positif de l'entrée Interrupteur d'origine suivie d'un front négatif de l'entrée Marqueur.
22	O/P	Marqueur - Interrupteur d'origine - +	Activer la vérification de la séquence d'événements spécifiée en tant que transition de front négatif de l'entrée Interrupteur d'origine suivie d'un front positif de l'entrée Marqueur.
23	O/P	Marqueur - Interrupteur d'origine - -	Activer la vérification de la séquence d'événements spécifiée en tant que transition de front négatif de l'entrée Interrupteur d'origine suivie d'un front négatif de l'entrée Marqueur.
24 à 27		(Réservé)	

7.3.5.2.2 Attribut N° 257 – Statut de vérification d'événement

Les 28 premiers bits de cet attribut de 32 bits (voir Figure 51) sont utilisés pour indiquer si le dispositif contrôle les événements en fonction des différentes entrées du dispositif, telles que les entrées de marqueur et d'enregistrement, par exemple (voir Tableau 41). La vérification d'événement est initiée lorsque le bit de commande de vérification d'événement correspondant est défini dans la connexion Contrôleur - Dispositif (C-D). Lorsqu'un événement se produit, le dispositif capture l'heure et la position exacte de l'axe, puis transmet ces informations au contrôleur dans des blocs de données d'événement. Mais pour que le contrôleur puisse traiter les données d'événement, le bit de Statut de vérification d'événement correspondant doit être défini. Les 4 bits de poids fort de cet attribut représentent le nombre de messages Notification d'événement présents dans le bloc de données d'événement de la dernière actualisation de la connexion Dispositif - Contrôleur (D-C). Pour une description détaillée du protocole de gestion des événements entre le dispositif et le contrôleur, voir la section Connexion (voir 6.4.4.4).

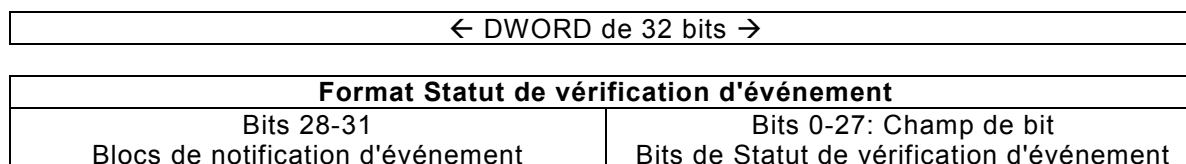


Figure 51 – Champ de mot Statut de vérification d'événement

Tableau 41 – Définitions de bit Statut de vérification d'événement

Bit	Oblig. Facult.	Nom	Description
0	O/P	Front pos Enreg 1	Vérification de la transition de front positif de l'entrée Enregistrement 1 ?
1	O/P	Front nég Enreg 1	Vérification de la transition de front négatif de l'entrée Enregistrement 1 ?
2	F/P	Front pos Enreg 2	Vérification de la transition de front positif de l'entrée Enregistrement 2 ?
3	F/P	Front nég Enreg 2	Vérification de la transition de front négatif de l'entrée Enregistrement 2 ?
4 à 7		(Réservé)	-
8	F/P	Réarm auto pos Enreg 1	Mécanisme de réarmement automatique après la transition de front positif de l'entrée Enregistrement 1 actif ?
9	F/P	Réarm auto nég Enreg 1	Mécanisme de réarmement automatique après la transition de front négatif de l'entrée Enregistrement 1 actif ?
10	F/P	Réarm auto pos Enreg 2	Mécanisme de réarmement automatique après la transition de front positif de l'entrée Enregistrement 2 actif ?
11	F/P	Réarm auto nég Enreg 2	Mécanisme de réarmement automatique après la transition de front négatif de l'entrée Enregistrement 2 actif ?
12 à 15		(Réservé)	
16	O/P	Front pos du marqueur	Vérification du front positif de l'entrée de marqueur de la voie de retour en position associée ?
17	O/P	Front nég du marqueur	Vérification du front négatif de l'entrée de marqueur de la voie de retour en position associée ?
18	O/P	Front pos de l'interrupteur d'origine	Vérification du front positif de l'entrée d'origine associée à cette instance d'axe ?
19	O/P	Front nég de l'interrupteur d'origine	Vérification du front négatif de l'entrée d'origine associée à cette instance d'axe ?
20	O/P	Marqueur - Interrupteur d'origine + +	Vérification de la séquence d'événements spécifiée en tant que transition de front positif de l'entrée Interrupteur d'origine suivie d'un front positif de l'entrée Marqueur ?
21	O/P	Marqueur - Interrupteur d'origine + -	Vérification de la séquence d'événements spécifiée en tant que transition de front positif de l'entrée Interrupteur d'origine suivie d'un front négatif de l'entrée Marqueur ?
22	O/P	Marqueur - Interrupteur d'origine - +	Vérification de la séquence d'événements spécifiée en tant que transition de front négatif de l'entrée Interrupteur d'origine suivie d'un front positif de l'entrée Marqueur ?
23	O/P	Marqueur - Interrupteur d'origine - -	Vérification de la séquence d'événements spécifiée en tant que transition de front négatif de l'entrée Interrupteur d'origine suivie d'un front négatif de l'entrée Marqueur ?
24 à 27		(Réservé)	

7.3.6 Attributs de génération de référence de consigne

7.3.6.1 Généralités

Les listes d'attributs du Tableau 42 et du Tableau 43 s'appliquent à la fonctionnalité de génération de référence de consigne du dispositif qui convertit la position, la vitesse et l'accélération de consigne, ainsi que les données de couple provenant d'un planificateur basé sur contrôleur, en signaux de références de consigne correspondants adressés aux structures

de commande de moteur du dispositif. La fonctionnalité de génération de référence de consigne inclut des interpolateurs fins, des interrupteurs de signal et des limiteurs dynamiques.

Set* → Ces attributs sont en général mis à jour par l'intermédiaire de l'Ensemble de données de consigne cycliques de la connexion Contrôleur - Dispositif (C-D) CIP Motion. S'ils sont inclus sous la forme de données de consigne cycliques, il convient de ne pas actualiser ces attributs via un service Set.

Tableau 42 – Attributs Signal du générateur de consigne

ID attr	Besoin dans la mise en œuvre	Règle d'accès	N V	Nom de l'attribut	Type de données	Description d'attribut	Sémantique des valeurs
93	Obligatoire – P Facultatif – VT	Get		Temps de consigne cible	LINT	Datation associée aux données de consigne provenant du planificateur de mouvement et utilisée par les interpolateurs fins de commande. Le temps de consigne cible est la somme de la valeur Datation du contrôleur et de l'intervalle Actualisation cible de consigne associée à cette instance d'axe. Le format de datation est absolu et suit la norme CIP Sync. 0 correspondant au 1 ^{er} janvier 1970.	Nano-secondes (CIP Sync absolu)
286	Obligatoire – P	Set*		Consigne de position du contrôleur	DINT	Valeur des données de position de consigne provenant du contrôleur, actualisée au cours de la période d'actualisation du contrôleur, et alimentant un interpolateur fin.	Unités d'asservissement de position
287	Obligatoire – Facultatif – P	Set*		Consigne de vitesse du contrôleur	REAL	Valeur des données de vitesse de consigne provenant du contrôleur, actualisée au cours de la période d'actualisation du contrôleur, et alimentant un interpolateur fin.	Unités de commande de vitesse/s
288	Facultatif – PVT	Set*		Consigne d'accélération du contrôleur	REAL	Valeur des données d'accélération de consigne provenant du contrôleur, actualisée au cours de la période d'actualisation du contrôleur, et alimentant un interpolateur fin.	Unités de commande d'accélération/s ²
289	Facultatif – PVT	Set*		Consigne de couple du contrôleur	REAL	Valeur des données de couple commandé provenant du contrôleur, actualisée au cours de la période d'actualisation du contrôleur, et alimentant un interpolateur fin.	% de la valeur assignée du moteur
290	Facultatif – P	Get		Position de consigne interpolée	DINT	Valeur de sortie provenant de l'interpolateur fin de position de consigne.	Unités d'asservissement de position
291	Facultatif – PV	Get		Vitesse de consigne interpolée	REAL	Valeur de sortie provenant de l'interpolateur fin de vitesse de consigne. En l'absence de signal Vitesse de consigne lors de l'exécution d'un asservissement de position, ce signal peut être déduit en ajustant la valeur de sortie Position différentielle de l'interpolateur fin de position de consigne.	Unités de commande de vitesse/s

ID attr	Besoin dans la mise en œuvre	Règle d'accès	N V	Nom de l'attribut	Type de données	Description d'attribut	Sémantique des valeurs
292	Facultatif - PVT	Get		Accélération de consigne interpolée	REAL	Valeur de sortie provenant de l'interpolateur fin d'accélération de consigne. En l'absence de signal Accélération de consigne lors de l'exécution d'un asservissement de position ou de vitesse, ce signal peut être déduit en ajustant la valeur de sortie Vitesse différentielle de l'interpolateur fin de vitesse de consigne. En l'absence de signal Vitesse de consigne, le signal Accélération de consigne interpolée peut être déduit en ajustant la deuxième valeur de sortie Position différentielle de l'interpolateur fin de position de consigne.	Unités de commande d'accélération/s ²

Tableau 43 – Attributs Configuration du générateur de consigne

ID attr	Besoin dans la mise en œuvre	Règle d'accès	N V	Nom de l'attribut	Type de données	Description d'attribut	Sémantique des valeurs
91	Obligatoire – PVT	Set*		Ensemble de données de consigne	OCTET	Champ matriciel qui détermine les valeurs de données de consigne actualisées par le contrôleur.	Champ de bit: 0 = Consigne de position (R/P) 1 = Consigne de vitesse (R/V) 2 = Consigne d'accélération (O) 3 = Consigne de couple (R/T) 4 = Compensation de position (O) 5 = Compensation d'accélération (O) 6 = Compensation de vitesse (O) 7 = Compensation de couple (O)
92	Obligatoire – PV	Set*		Commande d'interpolation	OCTET	Commande la méthode d'interpolation/extrapolation appliquée aux données de consigne de position, de vitesse et d'accélération provenant du planificateur de mouvement en fonction de la datation associée.	Voir la sémantique en 7.3.6.2
302	Obligatoire – V Facultatif – P	Set		Limite de vitesse - Positive	REAL	Cette valeur définit la consigne de vitesse positive admise dans la jonction de somme des vitesses. Si la limite de vitesse est dépassée, le dispositif répond en calant la consigne de vitesse à cette limite et en définissant le bit de statut Limite de vitesse.	Unités de commande de vitesse/s
303	Obligatoire – V Facultatif – P	Set		Limite de vitesse - Négative	REAL	Cette valeur définit la consigne de vitesse négative admise dans la jonction de somme des vitesses. Si la limite de vitesse est dépassée, le dispositif répond en calant la consigne de vitesse à cette limite et en définissant le bit de statut	Unités de commande de vitesse/s

ID attr	Besoin dans la mise en œuvre	Règle d'accès	N V	Nom de l'attribut	Type de données	Description d'attribut	Sémantique des valeurs
						Limite de vitesse.	
304	Obligatoire – V Facultatif – P	Set		Limite d'accélération	REAL	Cette valeur définit le taux de variation positif maximal admis de l'amplitude de la consigne de vitesse (en augmentant la vitesse). Si la limite d'accélération est dépassée, le dispositif répond en calant le taux de variation de la consigne de vitesse à cette limite et en définissant le bit de statut Limite d'accélération.	Unités de commande de vitesse/s ²
305	Obligatoire – V Facultatif – P	Set		Limite de décélération	REAL	Cette valeur définit le taux de variation négatif maximal admis de l'amplitude de la consigne de vitesse (en réduisant la vitesse). Si la limite de décélération est dépassée, le dispositif répond en calant le taux de variation de la consigne de vitesse à cette limite et en définissant le bit de statut Limite de décélération.	Unités de commande de vitesse/s ²
306	Facultatif – PV	Set		Commande de limite d'à-coups	REAL	Si l'accélération ou la décélération est limitée, cette valeur définit le pourcentage de temps d'accélération ou de décélération appliqué au ralentisseur de vitesse sous la forme d'une courbe S d'à-coups limités. Le temps de Courbe S est ajouté à la première et à la deuxième moitié de la rampe. Une valeur nulle ne donne lieu à aucune Courbe en S, c'est-à-dire une rampe d'accélération linéaire. Une valeur de 100 % donne lieu à un profil d'accélération triangulaire dont la crête est la limite accélération / décélération. La valeur Limite d'à-coups déduite diminue au fur et à mesure de l'augmentation de la valeur Commande de limite d'à-coups.	%
307	Facultatif – PV	Set		Activation de l'amorçage instantané	BOOL	Cette valeur booléenne permet d'activer ou de désactiver la fonction Amorçage instantané du dispositif. Si la valeur "true" est affectée à Amorçage instantané activé, et que l'axe de mouvement est désactivé, le dispositif détermine la vitesse en cours du moteur dans le cadre des activités d'initialisation Statut du démarrage. Juste avant de passer à l'état Exécution, le dispositif prédéfinit la sortie de la Limite d'accélération sur la vitesse en cours. De cette manière, le moteur passe en continu de sa vitesse en cours au profil de vitesse commandée généré par le planificateur de mouvement. Si la valeur "false" est affectée à Amorçage instantané activé, la	False = Amorçage instantané désactivé True = Amorçage instantané activé

ID attr	Besoin dans la mise en œuvre	Règle d'accès	N V	Nom de l'attribut	Type de données	Description d'attribut	Sémantique des valeurs
						vitesse du moteur n'est pas déterminée et une valeur nulle est affectée à la sortie Limite d'accélération.	
308	Facultatif - F	Set		Vitesse de saut 1	REAL	Définit la vitesse centrale d'une bande de vitesse de saut dans laquelle le dispositif ne fonctionne pas.	Unités de commande de vitesse/s
309	Facultatif - F	Set		Vitesse de saut 2	REAL	Définit la vitesse centrale d'une bande de vitesse de saut dans laquelle le dispositif ne fonctionne pas.	Unités de commande de vitesse/s
310	Facultatif - F	Set		Vitesse de saut 3	REAL	Définit la vitesse centrale d'une bande de vitesse de saut dans laquelle le dispositif ne fonctionne pas.	Unités de commande de vitesse/s
311	Facultatif - F	Set		Bande de vitesse de saut	REAL	Détermine la fenêtre de vitesse qui ne peut pas être commandée. Un point de consigne de commande dans cette fenêtre est ajusté par le bloc Vitesse de saut de manière à atteindre la valeur limite Bande de vitesse de saut supérieure ou inférieure. L'axe peut accélérer ou décélérer doucement sur la bande de vitesse de saut en fonction du bloc de limite d'accélération, mais il ne peut pas fonctionner à une vitesse définie dans la bande. La Bande de vitesse de saut est répartie à moitié au-dessus et à moitié au-dessous de la vitesse de saut. L'attribut Bande de vitesse de saut s'applique à toutes les vitesses de saut prises en charge dans le dispositif. Une valeur nulle désactive cette fonction.	Unités de commande de vitesse/s

7.3.6.2 Sémantique: Attribut N° 92 – Commande d'interpolation

L'attribut Commande d'interpolation régit la méthode d'interpolation/extrapolation appliquée aux données de consigne de position, de vitesse et d'accélération provenant du planificateur de mouvement en fonction de la datation associée (voir Figure 52). Les deux premiers bits de cet attribut de 8 bits représentent l'Actualisation cible de consigne, qui détermine le nombre de périodes d'actualisation ajouté à la datation des données de consigne afin de déterminer le Temps absolu auquel est destinée la valeur de données de consigne, c'est-à-dire le temps de consigne cible (voir Tableau 44). Pour plus de détails sur la commande d'interpolation et d'extrapolation, voir 7.6.6.2 (Interpolation fine de consigne).

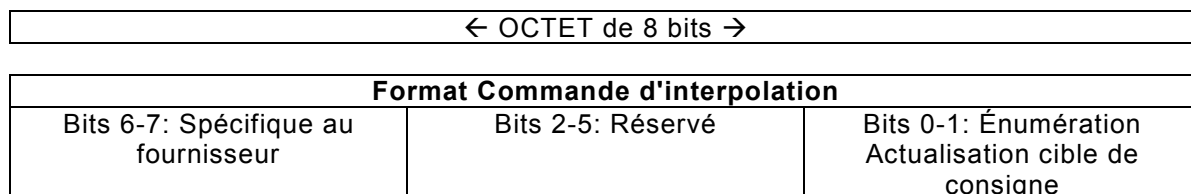


Figure 52 – Champ de mot Commande d'interpolation

Tableau 44 – Définition de l'énumération Actualisation cible de consigne

Énum.	Oblig. Facult.	Nom	Description
0	O/P	Immédiat	Une Actualisation cible de consigne nulle implique que les données de consigne doivent être immédiatement appliquées à la structure de commande.
1	O/P	Extrapoler (+1 UP)	Une Actualisation cible de consigne de 1 implique que l'extrapolation doit être utilisée pour appliquer les données de consigne à la structure de commande en ajoutant 1 période d'actualisation à la datation de la commande.
2	O/P	Interpoler (+2 UP)	Une Actualisation cible de consigne de 2 implique que l'interpolation fine doit être utilisée pour appliquer les données de consigne en ajoutant 2 périodes d'actualisation à la datation de la commande.
3		(Réservé)	-

7.3.7 Attributs du mode de commande

7.3.7.1 Attributs Boucle de position

Le Tableau 45 et le Tableau 46 contiennent tous les attributs d'asservissement de position associés à une instance Objet Axe de mouvement.

Set* → Cet attribut est en général mis à jour par l'intermédiaire de l'Ensemble de données de consigne cycliques de la connexion Contrôleur - Dispositif (C-D) CIP Motion. S'il est inclus sous la forme de données de consigne cycliques, il convient de ne pas l'actualiser via un service Set.

Tableau 45 – Attributs Signal de boucle de position

ID attr	Besoin dans la mise en œuvre	Règle d'accès	N V	Nom de l'attribut	Type de données	Description d'attribut	Sémantique des valeurs
420	Obligatoire – P	Get		Consigne de position	DINT	Sortie de position de consigne provenant de l'interpolateur fin (s'il est actif) dans la boucle de position s'il est configuré pour le contrôle de boucle de position.	Unités d'asservissement de position
421	Obligatoire – P	Set*		Compensation de position	DINT	Consigne de position supplémentaire ajoutée à la Consigne de position afin de générer le signal Référence de position dans la jonction de somme de boucle de position.	Unités d'asservissement de position
422	Obligatoire – P	Get		Référence de position	DINT	Signal de référence de position de consigne dans la jonction de somme de boucle de position à comparer à un signal de retour en position.	Unités d'asservissement de position
423	Obligatoire – PC	Get		Consigne d'action anticipatrice de vitesse	REAL	Référence d'action anticipatrice de vitesse dans une jonction de somme de boucle de vitesse.	Unités de commande de vitesse/s
424	Obligatoire – NPVC Facultatif – T	Get		Retour en position	DINT	Valeur de retour en position canalisé dans la jonction de somme d'asservissement de position.	Unités d'asservissement de position

ID attr	Besoin dans la mise en œuvre	Règle d'accès	N V	Nom de l'attribut	Type de données	Description d'attribut	Sémantique des valeurs
426	Obliga- toire – PC	Get		Erreur de position	REAL	Erreur entre la position commandée et la position réelle qui est la sortie de la jonction de somme de boucle de position.	Unités d'asservissement de position
427	Obliga- toire – PC	Get		Sortie de l'intégrateur de position	REAL	Sortie de l'intégrateur de position représentant la contribution de l'intégrateur de position à la Sortie de la boucle de position.	Unités de commande de vitesse/s
428	Obliga- toire – PC	Get		Sortie de la boucle de position	REAL	Sortie de la chaîne d'action de boucle de position représentant la mesure de contrôle totale de la boucle de position.	Unités de commande de vitesse/s

Tableau 46 – Attributs Configuration de boucle de position

ID attr	Besoin dans la mise en œuvre	Règle d'accès	N V	Nom de l'attribut	Type de données	Description d'attribut	Sémantique des valeurs
430	Obliga- toire – PC	Set		Kvff	REAL	Valeur de Gain d'action anticipatrice de vitesse qui multiplie le signal Consigne d'anticipation de vitesse avant de l'appliquer à la jonction de somme de boucle de vitesse.	%
431	Obliga- toire – PC	Set		Kpp	REAL	Valeur Gain proportionnel de position qui multiplie le signal Erreur de position avant de l'appliquer à la jonction de somme de boucle de vitesse.	rad/s
432	Obliga- toire – PC	Set		Kpi	REAL	Valeur Gain intégral de position qui multiplie le signal Erreur d'intégration de position avant de l'appliquer à la jonction de somme de boucle de vitesse.	rad/s ²
433	Obliga- toire – PC	Set		Tolérance de verrouillage de position	REAL	Cette valeur établit une fenêtre autour de la position de consigne en cours. Si la position réelle se trouve dans cette fenêtre, le bit de statut Verrouillage de position est défini. Si la position réelle ne se trouve pas dans cette fenêtre, le bit de statut Verrouillage de position est supprimé.	Unités d'asservissement de position
434	Obliga- toire – PC	Set		Tolérance d'erreur de position	REAL	Détermine la valeur Erreur de position maximale absolue qui peut être tolérée sans générer une exception Erreur de position excessive.	Unités d'asservissement de position
435	Facultatif – PVC	Set		Temps de tolérance d'erreur de position	REAL	Détermine la durée maximale de dépassement de la Tolérance d'erreur de position sans générer une exception.	Secondes
439	Obliga- toire – PC	Set		Commande d'intégrateur de position	OCTET	Ce champ de bit contrôle le comportement de l'intégrateur de boucle de position tout en commandant le mouvement par	Champ de bit: 0: Activation du maintien

ID attr	Besoin dans la mise en œuvre	Règle d'accès	N V	Nom de l'attribut	Type de données	Description d'attribut	Sémantique des valeurs
						l'intermédiaire du contrôleur. Si le bit d'activation du maintien de l'intégrateur est défini, l'intégrateur est maintenu pendant que le mouvement est commandé avec une vitesse non nulle. S'il est supprimé, l'intégrateur fonctionne sans qualification. Si le bit de prérégage automatique est défini, la valeur de préchargement de l'intégrateur est automatiquement chargée avec la consigne de vitesse en cours en présence d'une variation de mode de commande entre la commande de vitesse et la commande de position. S'il n'est pas défini, l'intégrateur est chargé avec la valeur de préchargement d'intégrateur de position configuré.	de l'intégrateur 1: Prérégage automatique
440	Obligatoire - PC	Set		Préchargement d'intégrateur de position	REAL	Valeur affectée à l'intégrateur de position lorsque la boucle d'asservissement de position est activée.	Unités de commande de vitesse/s

7.3.7.2 Attributs Boucle de vitesse

Le Tableau 47 et le Tableau 48 contiennent tous les attributs de commande de vitesse associés à une instance Objet Axe de mouvement.

Set* → Cet attribut est en général mis à jour par l'intermédiaire de l'Ensemble de données de consigne cycliques de la connexion Contrôleur - Dispositif (C-D) CIP Motion. S'il est inclus sous la forme de données de consigne cycliques, il convient de ne pas l'actualiser via un service Set.

Tableau 47 – Attributs Signal de boucle de vitesse

ID attr	Besoin dans la mise en œuvre	Règle d'accès	N V	Nom de l'attribut	Type de données	Description d'attribut	Sémantique des valeurs
450	Obligatoire – PV	Get		Consigne de vitesse	REAL	Sortie de position de consigne provenant de l'interpolateur fin (s'il est actif) dans la boucle de position s'il est configuré pour le contrôle de boucle de vitesse.	Unités de commande de vitesse/s
451	Obligatoire – PV	Set*		Compensation de vitesse	REAL	Consigne de vitesse supplémentaire ajoutée à la jonction de somme de boucle de vitesse.	Unités de commande de vitesse/s
452	Obligatoire – PVC	Get		Consigne d'anticipation d'accélération	REAL	Référence d'anticipation d'accélération dans une jonction de somme de boucle d'accélération.	Unités de commande de vitesse/s ²
453	Obligatoire – PV	Get		Référence de vitesse	REAL	Référence de vitesse de consigne dans une jonction de somme de boucle de vitesse.	Unités de commande de vitesse/s

ID attr	Besoin dans la mise en œuvre	Règle d'accès	N V	Nom de l'attribut	Type de données	Description d'attribut	Sémantique des valeurs
454	Obliga- toire – NPVC Facultatif – T	Get		Retour en vitesse	REAL	Vitesse réelle de l'axe appliquée à la jonction de somme de vitesse en fonction de la sélection Mode de commande.	Unités de commande de vitesse/s
455	Obliga- toire – PVC	Get		Erreur de vitesse	REAL	Erreur entre la référence de vitesse et la valeur de retour en vitesse qui est la sortie de la jonction de somme de boucle de vitesse.	Unités de commande de vitesse/s
456	Obliga- toire – PVC	Get		Sortie d'intégrateur de vitesse	REAL	Sortie de l'intégrateur de vitesse représentant la contribution de l'intégrateur de vitesse à la Sortie de la boucle de vitesse.	Unités de commande d'accélération/s ²
457	Obliga- toire – PVC	Get		Sortie de boucle de vitesse	REAL	Sortie de la chaîne d'action de vitesse représentant la mesure de contrôle totale de la boucle de vitesse.	Unités de commande d'accélération/s ²

Tableau 48 – Attributs Configuration de boucle de vitesse

ID attr	Besoin dans la mise en œuvre	Règle d'accès	N V	Nom de l'attribut	Type de données	Description d'attribut	Sémantique des valeurs
460	Obliga- toire – PC	Set		Kaff	REAL	Valeur Gain d'anticipation d'accélération qui multiplie le signal Consigne d'anticipation d'accélération avant de l'appliquer à la jonction de somme de boucle de vitesse.	%
461	Obliga- toire – PVC	Set		Kvp	REAL	Valeur Gain proportionnel de vitesse qui multiplie le signal Erreur de vitesse avant de l'appliquer à la jonction de somme d'accélération.	rad/s
462	Obliga- toire – PVC	Set		Kvi	REAL	Valeur Gain intégral de vitesse qui multiplie le signal Erreur d'intégrateur de vitesse avant de l'appliquer à la jonction de somme d'accélération.	rad/s ²
464	Obliga- toire – V Facultatif – P	Set		Kdr	REAL	Valeur Statisme de vitesse qui assure la conformité avec l'intégrateur de vitesse en soustrayant une partie de l'effort de boucle de vitesse de l'entrée d'erreur de vitesse à l'intégrateur de vitesse. La présence d'un gain d'échelle Couple/Force, Kj, dans le signal de statisme permet de spécifier le Statisme de vitesse dans les unités de vitesse par % de sortie de couple assigné. Ce paramètre est également valide pour les dispositifs d'entraînement V/Hz, et son comportement est pratiquement identique, mais au lieu d'associer le % assigné au couple, le % assigné est associé au courant.	Unités de commande vitesse/ % assigné
465	Facultatif – PVC	Set		Tolérance d'erreur de vitesse	REAL	Détermine la valeur Erreur de vitesse maximale absolue qui peut être tolérée sans générer	Unités de commande

ID attr	Besoin dans la mise en œuvre	Règle d'accès	N V	Nom de l'attribut	Type de données	Description d'attribut	Sémantique des valeurs
						une exception Erreur de vitesse excessive.	de vitesse/s
466	Facultatif – PVC	Set		Temps de tolérance d'erreur de vitesse	REAL	Détermine la durée maximale de dépassement de la Tolérance d'erreur de vitesse sans générer une exception.	Secondes
467	Obligatoire – PVC	Set		Commande d'intégrateur de vitesse	OCTET	Ce champ de bit contrôle le comportement de l'intégrateur de boucle de vitesse tout en commandant le mouvement par l'intermédiaire du contrôleur. Si le bit d'activation du maintien de l'intégrateur est défini, l'intégrateur est maintenu pendant que le mouvement est commandé avec une vitesse non nulle. S'il est supprimé, l'intégrateur fonctionne sans qualification. Si le bit de pré-régulation automatique est défini, la valeur de pré-chargement de l'intégrateur est automatiquement chargée avec la consigne de couple en cours en présence d'une variation de mode de commande entre la commande de couple et la commande de vitesse. S'il n'est pas défini, l'intégrateur est chargé avec la valeur de pré-chargement d'intégrateur de vitesse configuré.	Champ de bit: 0: Activation du maintien de l'intégrateur 1: Pré-régulation automatique
468	Obligatoire – PVC	Set		Pré-chargement d'intégrateur de vitesse	REAL	Valeur affectée à l'intégrateur de vitesse lorsque la boucle de commande de vitesse est activée.	Unités de commande d'accélération/s ²
469	Facultatif – PVC	Set		Bande passante du filtre passe-bas de vitesse	REAL	Contrôle la bande passante du filtre passe-bas appliqué au signal Erreur de vitesse. La mise en œuvre recommandée est composée d'un filtre IIR bipolaire. Une valeur nulle désactive cette fonction.	rad/s
470	Facultatif – PV	Set		Seuil de vitesse	REAL	Définit une vitesse absolue minimale en deçà de laquelle le bit Seuil de vitesse est défini.	Unités de commande de vitesse/s
471	Facultatif – PV	Set		Tolérance de verrouillage de vitesse	REAL	Cette valeur établit une fenêtre autour de la vitesse de consigne en cours. Si la vitesse réelle se trouve dans cette fenêtre, le bit de statut Verrouillage de vitesse est défini. Si la position réelle ne se trouve pas dans cette fenêtre, le bit de statut Verrouillage de vitesse est supprimé.	Unités de commande de vitesse/s
472	Obligatoire – PV	Set		Fenêtre de gel de vitesse	REAL	Cette valeur établit une fenêtre autour d'une vitesse nulle. Si la vitesse réelle se trouve dans cette fenêtre, le bit de statut Gel de vitesse est défini. Si la vitesse réelle ne se trouve pas dans cette fenêtre, le bit de statut Gel de vitesse est supprimé.	Unités de commande de vitesse/s

7.3.7.3 Attributs Commande d'accélération

Le Tableau 49 contient tous les attributs d'accélération associés à une instance Objet Axe de mouvement.

Set* → Cet attribut est en général mis à jour par l'intermédiaire de l'Ensemble de données de consigne cycliques de la connexion Contrôleur - Dispositif (C-D) CIP Motion. S'il est inclus sous la forme de données de consigne cycliques, il convient de ne pas l'actualiser via un service Set.

Tableau 49 – Attributs Signal d'accélération

ID attr	Besoin dans la mise en œuvre	Règle d'accès	N V	Nom de l'attribut	Type de données	Description d'attribut	Sémantique des valeurs
480	Facultatif - C	Get		Consigne d'accélération	REAL	Sortie d'accélération de consigne provenant de l'interpolateur fin (s'il est actif) dans la boucle d'accélération s'il est configuré pour la commande d'accélération.	Unités de commande d'accélération/s ²
481	Facultatif - C	Set*		Compensation d'accélération	REAL	Consigne d'accélération supplémentaire ajoutée à la jonction de somme de boucle d'accélération.	Unités de commande d'accélération/s ²
482	Facultatif - C	Get		Référence d'accélération	REAL	Référence d'accélération de consigne dans une jonction de somme de boucle d'accélération.	Unités de commande d'accélération/s ²
483	Obligatoire – NPVC Facultatif – T	Get		Retour en accélération	REAL	Accélération réelle de l'axe en fonction du dispositif de réaction sélectionné.	Unités de commande d'accélération/s ²

7.3.7.4 Attributs Référence de couple/force

Le Tableau 50 et le Tableau 51 contiennent tous les attributs de couple/force associés à une instance Objet Axe de mouvement.

Set* → Cet attribut est en général mis à jour par l'intermédiaire de l'Ensemble de données de consigne cycliques de la connexion Contrôleur - Dispositif (C-D) CIP Motion. S'il est inclus sous la forme de données de consigne cycliques, il convient de ne pas l'actualiser via un service Set.

Tableau 50 – Attributs Signal de référence de couple/force

ID attr	Besoin dans la mise en œuvre	Règle d'accès	N V	Nom de l'attribut	Type de données	Description d'attribut	Sémantique des valeurs
490	Obligatoire – C	Get		Consigne de couple	REAL	Sortie de couple de consigne provenant de l'interpolateur fin (s'il est actif) dans la jonction de somme d'entrée de couple s'il est configuré pour l'asservissement de couple.	% de la valeur assignée du moteur
491	Obligatoire – C	Set*		Compensation de couple	REAL	Consigne de couple supplémentaire ajoutée à la jonction de somme d'entrée de couple.	% de la valeur assignée du moteur
492	Obligatoire – C	Get		Référence de	REAL	Signal d'entrée de référence de couple commandé avant la	% de la valeur

ID attr	Besoin dans la mise en œuvre	Règle d'accès	N V	Nom de l'attribut	Type de données	Description d'attribut	Sémantique des valeurs
	toire – C			couple		section de filtre de couple, représentant la somme des entrées de signal Consigne de couple et Compensation de couple.	assignée du moteur
493	Obliga- toire – C	Get		Référence de couple filtrée	REAL	Signal d'entrée de référence de couple commandé après la section de filtre de couple.	% de la valeur assignée du moteur
494	Obliga- toire – C	Get		Référence de couple limitée	REAL	Signal d'entrée de référence de couple commandé après la section de limiteur de couple.	% de la valeur assignée du moteur

Tableau 51 – Attributs Configuration de référence de couple/force

ID attr	Besoin dans la mise en œuvre	Règle d'accès	N V	Nom de l'attribut	Type de données	Description d'attribut	Sémantique des valeurs
496	Obliga- toire – PVC	Set		Kj	REAL	Valeur Gain d'échelle couple/force qui convertit l'accélération commandée en couple/force assignés équivalent. Correctement définie, cette valeur représente l'inertie totale du système.	% de la valeur assignée du moteur/(Unités d'accélération/s ²)
498	Facultatif – C	Set		Compensation de frottement	REAL	Valeur ajoutée à la consigne de courant/couple afin de décaler les effets du frottement coulomb.	% de la valeur assignée du moteur
502	Facultatif – C	Set		Bande passante de filtre passe-bas de couple	REAL	Fréquence de rupture du deuxième filtre passe-bas appliqué au signal de référence de couple.	rad/s
503	Facultatif – C	Set		Fréquence de rejet de consigne de couple	REAL	Fréquence centrale du filtre coupe-bande appliqué au signal de référence de couple. Une valeur nulle désactive cette fonction.	rad/s
504	Obliga- toire – C	Set		Limite de couple - Positive	REAL	Détermine le couple positif maximal qui peut être appliqué au moteur. Si le dispositif tente de dépasser cette valeur, la consigne de couple est calée sur cette valeur.	% de la valeur assignée du moteur
505	Obliga- toire – C	Set		Limite de couple - Négative	REAL	Détermine le couple négatif maximal qui peut être appliqué au moteur. Si le dispositif tente de dépasser cette valeur, la consigne de couple est calée sur cette valeur.	% de la valeur assignée du moteur
506	Facultatif - C	Set		Limite de débit de couple	REAL	Limite le taux de variation du signal de référence de couple.	% de la valeur assignée du moteur/s
507	Facultatif – C	Set		Seuil de couple	REAL	Définit un niveau de couple absolu maximal qui, s'il est dépassé, définit le bit de statut Seuil de couple.	% de la valeur assignée du moteur
508	Facultatif – C	Set		Limite de sur-couple	REAL	Limite maximale du signal Référence de couple qui, si elle	% de la valeur

ID attr	Besoin dans la mise en œuvre	Règle d'accès	N V	Nom de l'attribut	Type de données	Description d'attribut	Sémantique des valeurs
						est dépassée pendant la durée spécifiée par l'attribut Temps de limite de sur-couple, donne lieu à une exception Limite de sur-couple. Cette fonction permet au dispositif de générer une exception en cas d'augmentation soudaine du couple de charge pendant le fonctionnement. Cette condition peut se produire en cas de défaillance du défilement ou d'une autre défaillance mécanique.	assignée du moteur
509	Facultatif – C	Set		Temps de limite de sur-couple	REAL	Spécifie la durée admise d'une condition Limite de sur-couple avant de générer une exception Limite de sur-couple. Une valeur nulle désactive la fonction Sous-couple.	Secondes
510	Facultatif – C	Set		Limite de sous-couple	REAL	Limite minimale du signal Référence de couple qui, si elle est dépassée pendant la durée spécifiée par l'attribut Temps de limite de sous-couple, donne lieu à une exception Limite de sous-couple. Cette fonction permet au dispositif de générer une exception en cas de diminution soudaine du couple de charge pendant le fonctionnement. Cette condition se produit en cas de rupture de couple de charge, de rupture de matériaux Web sous tension, etc.	% de la valeur assignée du moteur
511	Facultatif – C	Set		Temps de limite de sous-couple	REAL	Spécifie la durée admise d'une condition Limite de sous-couple avant de générer une exception Limite de sous-couple. Une valeur nulle désactive la fonction Sous-couple.	secondes

7.3.7.5 Attributs Boucle de courant

Le Tableau 52 et le Tableau 53 contiennent tous les attributs de boucle de courant associés à une instance Objet Axe de mouvement.

Tableau 52 – Attributs Signal de boucle de courant

ID attr	Besoin dans la mise en œuvre	Règle d'accès	N V	Nom de l'attribut	Type de données	Description d'attribut	Sémantique des valeurs
520	Obligatoire – C	Get		Consigne de courant Iq	REAL	Représente la valeur instantanée du couple commandé qui génère le signal de courant, Iq, avant de traverser le limiteur de courant.	% de la valeur assignée du moteur
521	Facultatif – C	Get		Limite de courant opératoire	REAL	Représente la limite de courant opératoire en fonction de plusieurs sources de limite.	% de la valeur assignée du moteur

ID attr	Besoin dans la mise en œuvre	Règle d'accès	N V	Nom de l'attribut	Type de données	Description d'attribut	Sémantique des valeurs
522	Facultatif – C	Get		Source de limite de courant	USINT	Représente la source opératoire d'une limite de couple lorsqu'une condition de limite de courant se produit.	Énumération: 0 = Non limité 1 = Limite de courant de crête de l'onduleur 2 = Limite de courant de crête du moteur 3 = Limite de courant thermique de l'onduleur 4 = Limite de courant thermique du moteur 5 = Limite de régulateur de shunt 6 à 127 = réservé 128 à 255 = spécifique au fournisseur
523	Obligatoire – C	Get		Angle électrique du moteur	REAL	Angle électrique calculé du moteur en fonction du nombre de pôles du moteur, du décalage de commutation et du dispositif d'entraînement de réaction sélectionné.	Degrés
524	Facultatif – C	Get		Référence de courant Iq	REAL	Signal de référence du courant, Iq, dans la jonction de somme de boucle de courant de couple.	% de la valeur assignée du moteur
525	Facultatif – C	Get		Référence de courant Id	REAL	Référence de courant de consigne, Id, dans la jonction de somme de courant de flux.	% de la valeur assignée du moteur
527	Facultatif – C	Get		Erreur de courant Iq	REAL	Erreur entre le courant commandé et le courant réel qui est la sortie de la jonction de somme de boucle de courant, q-axis, générant le couple.	% de la valeur assignée du moteur
528	Facultatif – C	Get		Erreur de courant Id	REAL	Erreur entre le courant commandé et le courant réel qui est la sortie de la jonction de somme de boucle de courant, d-axis, générant le flux.	% de la valeur assignée du moteur
529	Facultatif – C	Get		Retour de courant Iq	REAL	Courant de couple réel appliqué à l'axe en fonction du retour du capteur de courant.	% de la valeur assignée du moteur
530	Facultatif – C	Get		Retour de courant Id	REAL	Courant de flux réel appliqué à l'axe en fonction du retour du capteur de courant.	% de la valeur assignée du moteur
531	Facultatif – C	Get		Découplage Iq	REAL	Signal ajouté à la sortie de la boucle de régulation Iq pour compenser les effets de Id.	% de la valeur assignée du moteur
532	Facultatif – C	Get		Découplage Id	REAL	Signal ajouté à la sortie de la boucle de régulation Id pour compenser les effets de Iq.	% de la valeur assignée du moteur
533	Facultatif –	Get		Sortie de	REAL	Couple/force instantanés	% de la valeur

ID attr	Besoin dans la mise en œuvre	Règle d'accès	N V	Nom de l'attribut	Type de données	Description d'attribut	Sémantique des valeurs
	C			tension Vq		général la tension de sortie.	assignée du moteur
534	Facultatif – C	Get		Sortie de tension Vd	REAL	Flux instantané général la tension de sortie.	% de la valeur assignée du moteur
535	Facultatif – C	Get		Sortie de tension U	REAL	Tension instantanée appliquée à la phase U.	Volts
536	Facultatif – C	Get		Sortie de tension V	REAL	Tension instantanée appliquée à la phase V.	Volts
537	Facultatif – C	Get		Sortie de tension W	REAL	Tension instantanée appliquée à la phase W.	Volts
538	Facultatif – C	Get		Retour de courant U	REAL	Courant instantané mesuré sur la phase U.	Ampères
539	Facultatif – C	Get		Retour de courant V	REAL	Courant instantané mesuré sur la phase V.	Ampères
540	Facultatif – C	Get		Retour de courant W	REAL	Courant instantané mesuré sur la phase W.	Ampères
541	Facultatif – C	Get		Décalage de courant U	REAL	Décalage du transducteur de courant de phase U.	Ampères
542	Facultatif – C	Get		Décalage de courant V	REAL	Décalage du transducteur de courant de phase V.	Ampères
543	Facultatif – C	Get		Décalage de courant W	REAL	Décalage du transducteur de courant de phase W.	Ampères

Tableau 53 – Attributs Configuration de boucle de courant

ID attr	Besoin dans la mise en œuvre	Règle d'accès	N V	Nom de l'attribut	Type de données	Description d'attribut	Sémantique des valeurs
554	Facultatif – C	Get		Kqp	REAL	Valeur Gain proportionnel Iq qui multiplie le signal Erreur de courant Iq avant de l'appliquer à la jonction de somme de découplage Iq.	rad/s
555	Facultatif – C	Get		Kqi	REAL	Valeur Gain intégral Iq qui multiplie le signal Erreur de courant Iq avant de l'appliquer à l'accumulateur Erreur d'intégrateur Iq.	rad/s ²
556	Facultatif – C	Get		Kdp	REAL	Valeur Gain proportionnel Id qui multiplie le signal Erreur de courant Id avant de l'appliquer à la jonction de somme de découplage Iq.	rad/s
557	Facultatif – C	Get		Kdi	REAL	Valeur Gain intégral Id qui multiplie le signal Erreur de courant Id avant de l'appliquer à l'accumulateur Erreur d'intégrateur Id.	rad/s ²
558	Facultatif – IM	Set		Commande de flux ascendant	USINT	Si l'axe de mouvement est activé, le courant c.c. est appliqué à un moteur à induction afin de générer un flux de stator avant de passer à l'état Exécution. Cet attribut permet de contrôler l'application d'un flux à un moteur à induction à l'état Démarrage avant de passer à	Énumération: 0 = Pas de délai 1 = Délai manuel 2 = Délai automatique

ID attr	Besoin dans la mise en œuvre	Règle d'accès	N V	Nom de l'attribut	Type de données	Description d'attribut	Sémantique des valeurs
						l'état Exécution. Si Pas de délai est sélectionné, l'axe passe immédiatement à l'état Exécution lors de la génération du flux du moteur. Avec Délai manuel, l'axe reste à l'état Démarrage pendant le Temps de flux ascendant afin de laisser le temps au moteur de recevoir la totalité du flux. Avec Délai automatique, le dispositif d'entraînement détermine le délai de génération du flux du moteur en fonction des données d'attribut de configuration du moteur ou en fonction des mesures.	
559	Facultatif – IM	Set		Temps de flux ascendant	REAL	Définit le délai de génération d'un flux de moteur complet admis par le dispositif d'entraînement avant de passer à l'état Exécution.	Secondes

7.3.7.6 Attributs Contrôle de fréquence

Le Tableau 54 et le Tableau 55 contiennent tous les attributs associés à la méthode de fonctionnement Contrôle de fréquence d'une instance Objet Axe de mouvement.

Tableau 54 – Attributs Signal de contrôle de fréquence

ID attr	Besoin dans la mise en œuvre	Règle d'accès	N V	Nom de l'attribut	Type de données	Description d'attribut	Sémantique des valeurs
570	Obligatoire - F	Get		Compensation de glissement	REAL	Indique la valeur réelle de compensation de glissement appliquée.	r/min

Tableau 55 – Attributs Configuration de contrôle de fréquence

ID attr	Besoin dans la mise en œuvre	Règle d'accès	N V	Nom de l'attribut	Type de données	Description d'attribut	Sémantique des valeurs
575	Obligatoire - F	Set		Méthode de commande de fréquence	OCTET	La méthode de commande Volts/Hertz de base applique la tension au moteur, en général en relation directe avec la fréquence ou la vitesse commandée.	Énumération 0 = volts/hertz de base (R/F) 1 à 127 = réservé 128 à 255 = spécifique au fournisseur
571	Obligatoire - F	Set		Vitesse de glissement assignée	REAL	Représente la valeur de glissement au courant assigné (pleine charge) et à la fréquence assignée du moteur.	r/min
572	Obligatoire - F	Set		Tension maximale	REAL	Définit la tension la plus élevée entre phases que le dispositif d'entraînement peut	Volts (eff)

ID attr	Besoin dans la mise en œuvre	Règle d'accès	N V	Nom de l'attribut	Type de données	Description d'attribut	Sémantique des valeurs
						sortir.	
573	Obligatoire - F	Set		Fréquence maximale	REAL	Définit la fréquence la plus élevée que le dispositif d'entraînement peut sortir.	Hertz
574	Obligatoire - F	Set		Limite de fréquence	REAL	Définit la limite de fréquence généralement définie au-dessus de la Limite de vitesse, traduite en fréquence, permettant la compensation de glissement.	Hertz
575	Obligatoire - F	Set		Tension de rupture	REAL	Définit la tension de sortie entre phases du dispositif d'entraînement à la Fréquence de rupture à laquelle se termine la suralimentation.	Volts (eff)
576	Obligatoire - F	Set		Fréquence de rupture	REAL	Définit la fréquence de sortie du dispositif d'entraînement à la Tension de rupture à laquelle la suralimentation se termine.	Hertz
577	Obligatoire - F	Set		Suralimentation au démarrage	REAL	Définit le niveau de suralimentation de tension entre phases pour le départ et l'accélération.	Volts (eff)
578	Obligatoire - F	Set		Suralimentation à l'exécution	REAL	Définit le niveau de suralimentation de tension entre phases à la vitesse ou décélération en régime permanent.	Volts (eff)

7.3.7.7 Attributs Sortie de dispositif d'entraînement

Le Tableau 56 contient tous les attributs de sortie de l'onduleur associés à une instance Objet Axe de mouvement.

Tableau 56 – Attributs Sortie de dispositif d'entraînement

ID attr	Besoin dans la mise en œuvre	Règle d'accès	N V	Nom de l'attribut	Type de données	Description d'attribut	Sémantique des valeurs
600	Obligatoire - PVT	Get		Fréquence de sortie	REAL	Fréquence de sortie moyennée dans le temps appliquée au moteur.	Hertz
601	Obligatoire - PVT	Get		Courant de sortie	REAL	Courant de sortie moyenné dans le temps total appliqué au moteur.	Ampères (eff)
602	Obligatoire - PVT	Get		Tension de sortie	REAL	Tension de sortie entre phases moyennée dans le temps totale appliquée au moteur.	Volts (eff)
603	Obligatoire - PVT	Get		Puissance de sortie	REAL	Puissance de sortie moyennée dans le temps totale du moteur. Cette valeur repose sur le produit du signal Référence de couple et Retour en vitesse.	kW

ID attr	Besoin dans la mise en œuvre	Règle d'accès	N V	Nom de l'attribut	Type de données	Description d'attribut	Sémantique des valeurs
604	Facultatif - PVT	Set		Fréquence MID	UINT	Définit la fréquence de porteuse de la sortie Modulation d'impulsions en durée appliquée au moteur. Le déclassement du dispositif d'entraînement est requis à des fréquences MID plus élevées en raison de pertes de commutation. Le temps d'actualisation de la boucle de courant est directement lié à la fréquence MID. Par conséquent, les performances de la boucle augmentent en général au fur et à mesure de l'augmentation du taux MID.	Hertz

7.3.8 Attributs Arrêt et freinage

7.3.8.1 Généralités

Le Tableau 57 contient tous les attributs d'arrêt et de freinage associés à une instance Objet Axe de mouvement.

Tableau 57 – Attributs Arrêt/Freinage

ID attr	Besoin dans la mise en œuvre	Règle d'accès	N V	Nom de l'attribut	Type de données	Description d'attribut	Sémantique des valeurs
610	Obligatoire - PVT	Set		Mode d'arrêt	USINT	Lors de la désactivation ou de l'abandon d'un axe, cette valeur détermine la méthode d'arrêt à appliquer au moteur. L'état final appliqué à l'issue de la méthode d'arrêt est Arrêté. A l'état Arrêté, il convient que la structure d'alimentation de l'onduleur du dispositif soit Désactivée (sélection Désactivé) et sans couple ou maintenue active (Sélection Maintien) dans une condition statique via une action asservie.	Voir la sémantique en 7.3.8.2.1
611	Obligatoire - PVT	Set		Courant d'arrêt	REAL	Lors de la désactivation ou de l'abandon d'un axe, cette valeur détermine le couple maximal générant le courant disponible pour arrêter le moteur lorsque le Mode d'arrêt est défini sur Décel. de courant. Si cet attribut n'est pas pris en charge, le dispositif d'entraînement doit utiliser les limites positive et négative de courant de crête.	% de la valeur assignée du moteur
612	Obligatoire - PVT	Set		Limite de temps d'arrêt	REAL	Lors de la désactivation ou de l'abandon d'un axe, ce paramètre détermine la valeur maximale du Courant d'arrêt qui peut être appliquée pour arrêter le moteur avant que l'axe ne soit désactivé et la totalité du courant retirée du moteur.	Secondes
613	Facultatif – PVT (moteurs à	Set		Délai de contact de frein	REAL	Lorsqu'un frein résistif externe est utilisé, le Délai de contact de frein résistif peut être défini pour	Secondes Voir la sémantique

ID attr	Besoin dans la mise en œuvre	Règle d'accès	N V	Nom de l'attribut	Type de données	Description d'attribut	Sémantique des valeurs
	aimant permanent unique-ment)			résistif		retarder l'activation de la structure d'alimentation du dispositif d'entraînement pour que le frein résistif ait le temps de connecter le moteur au dispositif d'entraînement.	en 7.3.8.2.2 pour plus de détails.
614	Facultatif - PVT	Set		Commande de frein mécanique	USINT	Cet attribut permet de commander directement la sortie du dispositif d'entraînement connectée au mécanisme de freinage mécanique. S'il est défini pour libérer ou engager le frein, ce dernier est libéré ou engagé quel que soit l'état en cours de l'axe. Une transition d'état de l'axe peut toutefois remplacer le paramètre Commande de frein mécanique. Par exemple, si cet attribut est nul, que le frein est libéré et que l'axe passe de l'état Arrêté à l'état Exécution, puis revient à l'état Arrêté, le frein s'engage automatiquement lors du passage à l'état Arrêté. Le dispositif indique cette condition en attribuant la valeur 1 au paramètre Commande de frein mécanique, indiquant que le frein est engagé.	Énumération: 0 = Libérer le frein 1 = Engager le frein
615	Obligatoire - PVT	Set		Délai de libération de frein mécanique	REAL	Lors de l'activation de l'axe, la valeur Délai de libération de frein mécanique détermine la durée à laquelle le dispositif d'entraînement doit retarder le passage de l'état Démarrage à l'état Exécution ou Essais. Ce délai empêche tout mouvement commandé de l'axe de mouvement pour laisser le temps au frein mécanique externe de se désengager.	Secondes Voir la sémantique en 7.3.8.2.3 pour plus de détails.
616	Obligatoire - PVT	Set		Délai d'engagement de frein mécanique	REAL	Lors de la désactivation de l'axe, la valeur Délai d'engagement de frein mécanique détermine pendant combien de temps la structure d'alimentation de l'onduleur doit rester activée après la décélération de l'axe. Cela laisse le temps à un frein mécanique externe de s'engager. Le Mode d'arrêt configuré détermine le type de séquence d'arrêt appliqué.	Secondes Voir la sémantique en 7.3.8.2.4 pour plus de détails.

7.3.8.2 Sémantique

7.3.8.2.1 Attribut N° 610 – Mode d'arrêt

Lors de la désactivation ou de l'abandon d'un axe, cet attribut détermine la méthode d'arrêt à appliquer au moteur (voir Tableau 58). L'état final appliqué à l'issue de la méthode d'arrêt est Arrêté. A l'état Arrêté, il convient que la structure d'alimentation de l'onduleur de l'axe soit Désactivée (sélection Désactivé) et sans couple ou maintenue active (Sélection Maintien) dans une condition statique via une action asservie. Cet attribut n'a aucun impact sur les profils d'accélération et de décélération générés par le planificateur ni ne leurs sont liés.

Tableau 58 – Définitions de l'énumération Mode d'arrêt

Énum.	Oblig. Facult.	Nom	Description
0	R/CF	Désactivé & arrêt progressif	Désactivé & arrêt progressif désactive immédiatement la structure d'alimentation de l'onduleur, provoquant la mise en arrêt progressif du moteur, sauf si une certaine forme de freinage externe est appliquée. Équivaut à l'arrêt de catégorie O de la CEI 60204-1.
1	O/C	Décél. de courant & Désactiver	Décél. de courant & Désactiver oblige le dispositif d'entraînement à s'arrêter sous le contrôle du courant d'arrêt configuré. Pour ce faire, il s'agit de forcer les boucles de régulation en amont pertinentes à saturer en attribuant des valeurs fixes aux références de consigne appropriées. Dans le mode de position, la consigne de position est réinitialisée à sa position réelle lorsque l'axe atteint une vitesse nulle. A l'arrêt ou à l'expiration de la Limite de temps d'arrêt configurée, le dispositif désactive la structure d'alimentation et les boucles de régulation. Ce mode d'arrêt est conforme à l'arrêt de catégorie 1 de la CEI 60204-1.
2	R/V	Décél. rampe & Désactiver	Décél. rampe & Désactiver s'applique uniquement au mode de commande de vitesse. Ce mode d'arrêt annule immédiatement la sortie du Sélecteur de consigne de vitesse incitant le limiteur d'accélération à réduire la vitesse du moteur à zéro. A l'arrêt ou à l'expiration de la Limite de temps d'arrêt configurée, le dispositif désactive la structure d'alimentation et les boucles de régulation. Ce mode d'arrêt est également conforme à l'arrêt de catégorie 1 de la CEI 60204-1.
3	F/C	Décél. de courant & Maintenir	Décél. de courant & Maintenir se comporte comme Décél. de courant & Désactiver, mais laisse la structure d'alimentation et les boucles de régulation actives sous le contrôle de référence de consigne locale avec couple statique. De cette manière, un couple statique actif permet de conserver la condition d'arrêt. Ce mode d'arrêt est conforme à l'arrêt de catégorie 2 de la CEI 60204-1.
4	O/F	Décél. rampe & Maintenir	Décél. rampe & Maintenir se comporte comme Décél. rampe & Désactiver, mais laisse la structure d'alimentation et les boucles de régulation actives sous le contrôle de référence de consigne locale. De cette manière, un couple statique actif permet de conserver la condition d'arrêt. Ce mode d'arrêt est également conforme à l'arrêt de catégorie 2 de la CEI 60204-1.
5 à 127		(Réserve)	-
128 à 255		(Spécifique au fournisseur)	-

7.3.8.2.2 Attribut N° 613 – Délai de contact de frein résistif

Lorsqu'un frein résistif externe est utilisé, un contacteur externe commute les cosses du moteur UVW de la structure d'alimentation de l'onduleur vers une résistance de dissipation d'énergie pour arrêter le moteur. Cette commutation ne se produit pas instantanément, l'activation prématurée de la structure d'alimentation pouvant donc provoquer un arc électrique dans le contacteur. Pour éviter cette condition, le Délai de contact de frein résistif peut être défini comme étant la durée maximale de fermeture totale du contacteur dans les lignes de moteur UVW de sorte que, lors de l'activation de l'axe, la structure d'alimentation de l'onduleur n'est pas activée tant que le Délai de contact de frein résistif n'est pas activé. L'opération de freinage résistif s'applique uniquement aux types de moteur à aimant permanent. La séquence ci-dessous illustre les facteurs Délai de contact de frein résistif dans la séquence d'activation globale qui peut également inclure le fonctionnement d'un frein mécanique.

Séquence d'activation:

- 1) Passer à l'état Démarrage.

- 2) Activer le contacteur de frein résistif pour connecter le moteur à la structure d'alimentation de l'onduleur.
- 3) Attendre le "Délai de contact de frein résistif" pendant la fermeture des contacts de frein résistif.
- 4) Activer la structure d'alimentation de l'onduleur.
- 5) Activer la sortie Frein mécanique pour libérer le frein.
- 6) Attendre le "Délai de libération de frein mécanique" lors de la détente du frein.
- 7) Passer à l'état Exécution.

7.3.8.2.3 Attribut N° 615 – Délai de libération de frein mécanique

Lors de l'activation de l'axe, la valeur Délai de libération de frein mécanique détermine la durée à laquelle l'axe doit retarder le passage de l'état Démarrage à l'état Exécution ou Essais. Ce délai empêche tout mouvement commandé de l'axe pour laisser le temps au frein mécanique externe de se désengager.

Séquence d'activation:

- 1) Passer à l'état Démarrage.
- 2) Activer le contacteur de frein résistif pour connecter le moteur à la structure d'alimentation de l'onduleur.
- 3) Attendre le "Délai de contact de frein résistif" pendant la fermeture des contacts de frein résistif.
- 4) Activer la structure d'alimentation de l'onduleur.
- 5) Activer la sortie Frein mécanique pour libérer le frein.
- 6) Attendre le "Délai de libération de frein mécanique" lors de la détente du frein.
- 7) Passer à l'état Exécution (ou Essais).

7.3.8.2.4 Attribut N° 616 – Délai d'engagement de frein mécanique

Lors de la désactivation de l'axe, la valeur Délai d'engagement de frein mécanique détermine pendant combien de temps la structure d'alimentation de l'onduleur doit rester activée après la décélération de l'axe. Cela laisse le temps à un frein mécanique externe de s'engager. Le Mode d'arrêt configuré détermine le type de séquence d'arrêt appliqué.

Séquence de désactivation (Arrêt de catégorie 0):

- 1) Désactiver la structure d'alimentation de l'onduleur.
- 2) Passer à l'état Arrêté.
- 3) Désactiver la sortie Frein mécanique pour engager le frein.
- 4) Désactiver le contacteur de frein résistif pour déconnecter le moteur de la structure d'alimentation de l'onduleur.

Séquence de désactivation (Arrêt de catégorie 1):

- 1) Passer à l'état Arrêt.
- 2) Appliquer la méthode "Décél. de courant" ou "Décél ramp" pour arrêter le moteur.
- 3) Attendre la vitesse nulle ou la "Limite de temps d'arrêt", selon ce qui se produit en premier.
- 4) Désactiver la sortie Frein mécanique pour engager le frein.
- 5) Attendre le "Délai d'engagement de frein mécanique" lors de l'engagement du frein.
- 6) Désactiver la structure d'alimentation de l'onduleur.
- 7) Passer à l'état Arrêté.