

HORIDEL®

Advanced Servo Drive

User Manual

PRELIMINARY VERSION

18th February 2022

Not for general distribution

General.....	1
Inputs and Outputs	1
Hardware	1
Human-Machine Interface (HMI).....	2
Connector and Terminal Locations	3
Control terminals	4
24V Power Supply Output.....	4
Digital Inputs	4
Digital Outputs	5
PE	5
Relay Outputs.....	5
10V Analogue Power Supply Outputs.....	5
Analogue Inputs	6
Analogue Outputs	6
EIA/RS-485 Communications	7
PE	7
Thermal Protection	7
JP Link.....	8
Default analogue input setup	8
Default analogue output setup.....	8
Operation	10
Basic HMI operation.....	10
HMI Status indicator LEDs.....	11
Configuration and Control	12
Parameters.....	12
Actual Values.....	12
Terminal Strip.....	12
Set Points	13
Executing motor parameter identification tasks	13
Setup for initial testing.....	13
Initial testing steps	13
Configure parameters	15
Power stack parameters	15
Motor parameters.....	15

Induction Motor Parameters	15
Synchronous Motor (PMSM) Parameters	17
Shaft sensor parameters	18
Trial Operation	19
Test run	22
Motor parameter identification.....	25
Test operation in closed loop speed control	25
Position measurement.....	27
Configuring and testing position control modes	29
Position control modes	29
Position references	29
Position feedback.....	29
Selection of motor shaft sensors	29
Absolute angle sensors	29
Incremental angle sensors	29
Resolver selection for PMSM applications	31
Motor Parameter Identification (Auto-tuning).....	31
Induction motor systems	31
Synchronous motor (PMSM) systems.....	32
Activating the identification processes.....	32
MSC-3 Power stacks.....	36
Limit switches.....	38
CANBUS.....	38
Controller parameters	39
HM03 Measured Value Acquisition	41
Expansion sockets	41
Drive control	43
References	44
I2C I/O expansion.....	44
Additional topics	44
Once/twice control per switching interval	44
Modifications to MSC-3 circuits for use with ASD	46
ASD User Levels.....	48
Parameter setup for torque limitation	50

Software update using the ZAP updater.....	52
ASD - How to use the simulator feature v0.002	53
Various things to know	53
Setup	53
What else to consider... ..	55
Using simulation mode to verify the PWM outputs of ASD with software version r020	55
Restrictions in Software version r020	56
Verify the function of the PWM outputs on the control board.....	56
ASD - setpoint selection	58
Setpoint selection and processing	58
Configuration	58
Setpoint sources	59
Setpoint source <i>disabled</i>	59
Setpoint source terminals or terminals binary	59
Setpoint source <i>bus</i> or <i>bus var</i>	60
Setpoint source fix value or fix value varDir	60
Setpoint source <i>PID controller</i>	60
Setpoint source analog in1 to analog in3	60
Setpoint type, scaling and filtering	60
Setpoint Limitation	61
Fix setpoint limitation.....	61
Dynamic setpoint limitation	62
Direction reversal.....	62
Setpoint selection and processing parameter overview	63
Conditions for enable, jog mode and latched FWD/REV.....	64
Power stack temperature evaluation, monitoring and fan control.....	65
Differences between chassis A/B/C and chassis D	65
Signal processing.....	65
Different Type of temperature sensors:	65
SKiiP3 temperature sensors.....	66
LM335 temperature sensors.....	68
Custom linear Type temperature sensors	68
Temperature sensors with series resistance	69
Configuration example for a random temperature sensor	69

Over-temperature detection	69
Over-temperature switch off	69
Over-temperature warning.....	70
Fan control	70
Motor Lsq / Lsd measurement.....	71
Procedure.....	71
Alternative measurement	71
ASD - Modbus documentation for software version v10r020b	72
General purpose PID controller	79
V/Hz operation mode	91
Standard V/Hz parameters	91
V/Hz parameters for voltage and current limitation	91
ASD – IM and PMSM field weakening functions.....	93
Field weakening operation for induction motors	93
Field weakening operation for permanent magnet synchronous motors.....	94
Pulse train inputs	96
Actual values update rates and filter time constants	97
Update time intervals	97
Display filtering	97
ASD - Parameters for torque control with external torque transducer	100
Encoder less operation with ASD and PMSM. Revision 0.003	102
Parameter Identification (Auto-tuning)	112
Induction motor systems	112
Synchronous motor (PMSM) systems.....	113
Activating the identification processes.....	113
Display Messages	115
Error Messages	115
Warning Messages	117
Frequently Asked Questions - FAQ	118
How do you change the user level?	118
How do you set the ASD to operate in the simulator mode?	118
Reference diagrams	119
Speed control loop refM	119
Position control loop.....	119

Position control loop nFF, TFF.....	119
Position control loop nFF, TFF , n from ramp	119
Parameter list.....	120
Document change history	120

General

The ASD provides for control of induction (IM) and synchronous (PMSM) motors with the following operating modes:

- Torque control using internal torque evaluation
- Torque control with external torque transducer
- Closed loop speed control based on motor shaft sensor
- Closed loop position control based on motor shaft sensor
- Position difference control (position follower)

In addition V/Hz control and constant current modes are provided for testing purposes

Inputs and Outputs

Hardware

The ASD controller provides the following inputs and outputs to connect to other parts of a system:

Type	Quantity
Digital inputs – general purpose (individual functions configurable)	10
Digital input - Enable	1
Digital outputs - general purpose (individual functions configurable)	2
Relay outputs - general purpose (individual functions configurable)	2
Analogue inputs - general purpose (individual functions configurable)	3
Analogue outputs - general purpose (individual functions configurable)	2
PTC thermistor input for motor thermistors	1
NTC thermistor input to monitor motor or other temperature	1
Incremental encoder interface	1
Expansion connectors for plug-in interfaces to suit various sensor types including resolvers, incremental encoders, Hiperface, Heidenhain ERN1387 and Sin/Cos shaft sensors	2
Modbus RTU port for control from a computer or PLC	1
CANbus port for configuration upload/download	1
HMI interface (see below for HMI facilities)	1
Power stack interface	1

Human-Machine Interface (HMI)

The major features of the HMI are:

Alpha-Numeric LCD display for parameter editing and general display of operation with facility for multiple languages

Dedicated buttons for FWD, REV, STOP and RESET functions

Dedicated buttons for parameter editing

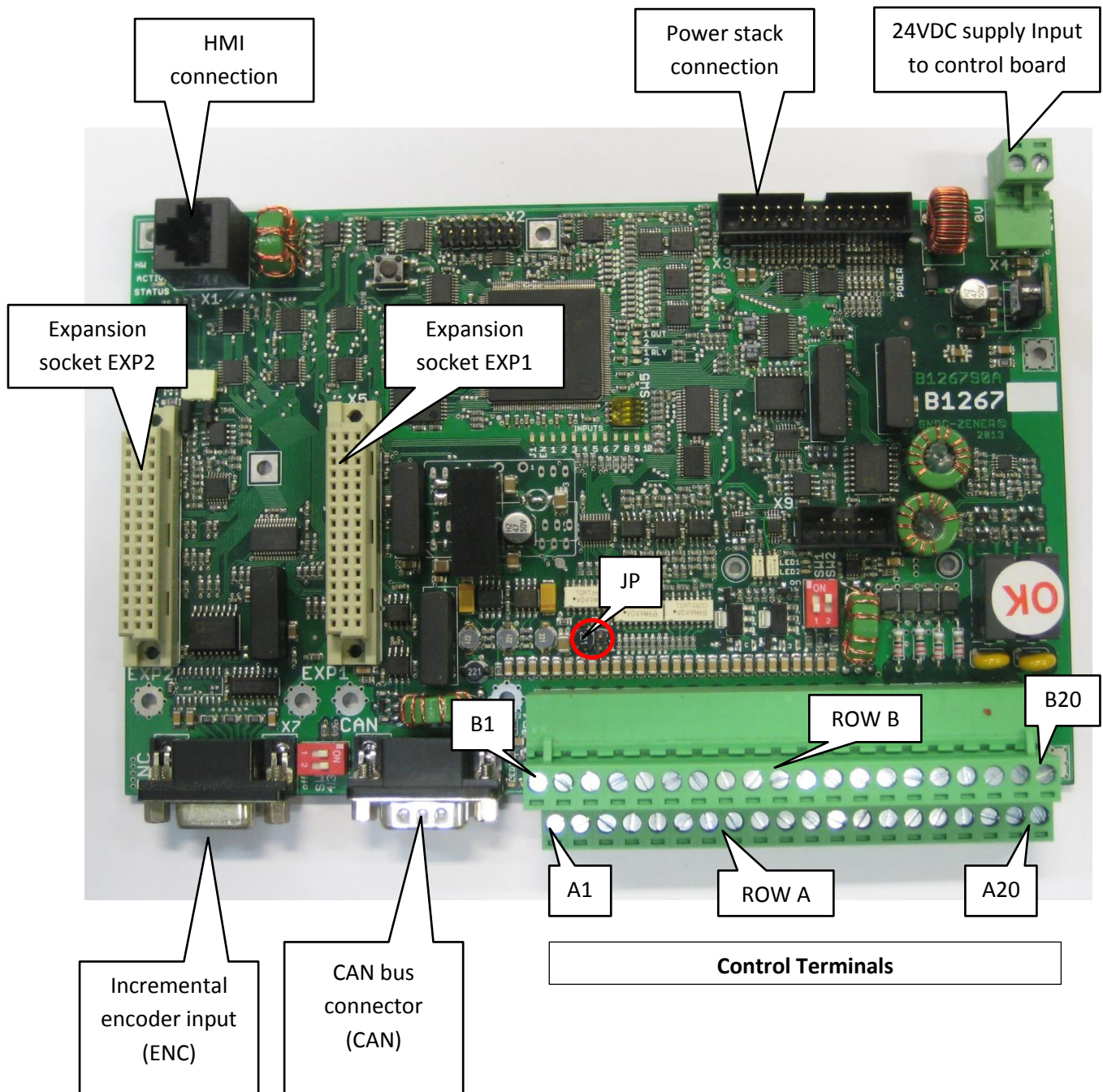
User configurable function buttons (2)

User configurable “soft” function buttons (8)

Dedicated LEDs for Active, Limit and Error status indication

Connection to the ASD controller circuit board by standard RJ45 UTP patch cable

Connector and Terminal Locations



Control terminals

The various inputs and outputs can be configured for a wide variety of functions with few exceptions. The functions shown in the table below are the default functions assigned to the various terminals. The default configuration should be considered just as a starting point to configure the system to your specific requirements.

24V Power Supply Output

Terminal	Name	Function	Note
A1	D_24V	+24V Power supply output	For customer use with digital inputs
A2	D_GND	Common connection for 24V power supply output and digital inputs	

Digital Inputs

Terminal	Name	Parameter to configure the function of this terminal	Default settings		
			Function	Interpretation	Source
		TInn = TI01 etc	TInn	TInn + 1	TInn +2
A3	Enable	TI01 fnc.Enable	inverter enable	active high	terminal
A4	DIN1	TI05 fnc.DigIn1	run	active high	terminal
A5	DIN2	TI09 fnc.DigIn2	emergency stop	active high	terminal
A6	DIN3	TI13 fnc.DigIn3	inactive	active high	terminal
A7	DIN4	TI17 fnc.DigIn4	Speed control	active high	terminal
A8	DIN5	TI21 fnc.DigIn5	Torque control	active high	terminal
A9	DIN6	TI25 fnc.DigIn6	inactive	active high	terminal
A10	DIN7	TI29 fnc.DigIn7	pre.pos.lim.rght	active high	terminal
A11	DIN8	TI33 fnc.DigIn8	pre.pos.lim.left	active high	terminal
A12	DIN9	TI37 fnc.DigIn9	pos.lim. right	active high	terminal
A13	DIN10	TI41 fnc.DigIn10	pos.lim. left	active high	terminal

Digital Outputs

Terminal	Name	Parameter to configure the function of this terminal	Default settings		
			Function	Interpretation	Source
		TOnn = TO01 etc	TOnn	TOnn + 1	TOnn +2
A14	DOUT1	TO01 fnc.DigOut1	inactive	active high	terminal
A15	DOUT2	TO05 fnc.DigOut2	inactive	active high	terminal

PE

Terminal	Name	Function
A16	PE	Protective earth terminal (use for cable screens etc)

Relay Outputs

Terminal	Name	Parameters to configure the function of this terminal	Default settings		
			Function	Interpretation	Source
		TOnn = TO01 etc	TOnn	TOnn + 1	TOnn +2
A17/18	RLY1	TO09 fnc.RlyOut1	inactive	active high	terminal
A19/20	RLY2	TO13 fnc.RlyOut2	inactive	active high	terminal

10V Analogue Power Supply Outputs

Terminal	Name	Function
B1	AIN_+10V	+10V output for use with analogue inputs
B2	AIN_GND	Common connection for 10V power supplies and also all analogue inputs
B3	AIN_-10V	-10V output for use with analogue inputs

Analogue Inputs

Terminal	Name	Parameters to configure the function of this terminal	Default Configuration
B4	AIN1+	AI10 4-20mA ana1	Voltage input
B5	AIN1-	AI11 en.neg.ana1 AI12 scale ana1 AI13 offset ana1 AI16 timec.ana1	disabled Scale 10V Offset 0V Filter time 2ms
B6	AIN2+	AI21 en.neg.ana2	disabled
B7	AIN2-	AI22 scale ana2 AI25 offst.cmp.2 AI26 timec.ana2	Scale 10V Offset 0V Filter time 2ms
B8	AIN3+	AI31 en.neg.ana3	disabled
B9	AIN3-	AI32 scale ana3 AI33 offset ana3 AI36 timec.ana3	Scale 10V Offset 0V Filter time 2ms

Analogue Outputs

Terminal	Name	Parameter to configure the function of this terminal	Default Configuration		
			Function	Offset	Offset value
B10	AOUT1	AA01 sign.chan.1	0V	AA03 offset ch.1	0mV
B11	AOUT_GND	Analogue output common connection			
B12	AOUT2	AA02 sign.chan.2	0V	AA04 offset ch.2	0mV

EIA/RS-485 Communications

Terminal	Name	Function	Note
B13	BUS_A	EIA/RS-485 data wire A	A used for Modbus RTU communications
B14	BUS_B	EIA/RS-485 data wire B	
B15	BUS_C	EIA/RS-485 common	

PE

Terminal	Name	Function
B16	PE	Protective earth terminal (use for cable screens etc)

Thermal Protection

Terminal	Name	Parameter to configure the function of this terminal	Default Configuration
B17	NTC+	TM01 NTC R25	1000 Ohms at 25°C
B18	NTC-	TM02 NTC B	B = 3600
		TM03 NTC tmp.off	Switch off temperature = 0°C
B19	PTC+	UE20 PTC-monitor	Off
B20	PTC-		

JP Link

JP is two pins labelled “J1” located on the ASD controller circuit board just behind terminal A3. The JP input is true if the shorting jumper is fitted and false if it is not. Do not make other connections to these pins.

Default settings		
Function	Interpretation	Source
TI45 fnc.DigInJP	TI46 lvl.DigInJP	TI47 src.DigInJP
inactive	active high	terminal

Default analogue input setup

The default setup for all of the analogue inputs is to accept a +/- 10V with no offset. By default, 2ms filtering is applied by the ASD control before using the analogue input signal. See parameters AI10...37

The following parameters are assigned to the analogue inputs by default:

Parameter	Setting	Notes
FR01 spd SP src	Analog in1	Speed control reference
DP01 trq.SP src	Analog in2	Torque control reference
AA01 sign.chan 1	Analog in3	This just connects the analogue input to the analogue output for testing and demonstration

Default analogue output setup

The following parameters are assigned to the analogue outputs by default:

[Need to fix parameter numbering and add span function]

Analogue output	Terminal	Parameter	Function	Offset	Span
		AAnn = AA01 etc	AAnn	AAnn + 1	
AOUT1	B10, B11(com)	AA01 sign.chan.1	Analog in3		

AOUT2	B12, B11 (com)	AA02 sign.chan.2	iS.q		
-------	-------------------	------------------	------	--	--

Operation

Basic HMI operation

Press the ESC button to switch between the display of the drive status and the parameter menu.

The parameter menu follows a tree structure with branches for various groups of parameters as below:

- >Start menu ASD
 - >Parameters
 - >parameter main menu
 - >parameter sub menu
 - >edit parameter
 - >Actual values
 - >list of actual values
 - >Setpoints
 - >place holder (nothing implemented here)
 - >Terminals
 - >place holder (nothing implemented here)
 - >ASD control
 - >execute user defined functions with the buttons F1/F2. Navigate with Left/Right
 - >Sys.information
 - >Soft and hardware versions are displayed
- >Drive status page

Navigate in the menu with the arrow keys up/down/left/right. Press OK to open a sub menu or to edit a parameter.

Parameters with a little 'key' symbol cannot be edited.

Press ESC to jump to the parent menu.

From the start menu press ESC to go to the drive status page and any navigation key (including OK and ESC) to go back to the start menu.

During parameter edit:

If the parameter value is a number: use left/right to select the digit and up/down to change its value.

If the parameter value is an item of a selection list: use left/right to select the choice

Press OK twice to use and save the new value or press ESC.

The menu structure is dynamic and can change while changing some of the parameters. If you change a parameter that changes the structure the HMI jumps back to the start menu.

Hold a key for automatic keyboard repeat.

Special keys:

RESET is to reset an error

F1 and F2 can be configured to have many different functions

FWD, REV and STOP are initially configured for those functions, but may also be re-configured.

Soft buttons are accessed by selecting the ASD Control page in the parameter menu.

HMI Status indicator LEDs

LED green (Ready)	LED yellow (Limit)	LED red (Error)	Description
			Connected + NOT ready for operation
			
			Connected + Ready for operation
			
			Inverter is active
			Inverter is active + warning
			Inverter is active + limitation
			Inverter is active + warning + limitation
			ERROR in ASD
			configuration error
			Not connected/ communication error or start up
			Restart required

By default the HMI buttons are mapped to the following functions....

Button	Parameter	Function	Comment
FWD	HB01 fct.FWD	Forward & latch	
REV	HB03 fct.REV	Reverse & latch	
STOP	HB05 fct.STOP	Stop Reset F/R Latch	
F1	HB07 fct.F1		
F2	HB09 fct.F2		
Soft button FN1	HB11 fct.Fn1	Jog forward	
Soft button FN2	HB13 fct.Fn2	Jog reverse	
Soft button FN3	HB15 fct.Fn3		
Soft button FN4	HB17 fct.Fn4		
Soft button FN5	HB19 fct.Fn5	Ls-ident.start	...start function, this means you can release

			the button after the procedure has started
Soft button FN6	HB21 fct.Fn6		
Soft button FN7	HB23 fct.Fn7	Sat+i0. Id.start	
Soft button FN8	HB25 fct.Fn8	RP-ident.start	

Note the arrow keys, OK button and ESC are permanently assigned to editing functions. The reset button is permanently assigned to the error reset function.

Configuration and Control

ASD configuration is determined by an extensive menu of parameters. See the parameter listing for full details. The parameters are arranged in groups as follows:

Parameters

HM01 Motor Parameters

HM02 Controller Parameters

HM03 Measured Value Acquisition

HM04 Characteristic curves

HM05 Drive Control

HM06

HM07 Monitoring

HM08 Power Stack

HM09 Application Parameters

HM10 CAN Parameters

HM11 Modbus Parameters

HM12 Analogue Output

HM13

HM14

HM15 User Control

Actual Values

Terminal Strip

Set Points

Executing motor parameter identification tasks

The ASD has several inbuilt task sequences used to identify important motor parameters. The execution of each identification task is controlled from an input terminal with the particular identification task assigned to it.

Motor identification tasks may be assigned to any input terminal (DIN1....DIN10) and certain buttons on the HMI. There is no need to assign terminals for the motor identification tasks that you will not use (eg tasks only required for a PMSM if your system uses an induction motor).

See section *Motor Parameter Identification (Auto-tuning)* later in this manual for additional details

Setup for initial testing

Initial testing steps

The main steps in setting up the ASD controller for basic initial testing are

- Configure parameters
- Trial operation to verify:
 - Measurements
 - Motor rotation direction
 - Motor shaft sensor operation
- Motor parameter identification
- Test operation in closed loop speed control

Setup for more complex configurations will require additional steps. The notes below highlight the important issues for each of these steps and detail where other necessary information can be found in this manual.

CAUTION



Possibility of damage to permanent magnet synchronous motors (PMSM)

Excessive or uncontrolled current in a PMSM may cause permanent damage to the motor by demagnetising the magnets. To reduce the risk of this type of damage, we strongly recommend that the initial setup of any new design of power stack be done with an induction motor of similar electrical rating. The PMSM should not be connected until you have verified that the current control and current limit features of the ASD have been configured and are working properly.

Configure parameters

Power stack parameters

See section of this manual **MSC-3 Power Stack** for parameter settings.



IMPORTANT

The power stack parameters should be set according to the power stack design / ratings at the beginning of the setup procedure and not changed later. If it is necessary to change the power stack parameters (especially the current normalisation LT08 curr.normal) due to discovery of some setting error, it is important to **repeat all of the motor parameter identification steps**. The reason is that the motor parameter identification steps depend on accurate voltage and current information. LT08 and other power stack parameters are a critical to ASD correctly interpreting the measured values.

Motor parameters

For an induction motor (IM) applications, set the motor related parameters according to the table below:

Induction Motor Parameters

Parameter	Setting	Notes
MP01 motor type	IM	
MP02 rated freq.	According to motor rated frequency in Hz	
MP03 rat. speed	According to motor rated speed in RPM	
MP04 rat.voltage	According to motor rated line voltage (voltage between phases)	
MP05 rat.current	According to motor rated current in Amps	
MP06 max.current	Current limit. Maximum current under any condition	Note that there is also a separate current limit parameter for the power stack (LT10). The maximum current is determined by the lower of MP06 and LT10.
MP07 no load cur	According to motor or 25% of MP05	Saturation/No-Load-current identification ('satur. Identific') process may fail if the initial value of MP07 is too

		low. See Motor Parameter Identification (Auto-tuning) Induction motor systems section of this manual.
MP08 rat. power	According to motor rated power in kW	
MP09 rat. torque	Display value – nothing to enter	Automatically calculated from other parameters
MP10 rat. cosphi	According to motor data	If unknown: set to 0 and set up MP22 (resistance between two motor terminals)
MP11 Lsd / Lsq	Not relevant for IM	
MP13 Lsq UV (op)	Not relevant for IM	
MP15 kE UV (op)	Not relevant for IM	
MP17 Rs UV (op)	According to motor winding resistance measured between two phases (3 rd phase open) in Ohms	Set to 0 if the value is not available
MP18 J (op)	According to motor moment of inertia in kgm ²	Set to 0 if the value is not available

For synchronous motor (PMSM) applications, set the motor related parameters according to the table below:

Synchronous Motor (PMSM) Parameters

Parameter	Setting	Notes
MP01 motor type	PMSM	
MP02 rated freq.	According to motor rated frequency in Hz	
MP03 rat. speed	According to motor rated speed in RPM	
MP04 rat.voltage	According to motor rated line voltage (voltage between phases)	
MP05 rat.current	According to motor rated current in Amps	
MP06 max.current	Current limit. Maximum current under any condition	Note that there is also a separate current limit parameter for the power stack (LT10). The maximum current is determined by the lower of MP06 and LT10.
MP07 no load cur	Not relevant for synchronous motors	
MP08 rat. power	Display value – nothing to enter	Automatically calculated from other parameters
MP09 rat. torque	According to motor data in Nm	Alternatively, calculate as $(kW * 9550)/RPM$
MP10 rat. cosphi	Not relevant for PMSM	
MP11 Lsd / Lsq	According to motor or 1.0 if the value is not available	
MP13 Lsq UV (op)	According to motor leakage inductance Lsq phase-to-phase in mH	Set to 0 if the value is not available
MP15 kE UV (op	According to motor voltage constant in V/1000rpm	Set to 0 if the value is not available

MP17 Rs UV (op)	According to motor winding resistance measured between two phases (3rd phase open) in Ohms	Set to 0 if the value is not available
MP18 J (op)	According to motor moment of inertia in kgm ²	Set to 0 if the value is not available

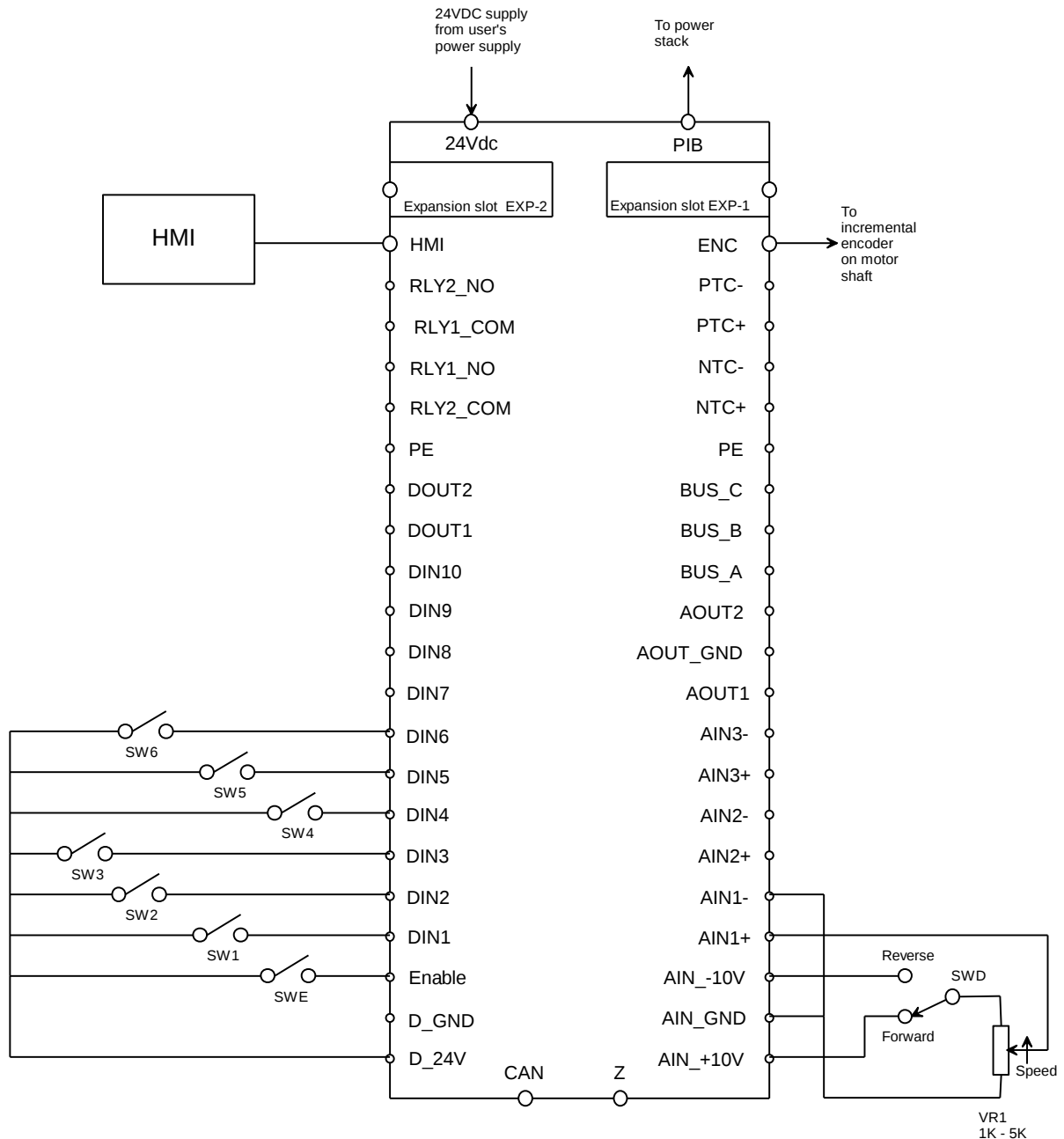
Shaft sensor parameters

These parameters are for an incremental shaft encoder connected to the ASD controller at connector X7. Other types of shaft sensor and those using one of the plug in expansion boards will require different settings.

Parameter	Setting	Notes
ME01 mot enc src	Onboard (ENC)	
EC01 IE sig.typ	quadrature signl	
EC02 IE line cnt	Number of lines per revolution for the shaft encoder	
EC03 IE timeMeas		
EC04 IE TtimMeas	5ms	
EC05 IE alarm	on	Some versions of the control board do not have the necessary hardware for alarm function on the ENC port fitted.
EC06 index eval.	on	Set to “on” if the encoder has an index pulse (Z channel), “off” if it does not. An index pulse is mandatory for incremental encoders used with synchronous motors.
EC07 cnt.reversl	off	

Trial Operation

Make the following temporary connections for trial operation:



CAUTION At analogue input values very close to zero, small errors may make the control of direction by switching between +10V and -10V problematic. In this case, a better solution is disable negative analogue signals and use forward and reverse input terminal functions to control direction.

Set the following parameters in the ASD controller for testing purposes:

Parameter	Setting	Notes
BF01 contrl mode	V/Hz control	
BF07 auto ident.	on	
ME04 rev phas. sq	off	
AI11 en.neg.ana1	enable	
AI12 scale ana1	10V	
AI13 offest ana1	0V	
FR01 spd SP src	Analog in1	
FR02 spd SP typ.	percentage	Percentage has no impact to the analog input scaling. The analog input scaling is defined by FR05
		Use FR04/06/07 to limitate the speed range (The limitation works in all operation modes). FR07 is typically below 0.
FR10 accel time	20 sec	
FR11 decel time	20 sec	
FR20 jog SP src.	fix value	
FR21 jogSpeedFWD	10	
FR22 jogSpeedREV	10	
TI01 fnc.Enable	inverter enable	
TI02 lvl.Enable	active high	
TI03 src.Enable	terminal	
TI05 fnc.DigIn1	Ls-identification	
TI06 lvl.DigIn1	active high	
TI07 src.DigIn1	terminal	

TI09 fnc.DigIn2	satur.identifica	
TI10 lvl.DigIn2	active high	
TI11 src.DigIn2	terminal	
TI13 fnc.DigIn3	RPR-ident.start	
TI14 lvl.DigIn3	active high	
TI15 src.DigIn3	terminal	
TI17 fnc.DigIn4	forward	
TI18 lvl.DigIn4	active high	
TI19 src.DigIn4	terminal	
TI21 fnc.DigIn5	reverse	
TI22 lvl.DigIn5	active high	
TI23 src.DigIn5	terminal	
TI25 fnc.DigIn6	run	
TI26 lvl.DigIn6	active high	
TI27 src.DigIn6	terminal	
LT10 max.current	To suit the motor overload current rating	
LT11 overcurrent	110% of the motor overload current rating	



Check that there is no mechanical load connected to the motor shaft and that the motor shaft is free to turn.

Check that the motor frame is securely bolted down or otherwise restrained against movement. Sudden acceleration an unrestrained motor may cause it to move unexpectedly or roll over!

Check that any key in the motor shaft is removed or secured

Check that rotation of the motor shaft will not create a hazard



Check that it is safe to apply power to the system

Test run

Set the controls wired to the ASD control board as follow:

Control	Function	Setting
Switch SWE	Enable	off
Switch SW1	Ls identification start	off
Switch SW2	Saturation identification start	off
Switch SW3	Rotor pole position identification start	off
Switch SW4	forward	off
Switch SW5	reverse	off
Switch SW6	run	off
Switch SWD	Direction selection	Forward
Potentiometer VR1	Speed reference	10%

Apply power and wait for the power stack capacitors to be charged

Check the following measurements using the HMI

Parameter	Measurement	Notes
IW76 line volt	AC line voltage	Verify that the displayed value agrees with the actual measured value.
IW20 dc-link vol	DC link voltage	Verify that the displayed value agrees with the actual measured value. If the displayed DC link voltage is incorrect, check the value of LT05.
IW22 HS-temp.1	Heat sink temperature 1	Verify that the displayed

		temperature is reasonable
IW31 PCB temp	ASD controller PCB temperature	Verify that the displayed temperature is reasonable
IW32 drive state	Drive state	Verify that the inverter state is “enabled”. If it is not, find the cause and rectify it before proceeding further.
AI17 value ana1	Analogue input 1	Should indicate approximately 10% according to the setting of VR1
	LEDs on HMI	The green LED on the HMI should be on. Other LEDs on the HMI should be off.

Start the ASD controller by setting SWE (enable) and SW4 (forward) to ON and then setting SW6 (run) to ON

The motor should begin to rotate at approximately 10% of rated speed. Observe the direction of rotation. It should be rotating the forward¹ direction. If it is rotating in the reverse direction, stop the drive, switch off the power and wait for DC link capacitors to discharge. After you have verified that it is safe to do so, exchange the connections to two of the motor phase. Alternatively, parameter ME04 rev phas. Sq may be set to “on” to achieve a similar result.

¹ The International Electro technical Commission (IEC) definition for forward rotation is clockwise looking at the shaft (drive) end of the motor. You are free to make your own definition of forward to suit your own application, however unless there is some good reason to do otherwise, we suggest adoption of the IEC definition for consistency.

Check the following measurements using the HMI

Parameter	Measurement	Notes
Observe motor shaft	Motor rotation direction	Should rotate in what you consider to be the positive (forward) direction. For preference, use the IEC definition of forward rotation.
IW04 el.rot freq	Motor shaft sensor direction	The value should be positive . Change the shaft sensor wiring or use EC07 (or EA22 or EB22) ² cnt reversl to reverse the sensed direction if a negative value is shown.

Increase the setting of the potentiometer VR1 (speed) connected to analogue input 1. The motor speed should increase smoothly. Set the potentiometer to full clockwise to give maximum speed.

Check the following measurements using the HMI

Parameter	Measurement	Notes
IW01 out current	Stator current	Verify that the displayed value agrees with the actual measured value. Use a clamp on (or other) ammeter on one motor phase.
IW03 stator freq and IW04 el.rot freq	Check that motor frequency and shaft sensor speed agree	These values should agree exactly for PMSM, very close for an unloaded induction motor

² Use EC07 if the shaft sensor is an incremental encoder connected to the on-board interface or EA22 or EB22 if the shaft sensor is connected to expansion socket A or B respectively. See parameter list for more detail.

Motor parameter identification

Need to write a basic procedure to do the relevant id sequences (switch this, switch that, observe...) with reference to Parameter identification section for more detailed information.

On PMSM: execute Ls-identification and then RPR-identification

On IM: execute saturation+no-load-current-identification and afterwards Ls-identification.

Any change in the motor parameter section means you have to redo the identification procedure!

Test operation in closed loop speed control

Set UE01 to a value 10% higher the intended speed rang.

Stop the ASD system by setting switch SW6 (run) to OFF. Wait for the motor shaft to stop rotating.

Set parameter BF01 contrl mode to “speed control”

Start the ASD system by setting switch SW6 (run) to ON.

Use the potentiometer VR1 to adjust the speed reference and observe that the system is much more responsive than when previously operating in V/Hz control mode.

Reduce the acceleration (parameter FR10) and deceleration (parameter FR11) times to check that the dynamic response to changes in the speed reference is much faster. Settings of 0.5 seconds are suggested for this trial.

Set switch SWD to the “reverse” position and observe that the direction of motor rotation also reverses. It should be possible to alternate between “forward” and “reverse” settings of this switch while the motor is running at speed and see a smooth but rapid transition between forward and reverse motor shaft rotation. The HMI “limit” LED may illuminate momentarily during this process to show that some limit condition has been reached. The most likely limit conditions are voltage limit and current limit.

The dynamic of the speed control loop is defined by

- Current controller speed (SR02)
- Moment of inertia (MP24/DR03)
- Speed signal filtering (DR04)

-Speed controller adjustment parameters (DR01/DR02)

If the speed controller creates noise, increase the speed signal filter time. Typical values are:

0ms...1ms for hipurface and Sin/Cos encoders

0.5ms...4ms for incremental encoders and resolvers

*****End of basic testing*****

Position measurement

The ASD has 3 possible sources of position information. These are the incremental encoder interface on the control board and optional shaft sensor interface boards connected to expansion connectors EXP-1 and EXP-2 on the control board.

The optional shaft sensor interface boards that can be used with EXP-1 or EXP-2 are as follows:

Resolver interface

Incremental encoder interface

Hiperface interface

Heidenhain ERN1387 interface

SIN/COS encoder

Pulse train interface

Please note that certain internal signals are shared between EXP1 and the built-in incremental encoder interface ENC. It is not possible to use Hiperface, an incremental encoder, SIN/COS encoder or a pulse train interface on EXP2 at the same time as an encoder on the ENC port. A resolver or a current sensor interface³ may be used on the EXP2 interface at the same time as an incremental encoder on the ENC port.

Signals from each of the three possible interface locations are processed to provide data for the control algorithms. The data derived from position measurement sensors is as follows:

- (a) Motor electrical position. This is a position within one electrical cycle of the motor shaft used for motor control and is always scaled to be 360 electrical degrees = 65365 counts. The internal algorithms are processing position information with a resolution of 32bit (mechanical as well as electrical))
- (b) A multi-turn position measurement with a very large range (many turns) that is used for position control. This position can be scaled in arbitrary, user defined units.
- (c) A speed signal.

³ The current sensor interface is an optional part that connects to EXP1 or EXP2 and allows external motor current sensors to be used instead of the current signals from the power stack.

Shaft sensor	Motor electrical position	Multi-turn position	
Incremental encoder	Not suitable for PMSM, OK for IM		
Resolver			
Hiperface		Multiturn Hiperface encoders are available but ASD only supports single turn	
Heidenhain ERN1387	Z1 track and reference mark		
SIN/COS encoder			
Pulse train interface			

- (d) a preset position (which parameter sets this?) when the active edge of the index pulse occurs.
- (e) and is reset to zero at power up.

When an index occurs, the position information that includes the multi turn position changes at the same time with the single turn position up to +/- 0.5 revolutions.

There is a spare parameter (EA20) that we can be if you want the single turn position be reset to a different value other than 0. The function is not implemented. If this is important for you, I can set this topic on top of my list.

Note also that the multi turn position information is reset to 0 after power up. The only way to change this to execute a homing operation. The parameters LM25 and LM26 are then to define the position of 'home'

- (c) A speed signal. The speed signal is not influenced by any index pulse activity.

Right

. In the case of the measurement device being an incremental encoder or a sin/cos encoder with more than one cycle per electrical motor shaft revolution, there is an index pulse (Z1). When the active edge of the index pulse occurs this position is reset to an angular offset position (eg EA19). The first time that this active edge occurs, there may be a sudden jump in measured position of up to 180 degrees in either direction.

Look at the description to the Z1 track above. A thing to notice is that the actual electrical position is not displayed to the user. Only after rotor position reference identification he gets to see an angle value. If you like, I can easily add an actual value that shows the rotor

position all the time. Just for your information: internally we calculate with 32bit single turn positions, but 16bit for rotor position reference is plenty.

Configuring and testing position control modes

Position control modes

Review position control modes

Position references

Sources of position reference

Position feedback

Sources of position feedback

Selection of motor shaft sensors

The motor shaft sensors used with the ASD may be classified into two families:

Absolute angle sensors

An "absolute" sensor maintains angle information when power is removed from the system. The position of the encoder is available immediately on applying power. The relationship between the encoder value and the physical position of the shaft is set at assembly and remains fixed. A common example of an absolute angle sensor is a resolver. A resolver provides signals proportional to the sine and cosine of the shaft angle. The actual angle is calculated within the ASD from these signals using the well-known trigonometrical relationships. The ASD provides interfaces for resolvers and several other type of absolute shaft angle sensors.

An important specification of an absolute shaft angle sensor is the range of angles that are measured before angle measurement repeats itself. For resolvers, this may be one turn (360°), some fraction of a turn, depending on the resolver pole number. There are other sensor types that can provide angle information that is absolute over more than one turn, in some case a very large number of turns. Sensors using the Hiperface interface are an example of these capabilities.

Incremental angle sensors

An incremental shaft sensor simply provides information about shaft movement. One common arrangement is to provide a certain number of pulses per revolution of the shaft. The interface electronics (inside the ASD controller) simply counts the pulses to determine the amount of shaft

movement. In a practical sensor of this type it is usual to provide two sets of pulses (so called A and B channels) so that the direction of shaft movement can also be determined.

It is quite common to provide a separate output from this type of sensor that gives just one pulse per revolution (usually called the index or Z channel). This feature makes it possible for the ASD controller to know the absolute position of the shaft and therefore mimic the function of an absolute angle sensor, but only after the shaft has rotated past the index pulse position at least once since the system was powered each time. This arrangement is often used for position control of CNC machinery by incorporating some kind of “find home” sequence into the machine operation each time the system is powered up.

It is theoretically possible to employ a similar arrangement with the index pulse to use an incremental shaft angle sensor instead of an absolute one for synchronous motor control purposes. The need for initial rotation of the motor shaft past the index location before the motor can be controlled properly generally means that this arrangement is not practical for real applications but perhaps useful for testing purposes if a more suitable sensor is not available.

The ASD potentially uses information from the motor shaft sensor for three different purposes:

1. As an input to the basic motor control algorithm that controls flux and torque.
 - a. In the case of an induction motor (IM), the control algorithm requires information about the incremental movement of the motor shaft, so either an incremental type of shaft sensor is suitable.
 - b. In the case of a synchronous motor (PMSM), the algorithm requires information about the present angle of the shaft relative to the motor frame (stator). This requires a shaft sensor that can provide absolute angle information.
2. To provide a measurement of shaft speed for the closed loop speed control operating mode of the ASD. An incremental shaft sensor is sufficient for this purpose.
3. To provide a measurement of shaft position for the position control operating modes of the ASD. An incremental shaft sensor may be sufficient or an absolute shaft sensor may be necessary depending on the position control application.

The ASD has the option of using a separate encoder for position control purposes. Where a separate encoder is used, the requirements are determined purely by the position control application requirements and are independent of the motor control and speed control considerations described above.

Resolver selection for PMSM applications

The chart below shows the compatibility of various pole numbers of PMSM and resolvers. This chart is only valid for resolvers mounted directly on the motor shaft or with a 1:1 ratio mechanical connection. Other aspects of electrical and mechanical compatibility are not considered here.

		PMSM pole number				
		2	4	6	8	10
Resolver pole number	2	Best choice	OK	OK	OK	OK
	4	NO!	Best choice	NO!	OK	NO!
	6	NO!	NO!	Best choice	NO!	NO!
	8	NO!	NO!	NO!	Best choice	NO!
	10	NO!	NO!	NO!	NO!	Best choice

General rules:

PMSM pole number = (Resolver pole number) x n

n can be 1, 2, 3, 4, 5.....

Smaller values for n give better angle measurement resolution and therefore better performance.

Motor Parameter Identification (Auto-tuning)

Induction motor systems

There are two separate functions to automatically identify induction motor parameters.

1. Controller gain ('Ls-identification')

This function determines the value for parameter SR03 Current controller proportional gain correction. The motor shaft does not rotate during this identification process.

2. Saturation/No-Load-current identification ('satur. Identific')

This function determines values for the following parameters

MP07 No-load current⁴

SA01 Saturation degree at 50 % rated magnetizing current

SA02 Saturation degree at 75 % rated magnetizing current

SA03 Saturation degree at 100 % rated magnetizing current

SA04 Saturation degree at 125 % rated magnetizing current



WARNING During this identification process the motor will rotate at approximately half of rated speed. Before beginning this process, disconnect any mechanical load from the motor shaft and ensure that the motor shaft is free to turn without risk of injury to persons or damage to property. Remove or secure any key fitted to the motor shaft.

Synchronous motor (PMSM) systems

There are two separate functions to automatically identify induction motor parameters.

3. Controller gain ('Ls-identification')

This function determines the value for parameter SR03 Current controller proportional gain correction. The motor shaft does not rotate during this identification process.

4. RP-identification

This function identifies the offset angle between the rotor magnet and the shaft sensor zero position.

WARNING During this identification process the motor shaft will possibly rotate in either direction by up to half a turn. Before beginning this process, disconnect any mechanical load from the motor shaft and ensure that the motor shaft is free to turn without risk of injury to persons or damage to property. Remove or secure any key fitted to the motor shaft.

Activating the identification processes

Each of the identification processes is controlled by assigning the function to a digital input and then operating that input.

Choose an unused digital input and assign the required to identification function to it. See parameters TI05 – T147 regarding the assignment of functions to digital inputs. The available identification functions are:

⁴ This function may fail if the initial setting of MP07 is too low. In the case that you don't know the no load current of the motor, set MP07 to 50% of the motor rated current before starting the identification process.

Ls-identification (for all motors)

RP-identification (for PMSM only)

satur. Identific (for IM only)

Do not assign an identification function to the enable terminal.

For example, you may choose to assign input 6 (terminal DIN6) to the 'Ls-identification' function, set parameter TI25 to 'Ls-identification'. Other input terminals may have other identification tasks assigned to them as required.

Before proceeding with the parameter identifications decide if you want the identification results to be save in non-volatile storage (EEPROM). Set parameter BF07 auto ident to 'on' to automatically save the results to non-volatile storage. This should be done unless there is some special reason to not save the new identification results.

To start and end an identification process:

1. Enable terminal must be low (0).
2. The digital input terminal that has the required identification function assigned to it must be set high (1) while the Enable terminal is low (0)
3. Set the Enable terminal to high (1)
4. SW6 run must be set high (1).
5. The identification process for the selected parameter will proceed automatically. The green LED on the HMI will blink during the identification process. The status of the identification process can be checked as an *Actual Value* (IW38 etc) on the HMI.

Process	Actual value for process status	Display for identification in progress ⁵	Display for identification complete	Actual value display for result
Ls-identification	IW38 Ls-id.state	Initialisation (Green LED on HMI blinks)	LS-ident accomp	SR03 prop.g.corr
RP-identification	IW40 RPR id.stat	Initialisation (Green LED on HMI blinks)	RPR ID ready	EC09 ⁶ or EA24 ⁷ or EB24 ⁸ rot.pos.ref
satur. Identific	IW56 sat100	Initialisation (Green LED on HMI blinks)	LshSatID accomp	MP07 no load cur SA01 S-levl 50%M SA02 S-levl 75%M SA03

⁵ Each parameter shows descriptions of the various internal steps during the identification (there are a few steps, not important for the customer). Before the identification is started, the parameter displays 'inactive'. After the transition of the associated input terminal from 0 to 1 it displays 'initialization'. An error message 'Wrong motortype' will be displayed when the motor type does not support the selected identification algorithm (eg attempting RP-identification for an induction motor) Parameter IW32 drive state also shows if an identification process is active.

The status conditions 'Ident active' and 'Ident finished' are also available for use with digital outputs.

⁶ EC09 is used if the motor shaft sensor location (ME01 mot.enc.src) is selected as onboard (ENC)

⁷ EA24 is used if the motor shaft sensor location (ME01 mot.enc.src) is selected as expansion EXP1

⁸ EB24 is used if the motor shaft sensor location (ME01 mot.enc.src) is selected as expansion EXP2

				S-levl100%M
				SA04 S-levl125%M

6. The green LED will stop blinking when the process has finished. After process is finished, set the enable input, SW6 run and the input terminal associated with the identification function low (0).
7. Repeat the process for the other required identification function.

Note that the above description is based on the various inputs being configured as active high (default). The identification process will terminate immediately if the En-input becomes inactive.

Once an identification process has started, it will continue indefinitely⁹ until terminated by the completion of the identification, setting the En-input to 0 or an error. If an error (overspeed or overcurrent etc) occurs during the identification process, the identification process will stop and the error will be indicated with the usual message for that error. Checking the limit LED on the HMI and the limitation state IW33 may be of assistance in fault finding motor identification process problems.

⁹ There is no timeout on the identification processes. Take care to supervise the identification processes closely.

MSC-3 Power stacks

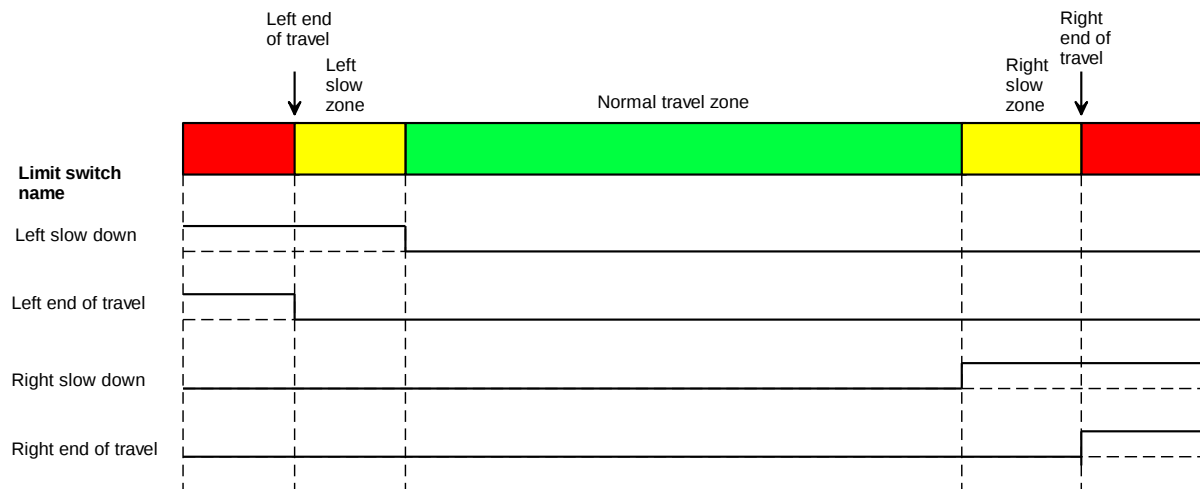
Setup for use with MSC-3R power stacks using standard current sensors, calibrated according to the standard specification for the particular MSC-3 model.

Parameter	Short name	Name	Setting
LT01	switch.freq	Switching frequency	4000 Hz (def)
LT02	min.turn tm	Minimum turn-on and turn-off time	3 us (def)
LT03	interlock	Interlock dead time	3 us
LT04	rat. Vmains	Nominal grid voltage	380, 400, 415V as required
LT05	Vdc ref.vol	DC link voltage scaling	808 V (def)
LT06	Vdcmax dis.	DC link over-voltage trip threshold	805 V
LT07	srcCur.sens	Current sensor signal source	Auto select (def)
LT08	curr.normal	Current signal normalisation	(1.848 to 2.22) x LT10 MUST match hardware current scaling
LT09	nb.cur.sens	Number of current sensors	2 (def)
LT10	max.current	Current limit	RMS overload current (highest value at any speed). This parameter is intended for power stack protection. See MP06 which has a similar function to allow the user to set a current limit.
LT11	overcurrent	Overcurrent disable threshold	1.2 x LT10
LT12	Corr.lu	Measurement correction lv	100% (def)
LT13	corr. lv	Measurement correction lv	100% (def)
LT14	trq.lim.Vdc	Voltage limit proportional range	20V (def)
LT15	t0-thrs.Vmx	Voltage limit (high) threshold	780 (def)
LT16	t0-thrs.Vmn	Voltage limit (low) threshold	0V (def)

LT17	Idc max.	Maximum DC current (positive)	1.35 x LT10
LT18	Idc min.	Maximum DC current (negative)	-1.35 x LT10 Note negative value!
LT19	PS I2t current	I2t Current setting	Set to the continuous RMS current rating of the drive or the continuous motor FLC, whichever is less
LT20	PS I2t freq	Frequency threshold for current derating at small stator frequencies	To suit motor ratings
LT21	PS I2t @0Hz	Current derating factor at stator frequency 0 Hz	To suit motor ratings
LT22	Mot.I2t cur	I2t Current setting for motor protection	
LT23	mot.I2t frq	Low frequency threshold. For motor protection	
LT24	mot.I2t@0Hz	Current derating factor at stator frequency 0 Hz	
LT25....32	HS-tmp_ typ	Power stack temperature sensor _ type	LM335 (10mV/K temperature sensor)
LT33	PS code	PS Code	To match power board ID code ¹⁰
LT34	pwr supply	Power supply type	3 phase supply
LT35	invertr.typ	Inverter family type	Chassis A/B/C (def) Or Chassis D

¹⁰ The power board schematics shows which code is used.

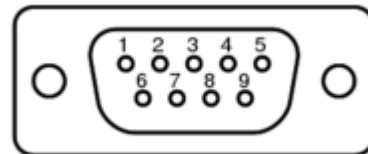
Limit switches



CANBUS

9-PIN (MALE) D-SUB CAN BUS PINOUT

Pin Number	Signal Name	Signal Description
1	No Connect	N/A
2	CAN_L	Dominant Low
3	GND	Ground
4	No Connect	N/A
5	No Connect	N/A
6	GND	Ground
7	CAN_H	Dominant High
8	No Connect	N/A
9	No Connect	N/A



Controller parameters

SR01	setl.time d		
SR02	setl.time q		
SR03	prop.g.corr		
SR04	int.g.cor.d		
SR05	int.g.cor.q		
SR06	Rs correct.	100%	
SR07	Vdc compen.	on	
SR08	timecon.Vdc		
SR09	deadtimecmp		
SR10	two-phase-m	off	
SR11	freq. TPM	Not relevant if SR10 off	
SR12	overmod.lvl	0%	
DR01	prop.g.corr	100%	
DR02	int.g.corr.	100%	
DR03	J-correct.	100%	
DR04	speed fil.	1ms	
LR01	prop.g.corr		
LR02	speed ff		
FS01	f.weak.IM	off	
FS02	f.weak.PMSM		
FS03	fw char.low		
FS04	isdmax corr		
MG01	imag corBSR		
MG02	imag.thres.		

MG03	TR-red.fact			
TA01	TR-method	inactiv		
TA02	int.g.corr.			
TC01	input src			
TC02	invert trq.			
TC03	prop.g.corr			
TC04	int.g.corr.			
TC05	trq.imp.abs			
TC06	trq.scaling			
TC07	trq.offset			
TC08	torque fil.			
RE01	double.cntr	off		

HM03 Measured Value Acquisition

Expansion sockets

	ME01 mot.enc.src	Onboard (ENC)	
	ME02 load.en.src	No encoder	
	ME03 res.min.exc		
	ME04 rev.phas.sq	off	
	ME05 RPR-id curr		
	ME06 Ls-id curr		
EA01 expans.type	EB01 expans.type		
EA02 IE line cnt	EB02 IE line cnt	EC01 IE line cnt	
EA03 IE timrMeas	EB03 IE timrMeas	EC02 IE timeMeas	
EA04 IE TtimMeas	EB04 IE TtimMeas	EC03 IE TtimMeas	
EA05 IE alarm	EB05 IE alarm	EC04 IE alarm	
EA06 index eval.	EB06 index eval.	EC05 index eval.	
EA07 RES polpair	EB07 RES polpair		
EA08 RES err.cor	EB08 RES err.cor		
EA09 RES exc.phs	EB09 RES exc.phs		
EA10 RES exc.RMS	EB10 RES exc.RMS		
EA11 RES exc.lim	EB11 RES exc.lim		
EA12 RES UV Vin	EB12 RES UV Vin		
EA13 RES n windw	EB13 RES n windw		

EA14 HPF periods	EB14 HPF periods		
EA15 Sin/Cos prd	EB15 Sin/Cos prd		
EA16 Z1 track	EB16 Z1 track		
EA17 cnt.reversl	EB17 cnt.reversl	EC06 cnt.reversl	
EA18 signl.delay	EB18 signl.delay	EC07 signl.delay	
EA19 rot.pos.ref	EB19 rot.pos.ref	EC08 rot.pos.ref	
EA20 encIndx.ref	EB20 encIndx.ref	EC09 encIndx.ref	
EB40 nb.cur.sens			
EB41 curr.normal			
EB42 curr.ref			
EB43 overcurrent			
EB44 corr. lu			
EB45 corr. lv			

Drive control

BF01	ctrl mode	V/Hz control (def)	
BF02			
BF03	brake cntrl	off	
BF04	act.frntend	off	
BF05	main contac	off	
BF06	mot.contact	off	
BF07	auto ident	on	
BF08	dyn.lim.sw	off	
BF09			
BF10	trq.lim.Vdc	on	
BF11	trq.lim.Idc	off	

References

FR01	spd SP src.		

I2C I/O expansion

There is provision to connect an external I/O expansion device to the internal I2C bus.

The I/O expansion device is accessed as if it is a PCF8575C Remote 16-bit I/O expander for I2C-bus IC. This provides for the addition of 16 digital inputs and 16 digital outputs.

See IC2xx parameters and the PCF8575C datasheet.

Additional topics

Once/twice control per switching interval

The following restrictions apply to the choice of this parameter:

PWM switching frequency	Once/twice control per switching interval	Notes
Xxx KHz or less	twice	
Xxx > Fsw > ZZZ	Optional, however twice recommended	

Zzz kHz or more	once	
-----------------	------	--

Modifications to MSC-3 circuits for use with ASD

These modifications change the characteristic of the low pass filter in the MSC-3 current sensor circuits to increase the bandwidth. This is necessary because a high dynamic controller such as ASD requires faster responding current measurement than a standard AC drive application.

Chassis	PCB assembly	Relevant schematic	Essential Changes	Additional changes for better performance
CHA	B1070xx	I107002W I107003W I107004W I107005W I107006W	C103 = 1n (was 10n) C105 = 1n (was 10n)	R21 = 10k (was 160k) R26 = 10k (was 160k) C60 = 100p (was 10n) C62 = 100p (was 10n)
CHB	B1090xx	I109002V I109003V I109004V I109005V I109006V	C60 = 1nF (was 10nF) C62 = 1nF (was 10nF)	
CHC	B1124xx	I112402H I112403H	C1 = 1nF (was 10nF) C4 = 1nF (was 10nF)	From Andreas 18 Mar 2014 R1: 10k (was 160k) R8: 10k (was 160k) C1: 100p (was 10n) C2: 100p (was 10n) C4: 100p (was 10n) C5: 100p (was 10n) C3: 470p (was 10n) C6: 470p (was 10n) Especially for senseless applications with

				PMSM we need as much bandwidth as possible so I recommend somebody is checking the values above.
CHD and other products based on the same power interface board (PIB)	B1106xx	I110602G I110603G I110604G I110605G I110606G	C60 = 10pF (was 10nF) C61 = 10pF (was 10nF) C68 = 10pF (was 10nF) C69 = 10pF (was 10nF) C65 = 1nF (was 10nF) C73 = 1nF (was 10nF)	

For CHD.... Information from 400kW dynamometer project

PIB board:

C58, C59, C66, C67 change to 100pF from 1nF

C60, C61, C68, C69 change to 47pF from 10nF

C65, C73 change to 1nF from 10nF

R39, R40, R41, R42, R46, R47, R48, R49 change to 10K from 160k

Configuration resistors: R43, R50 are 27K, R44, R51 are 12K

Current controllers settling time: 2ms

ASD User Levels

User Level	Name	Purpose	Typical items
0	Operator	End product machine operator	Only items necessary to operate final product. Maybe just display something.
1	Supervisor	End product machine supervisor	Determine what level 0 can access Setting level 0 password
2	Technician	Advanced machine supervisor / Field service technician	Determine what level 0-1 can access Setting level 0-1 passwords
3	OEM configuration	Configuration by OEM customer	Motor parameters Motor protection parameters Most other parameters Determine what level 0-2 can access Setting level 0-2 passwords
4	Product configuration	Application configuration by GNDC/QD	Parameters that need to be restricted for commercial reasons or to avoid the product appearing too complex to the OEM. Determine what level 3 can access Setting level 3 password
5	Hardware protection	Hardware protection settings (to match power stack) by GNDC/QD	• Maximum current limit • Temperature sensor types • Heatsink trip temperatures

			• Other power stack temperature trips • Maximum switching frequency • Maximum I2t setting • Disallow any mode that does not provide current limit • Parameters related to calibration of power stack measurements of current & voltage • Any parameter that is necessary to prevent power stack damage
6	R&D	To set parameters for some special or experimental feature	
7	Debug	Software debug (no user access)	For software R&D purposes only

Notes:

1. User levels 3, 4 & 5 will always be used. Use of other user levels 0-2 is optional.
2. Any user level can also access anything assigned to a lower user level
3. All parameters are available to level 3 except for those related power stack protection and any parameters that there is some particular reason to hide (commercial, complexity etc).
4. Hardware protection parameters (set at level 5) MUST match the associated power stack design.

Parameter setup for torque limitation

Fast responding torque limitation can be used in all current controlled operations like position/speed and torque control. It limits the maximum output current according to the DC-link voltage (Vdc) to effectively prevents it from being over- or under-charged.

Parameter setup for torque limitation in case of Vdc high or low

To enable torque limitation, the associated drive control function must be set to *on*:

Main menu: HM05 drive ctrl.		
Sub menu: AS01 oper. manag		
BF10 trq.lim.Vdc	<i>on</i>	select torque limitation if Vdc is high or low

The voltage thresholds for torque limitation can be found in the power stack menu:

Main menu: HM08 power stack		
LT14 trq.lim.Vdc	50V	Torque limitation range in V
LT15 t0-thrs.Vmx	450V	Value of Vdc at which (positive) torque values are set to zero
LT16 t0-thrs.Vmn	700V	Value of Vdc at which (negative) torque values are set to zero

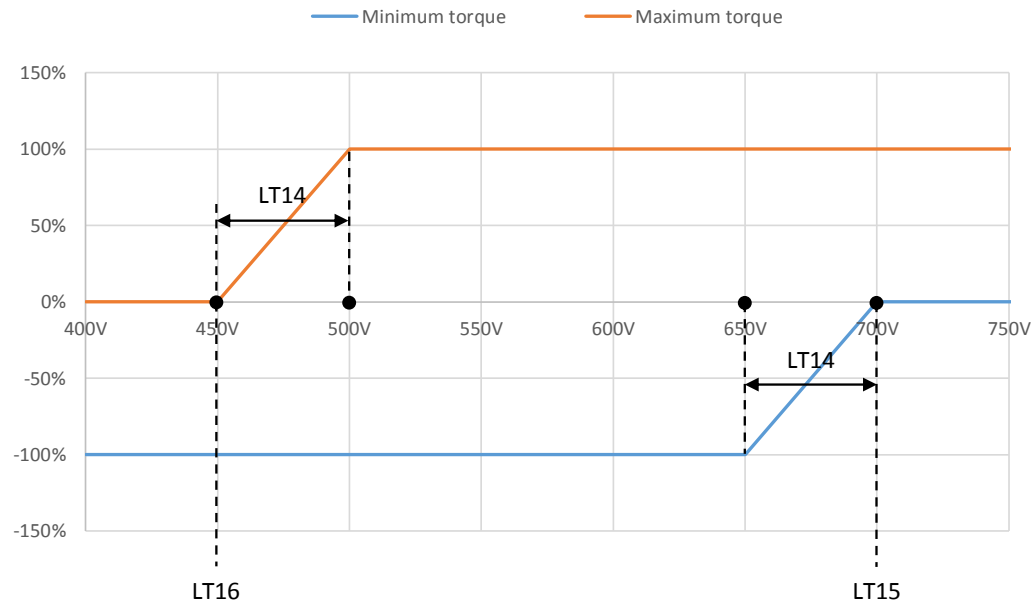
Function of torque limitation

The function of torque limitation is a linear interpolation between zero and maximum torque.

During Motoric operation, full torque is available above a Vdc voltage value of LT16+LT14 whereas zero-torque is allowed below the voltage level LT16.

During Regenerative operation, full torque is available below a Vdc voltage value of LT15-LT14 whereas zero-torque is allowed above the voltage level LT15.

The Diagram below shows both maximum and minimum torque limitation values in % of the maximum allowed torque magnitude. The maximum allowed torque magnitude is basically defined by the maximum Output current and the actual magnetizing state of the motor.



Software update using the ZAP updater

1. Connect the ZAP updater to the I2C connector to a pre-programmed control board
2. Power the control board with 24V or connect the ZAP updater to the PC to provide power supply via USB
3. Activate save guards operation mode:

In the **Setpoints** menu, set **SO23 save guards** to **1**

All outputs are set to logical zero. PMW is deactivated.

If this mode was activated by accident it can only be left by a software restart.

4. Connect the ZAP updater to a PC if not yet done
5. Activate the software upload function:

In the **Setpoints** menu, set **SO24 SW load USB** to **1**

6. A new mass storage device will appear in the file manager of the PC. Copy update.ini and *.DIS into this device.
7. Use “eject removable disc” after the file transfer. The control board will restart.
8. Activate save guards operation mode again:

In the **Setpoints** menu, set **SO23 save guards** to **1**

9. Execute the software update function:

In the **Setpoints** menu, set **SO25 SW update** to **1**

The software update procedure will take place. This might take a while. Wait until the control board restarts.

ASD - How to use the simulator feature v0.002

The ASD software version r10v020 is containing a simulator that allows to simulate any induction or PMSM. It can be very useful to test control functions

Before you use simulation mode, read this document carefully. A misuse easily can cause damage to motor and inverter.

Various things to know

The internal motor simulator uses the rated values that are configured in the menu 'HM01 motor param' to simulate a motor with exactly the same behavior.

The simulator uses uncorrected (not modified) as well as corrected (modified) physical values.

Inertia J and **saturation characteristic** are corrected values. That means that you can change the inertia and the saturation characteristic of simulate motor by modification of the relevant values.

(Inertia correction factor = 'DR03 J-correct.'; saturation characteristic = 'SA01 S-levl 50%M' to 'SA04 S-levl125%M').

Stator resistance Rs and **stator inductance Ls** are uncorrected values. Correction factors other than 100% will decrease the control quality during simulation. These correction factors on the other hand can be used to simulate the impact of a wrong stator resistance or inductance.

(Stator resistance correction factor = 'SR06 Rs correct.'; leakage inductance correction factor = 'SR03 prop.g.corr').

The **automatic parameter identification** functions can be executed during simulation mode. Identified values are not stored into the EEPROM. Note that a manual modification of the corresponding values or the command 'SO30 EPROM save' still causes an EEPROM storage.

Note that for closed loop application, a suitable motor encoder must be configured whether it is connected or not. Exceptions are encoderless operations.

The easiest way of testing dynamic behavior is to monitor simulated values with the analog outputs.

Setup

The configuration parameters of the simulator is located in the main menu *HM20 SIMULATOR*.

Digital and analog output terminals are controlled. Make sure that there are no external devices connect

1.1. Simulator is off

SM01 simulator	off
----------------	-----

Normal operation. Simulator is not used.

1.2. Simulator is in safe mode

SM01 simulator	on – safe mode
----------------	----------------

Motor Simulator is on.

The power stack is inactive. All physical values like DC link voltage or power stack temperature is simulated. PWM is NOT applied to the outputs.

Recommended simulation mode if control board is connected to a power stack or if the control board is not connected to a power stack and

1.3. Simulator controls active power stack with PWM (!ACTIVE PS!-mode)

SM01 simulator	on – !ACTIVE PS!
----------------	------------------

Motor Simulator is on.

The power stack is active like during normal operation.

Physical actual hardware relevant values like DC link voltage or power stack temperature are NOT simulated.

PWM is applied to the outputs!!!

!!! NEVER use this simulation mode if a physical motor is connected to the power stack !!!

1.4. Simulator runs on a standalone-control board (!FULL SIM!-mode)

SM01 simulator	on – !FULL SIM!
----------------	-----------------

Motor Simulator is on.

Control board outputs are active but physical input values are simulated.

!!! NEVER use this simulation mode if the control board is connected to a power stack !!!

PWM signals and other power stack communication signals are applied to the outputs

This mode can be useful to test the PWM outputs of the control board itself

(document ASD - PWM signal verification during simulation mode)

What else to consider...

- You need to restart if you changed the simulator configuration. The configuration is only loaded during start-up.
- You need at least HMI user level 5 during startup (7 for software version r020b). An active simulator feature at a lower HMI user level during startup will produce an error.
- the simulator uses a virtual incremental encoder on EXP1. You don't have to select that encoder but you have to setup a sensible line count number for it (EA02).
- You can use all types of physical load encoders if you want to simulate position difference control except an incremental encoder on EXP1
- You need +/-12V for X3 to make analog inputs work and also for some encoder evaluations. The simulator will not generate errors if these voltages are missing, so you might wonder why the analog inputs will not work.

Using simulation mode to verify the PWM outputs of ASD with software version r020

More information to the simulation mode can be found in the document ASD – Simulator function.

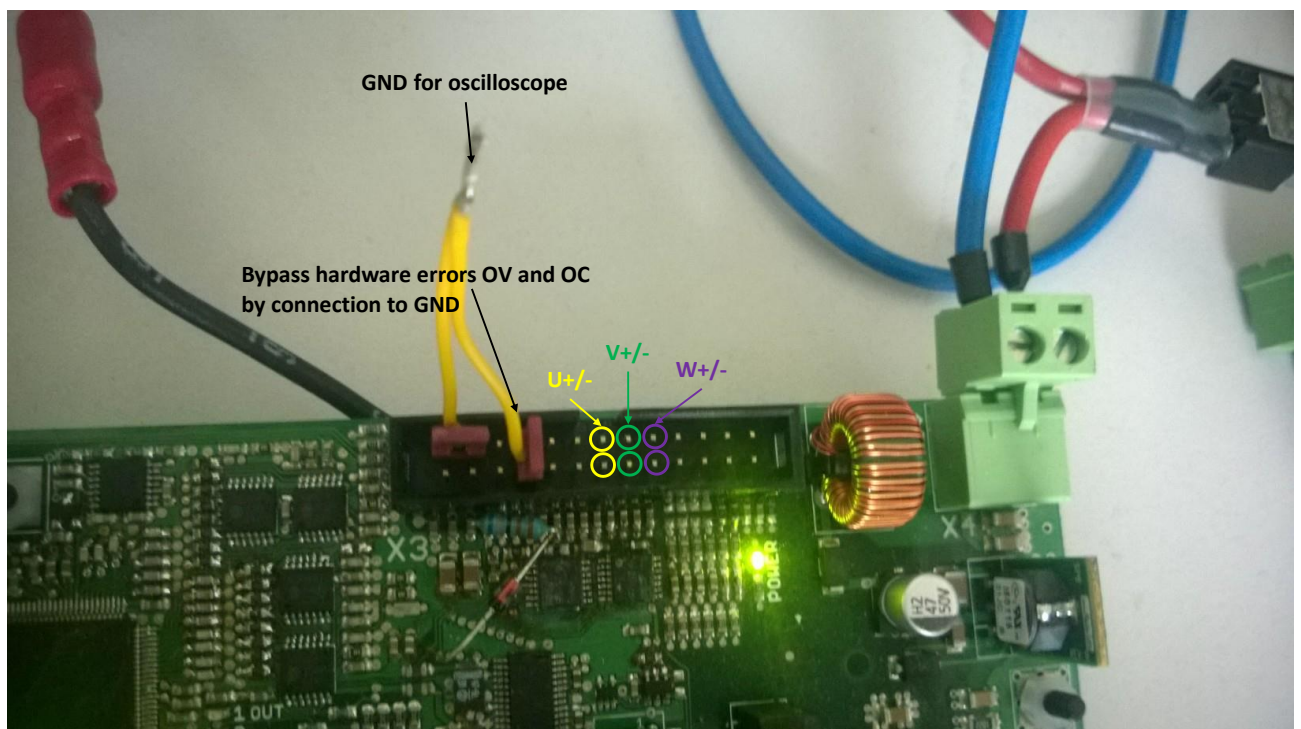
Restrictions in Software version r020

Simulation mode in software r020 only works properly if the power supply type (LT34 pwr supply) is set to *3 phase supply*. This is fixed in version r020b and higher.

During simulation, change LT34 pwr supply to *3 phase supply*. Do not forget to change it back after simulation. If normally 565V DC supply is used, it then might be necessary also to set the input voltage parameter (LT04 Input volt.) to 400V.

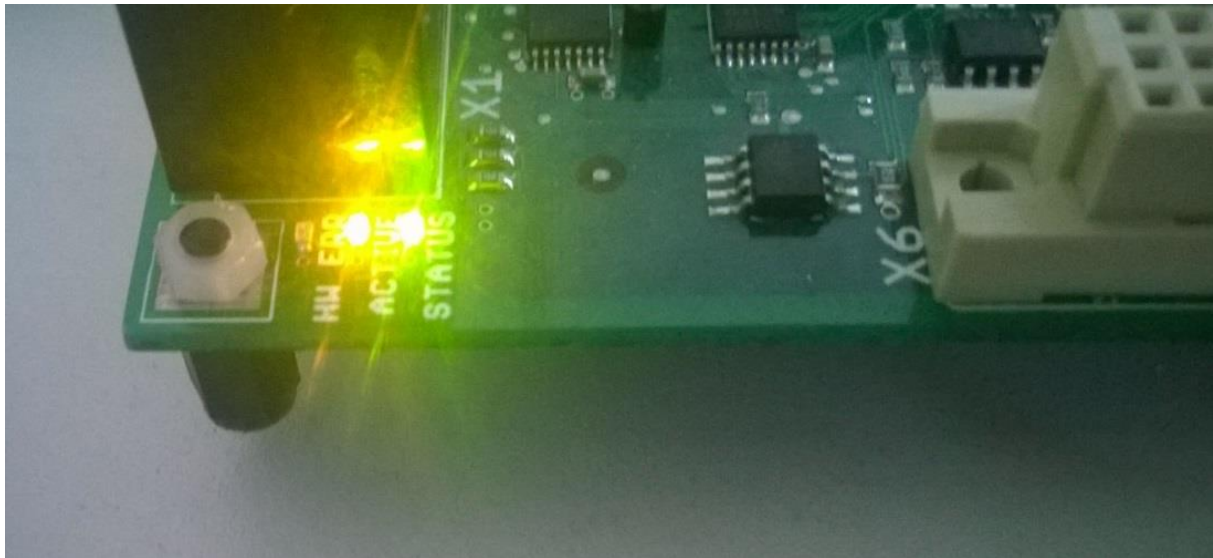
Verify the function of the PWM outputs on the control board

- (1) Disconnect the control board from the power stack. Disconnect also all peripherals systems from the control board outputs.
- (2) Because hardware errors will disable the PWM, these errors need to be bypassed. Bridge the hardware errors by connecting X3-pins 19, 20, 23 and 25 together like displayed below.

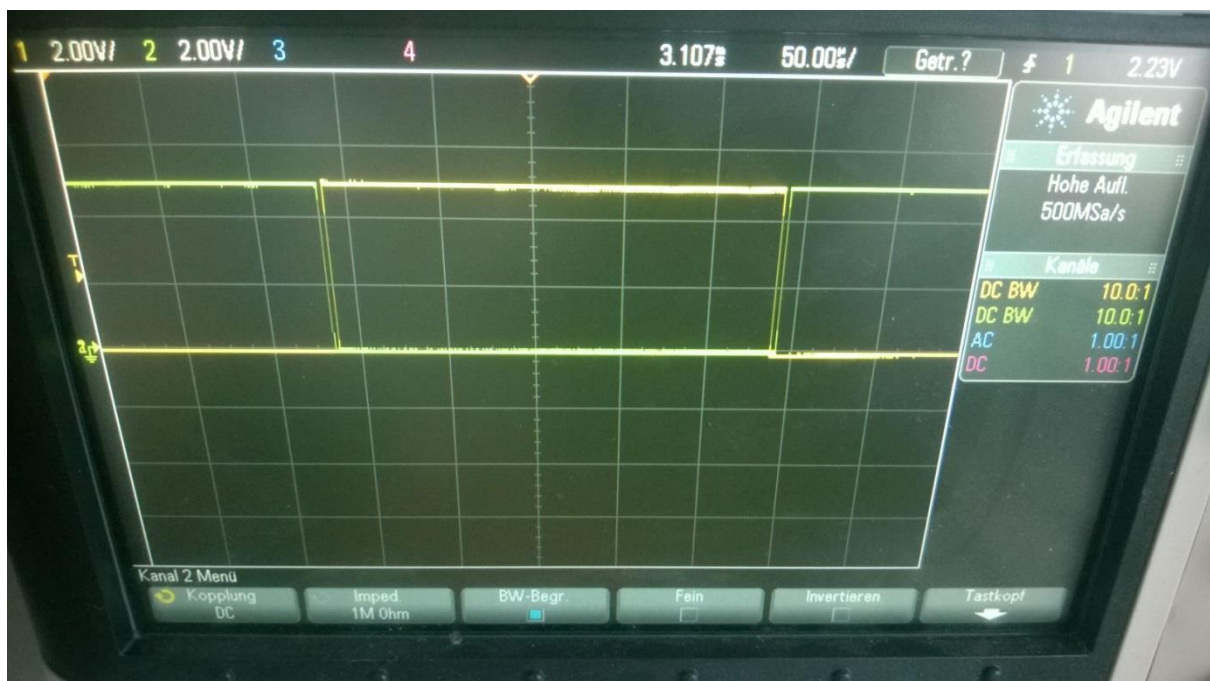


- (3) If done correctly, a software restart should switch off the hardware error latch. The red LED 'HW ERR' should go off. If not, make sure that no current sensor expansion board is connected to EXP2. If it still does not work, connect 5V to X3-21 (only necessary if R732 is assembled).
- (4) Switch the simulator (HM20->SM01 simulator) to simulation mode *on - !FULL SIM!*.
- (5) Restart the software and make sure that the parameter configuration is suitable to run a motor in principle. All possible operation modes can be used in general.

- (6) Enable the drive. Now the yellow LED 'ACTIVE' goes on and the green LED 'STATUS' starts blinking. The control board now simulates a virtual motor and PWM is generated according to the simulation.



- (7) Monitor the PWM signals U+,U-,V+,V+,W+ and W- on connector X3-pins 9-14. Monitor the upper and lower-side PWM signals simultaneously. The signals should look like in the picture below. Optionally check the interlock dead time if the real value matches the setup.



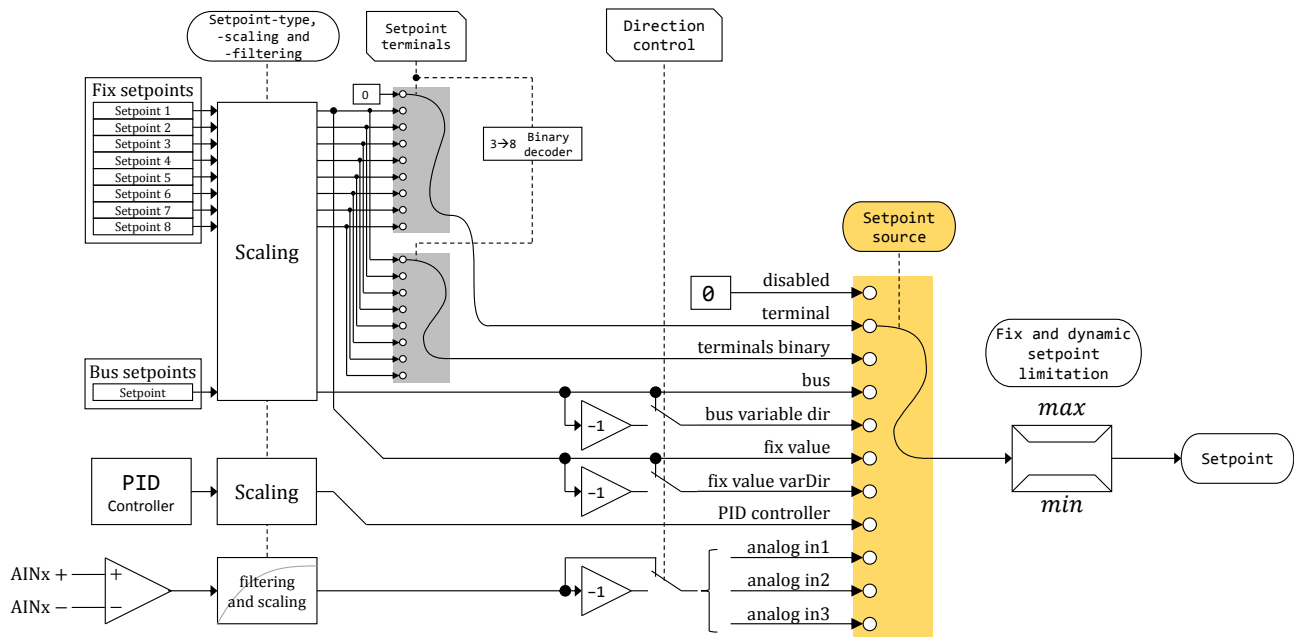
- (8) If finished it is very important to switch off simulation mode immediately. If it is still configured when a drive is connected, it may cause serious damage.

ASD - setpoint selection

Setpoint selection and processing

Selection and processing of setpoints is following the same principle for jog-mode-, AC-source-, position-, PID-, speed- and torque-setpoints. This document should help to understand the general approach.

The drawing below gives a rough overview of the involved functions. Note that not all the functions are available in all operation modes. The parameter list gives more detailed information.



Configuration

There are fix parameters to configure the setpoint selection and processing functions. These parameters should not be changed during operation:

- Setpoint source (0)
- Setpoint type, scaling and filtering (0)
- Setpoint limitations (0)

During operation

Setpoint values during operation can be modified by:

- State of setpoint terminals (0)
- Direction change functions (0 / 0)
- Dynamic setpoint limitations (0)
- Bus setpoint values (0)

-Analogue input values (0)

Setpoint sources

One setpoint source parameter for each different sepoint:

Main menu	Submenu	Parameter	Parameter description
HM02 contr.param	RP09 PID control	PI01 PID SP src.	PID controller setpoint source
HM06 setp.proces	SW01 torq. proc.	DP01 trq.SP src.	Torque setpoint source
	SW03 speed proc.	FR01 spd SP src.	Speed setpoint source
		FR20 jog SP src.	Jog speed setpoint source
	SW05 posit.proc.	LM01 pos.SP src.	Position setpoint source
	SW08 AC volt.Src	VS01 Volt.SP src	Voltage setpoint source
		VS06 freq.SP src	Frequency setpoint source (in AC source operation mode)

Setpoint source disabled

The setpoint source *disabled* implies a setpoint value of zero. This does not mean that the drive is disabled.

Setpoint source terminals or terminals binary

Up to 8 digital inputs can be used to switch between up to 8 fix predefined, non-volatile setpoint values plus zero.

The table below shows the active setpoint value depending on the state of the input terminals and whether the setpoint selection is binary coded or not.

'0' = input is inactive; '1' = input is active; '-' = input state does not matter

Setpoint source	State of input terminal function...								Active setpoint (according to the setpoint source and the input terminal states)
	setpoint 1	setpoint 2	setpoint 3	setpoint 4	setpoint 5	setpoint 6	setpoint 7	setpoint 8	
terminals	0	0	0	0	0	0	0	0	zero (setpoint = 0)
	1	0	0	0	0	0	0	0	setpoint 1
	-	1	0	0	0	0	0	0	setpoint 2

	-	-	1	0	0	0	0	0	setpoint 3
	-	-	-	1	0	0	0	0	setpoint 4
	-	-	-	-	1	0	0	0	setpoint 5
	-	-	-	-	-	1	0	0	setpoint 6
	-	-	-	-	-	-	1	0	setpoint 7
	-	-	-	-	-	-	-	1	setpoint 8
<i>terminals binary</i>	0	0	0	-	-	-	-	-	setpoint 1
	1	0	0	-	-	-	-	-	setpoint 2
	0	1	0	-	-	-	-	-	setpoint 3
	1	1	0	-	-	-	-	-	setpoint 4
	0	0	1	-	-	-	-	-	setpoint 5
	1	0	1	-	-	-	-	-	setpoint 6
	0	1	1	-	-	-	-	-	setpoint 7
	1	1	1	-	-	-	-	-	setpoint 8

Setpoint source *bus or bus var*

The setpoint value is taken from the setpoint section of the ASD menu. The values there are stored in a volatile memory section and are zero after a reset. Use this setpoint source if setpoints need to be modified by bus during operation.

Setpoint source *fix value or fix value varDir*

Setpoint 1 is constantly used. See section 0 for more details. Fix value varDir also allows to reverse the sign of the setpoint value in case the motor is in reverse mode.

Setpoint source *PID controller*

The output of the PID controller is used as a setpoint.

Setpoint source *analog in1 to analog in3*

An analog input value is used as a setpoint.

Setpoint type, scaling and filtering

The setpoint values from the ASD menu structure are unitless. Parameters for setpoint **type** do define how the values are to interpret. Typical selections are:

Setpoint type	Description
---------------	-------------

Percentage	The setpoint is in % of the rated motor nameplate data
Fractional INT32	The setpoint value 2^{31} represents the external reference value (or 2^{15} revolutions in case of position control)
Nm/rpm/Hz/...	The setpoint is directly interpreted as a physical value in various possible units

Setpoints from other sources are scaled directly by the external reference value. For example: Full scale on an analog input or PID controller output of 100% is scaled to this value.

Setpoint filtering

Analog inputs are filtered and preprocessed. Therefore, each analog input has its own parameters. More informations can be found in section xxx.

Setpoint Limitation

Setpoints are generally limited before they are used. Each setpoint value can be limited by fix limitation parameters. Dynamic limitation is available in some operation modes.

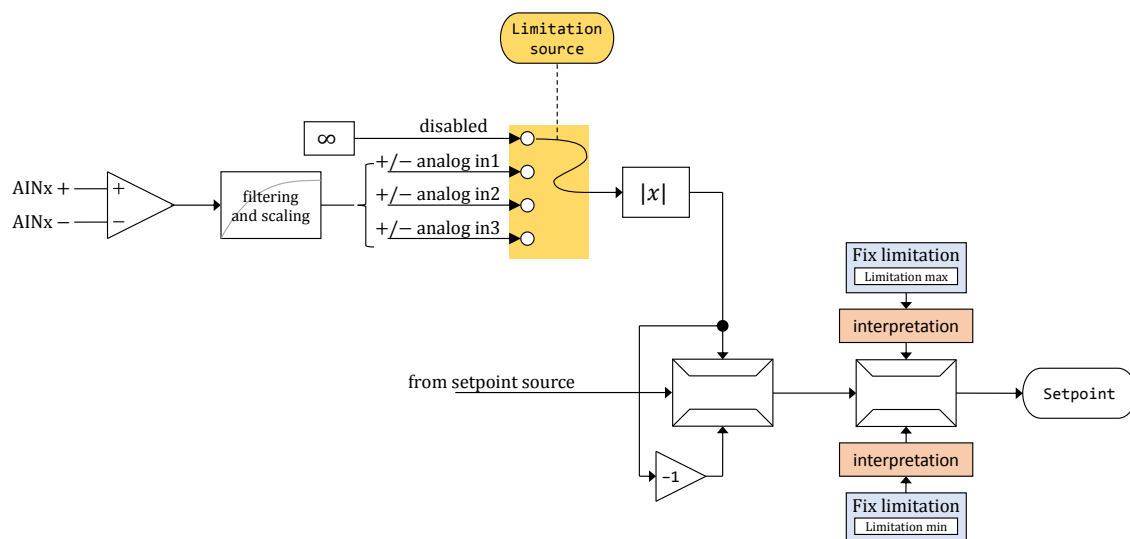


Figure 1: fix and dynamic setpoint limitations

Fix setpoint limitation

All setpoints have fix limits stored in parameters to allow operation within a predefined range.

For example: Speed setpoints might be limited to 0 rpm and 1500 rpm while torque setpoints might be limited to values between -5 Nm and 5 Nm.

Note that in a cascaded control structure like speed- / position- and PID-control, all associated values (torque, speed etc.) are limited simultaneously by the associated parameters.

Similar to *Setpoint type*, the **Limitation type** defines independently the **interpretation** of the limitation values. As an example: The Setpoint values for torque can be interpreted in 'Nm' whereas the limitation values can be in '%' of the rated motor torque.

Dynamic setpoint limitation

It is also possible to use dynamic limitation in addition to the fix limits in some operation modes. The scaling and filtering is described in the analogue input processing section xxx.

Direction reversal

Depending on the setpoint source, it is possible to use the rotational direction reversal or not. The description of the direction reversal signal is to find in section xxx

Setpoint selection and processing parameter overview

Operation mode Parameter	PID control	Torque-control	Speed-/ V/Hz-control	Jog-mode	Position control	
Setpoint source	PI01 PID SP src.	DP01 trq.SP src.	FR01 spd SP src.	FR20 jog SP src.	LM01 pos.SP src.	VS01
Setpoint type	Unitless scaling in %	DP02 trq.SP typ. (percentage, fractional INT32, torque (Nm))	FR02 spd SP typ. (percentage, Fractional INT32, frequency (Hz), speed (rpm))	Absolute in Hz or relative to ext. reference (depending on source)	LM02 pos.scaling (incremental, fractional INT32, custom scale (stps))	Absol relativ referen (depe
External reference	-	DP05 ext.trq.ref	FR05 ext.spd.ref	FR05 ext.spd.ref	-	VS02
Fix setpoint limitation	PI06 PID SP max. PI07 PID SP min.	DP06 max. torque DP07 min. torque	FR06 max. speed FR07 min. speed	FR06 max. speed FR07 min. speed	In stps: LM11 max.pos. LM12 min.pos. else: LM13 max.pos. LM14 min.pos.	VS03
Dynamic setpoint limitation source	-	DP03 trq.lim.src	FR03 spd.lim.src	FR03 spd.lim.src	-	
Dynamic setpoint limitation type	-	DP04 trq.lim.typ	FR04 spd.lim.typ	FR04 spd.lim.typ	-	
Fix setpoints	PI11 PID setp. 1 PI12 PID setp. 2 PI13 PID setp. 3 PI14 PID setp. 4 PI15 PID setp. 5 PI16 PID setp. 6 PI17 PID setp. 7 PI18 PID setp. 8	MF01 torq.setp.1 MF02 torq.setp.2 MF03 torq.setp.3 MF04 torq.setp.4 MF05 torq.setp.5 MF06 torq.setp.6 MF07 torq.setp.7 MF08 torq.setp.8	NF01 speed SP 1 NF02 speed SP 2 NF03 speed SP 3 NF04 speed SP 4 NF05 speed SP 5 NF06 speed SP 6 NF07 speed SP 7 NF08 speed SP 8	FR21 jogSpeedFWD FR22 jogSpeedREV	In stps: LU01 pos. setp.1 LU02 pos. setp.2 LU03 pos. setp.3 LU04 pos. setp.4 LU05 pos. setp.5 LU06 pos. setp.6 LU07 pos. setp.7 LU08 pos. setp.8 else: LF01 pos. setp.1 LF02 pos. setp.2 LF03 pos. setp.3 LF04 pos. setp.4 LF05 pos. setp.5 LF06 pos. setp.6 LF07 pos. setp.7 LF08 pos. setp.8	
Bus sepoints	S005 PID setpnt	S002 torque setp	S001 speed/freq	-	In stps: S003 position SP else: S004 position SP	S007

Conditions for enable, jog mode and latched FWD/REV.

En input	Forward input	Reverse input	Run input	Behavior in case of no Jog and no latched FWD/REV active		Jog mode or latched FWD/REV allowed
				Open loop applications	Closed loop applications	
0				<i>Not powered</i>	<i>Not powered</i>	No
1	0	0	-/1	<i>Not powered</i>	<i>In operation – stop mode</i>	yes
1	1	1				
1	0	1	-/1	<i>In operation</i>	<i>In operation</i>	no
1	1	0				
1	0/1	-				
1	-	0/1				
1	-	-				
1			0	<i>Not powered</i>	<i>Not powered</i>	yes

Explanation:

Doesn't matter if terminal function is configured or not or which state it has

Terminal function is not configured (-)

Terminal function is configured and state is active (1)

Terminal function is configured and state is inactive (0)

Things to note:

-Forward/Reverse and Run have a higher status as the latched FWD/REV version and jog mode.

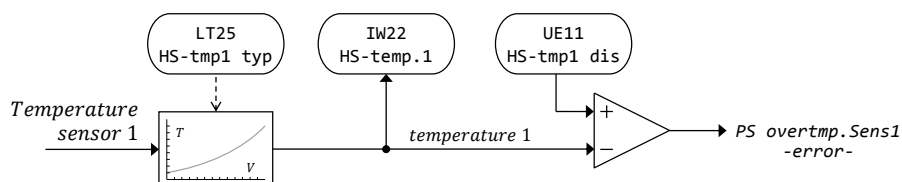
-A transition from *In operation* to *jog/latched... active* is **not** possible because there is already an active setpoint.

-A transition from *Not powered* to *jog/latched... active* may take longer as from *In operation – stop mode* because the motor has to be magnetized. Same from *Not powered* to *In operation*

-In position control FWD/REV starts the positioning. Is that sensible? Or jump into speed control? Latch reset after position reached?

-In all other operation modes latched FWD/REV will run the motor with the actual speed setpoint in speed control (or V/Hz if the is the standard operation mode).

Power stack temperature evaluation, monitoring and fan control



Differences between chassis A/B/C and chassis D

The available **maximum number of temperature sensors** is limited by the power stack type:

Chassis A, B and C: 1 temperature sensor is evaluated (temperature sensor 1)

Chassis D: Up to 8 temperature sensors can be evaluated (temperature sensors 1...8)

Signal processing

In chassis D, the analogue input voltage is automatically multiplied by factor 1.0051. This is for error compensation purpose because the analogue input resistance of ASD is 45kΩ and the CHD series resistances within the signal chain is around 230Ω.

Different Type of temperature sensors:

All temperature sensor inputs can be configured independently to use different types of sensors.

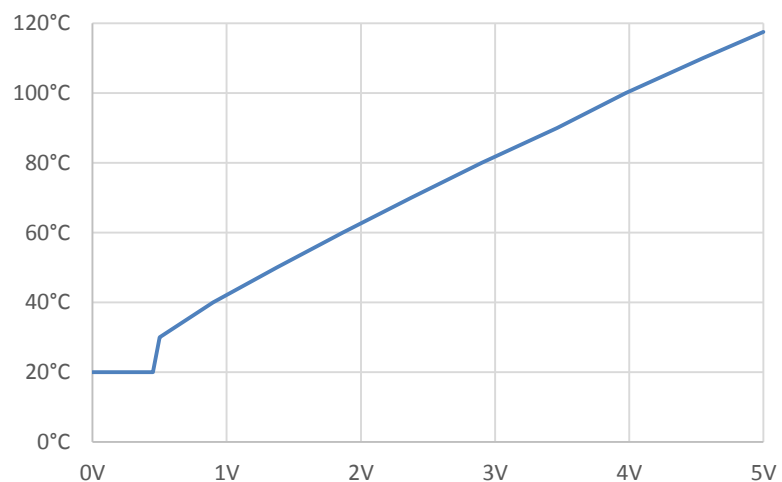
HM08 power stack	
KL03 PS tmp.sens	
LT25 HS-tmp1 typ	Type of power stack temperature sensor 1 sensor inactive: temperature sensor is not used LM335: temperature sensor type LM335 is used SEMIKRON SKiiP3: temperature sensor from SKiiP 3 modules is used

	Linear type sens: Custom linear type temperature sensor
LT26 HS-tmp2 typ	Type of power stack temperature sensor 2 (CHD only) ...
LT27 HS-tmp3 typ	Type of power stack temperature sensor 3 (CHD only) ...
LT28 HS-tmp4 typ	Type of power stack temperature sensor 4 (CHD only) ...
LT29 HS-tmp5 typ	Type of power stack temperature sensor 5 (CHD only) ...
LT30 HS-tmp6 typ	Type of power stack temperature sensor 6 (CHD only) ...
LT31 HS-tmp7 typ	Type of power stack temperature sensor 7 (CHD only) ...
LT32 HS-tmp8 typ	Type of power stack temperature sensor 8 (CHD only) ...

SKiiP3 temperature sensors

A fix characteristic is used to evaluate temperature sensors from type **SEMIKRON SKiiP3**

Sensor output voltage / V	Temperature / °C
0	20
0,45	20
0,5	30
0,9	40
1,375	50
1,865	60
2,375	70
2,9	80
3,462	90
3,975	100
4,55	110
5	117,5

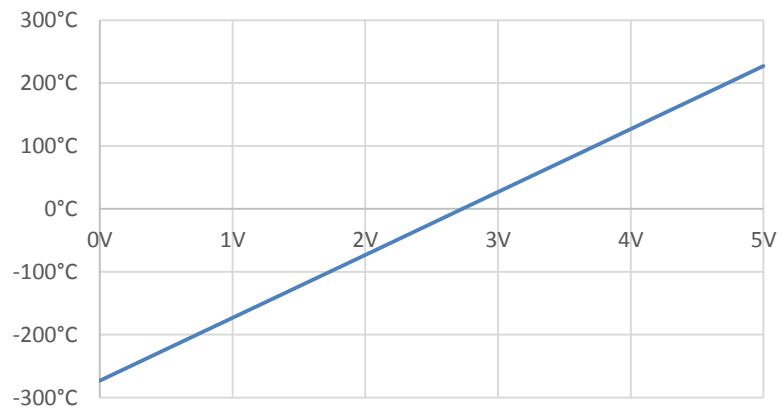


LM335 temperature sensors

A fix characteristic is used to evaluate temperature sensors from type **LM335**

[Note] Characteristic is defined by table values in the source code.

Sensor output voltage / V	Temperature / °C
0	-273,15
1,2315	-150
1,7315	-100
2,2315	-50
2,7315	0
3,2315	50
3,7315	100
4,2315	150
4,7315	200
5	226,85



Custom linear Type temperature sensors

Other temperature sensors than the ones described above can be evaluated in case there characteristic is linear.

Also Non-linear temperature sensors can be used if an approximation is done around the trip point. In this situation, other temperature values might not be measured accurately.

The configuration of custom temperature sensors can be done by setting a temperature value for 0V as well as for 5V sensor output. Values between both points are interpolated.

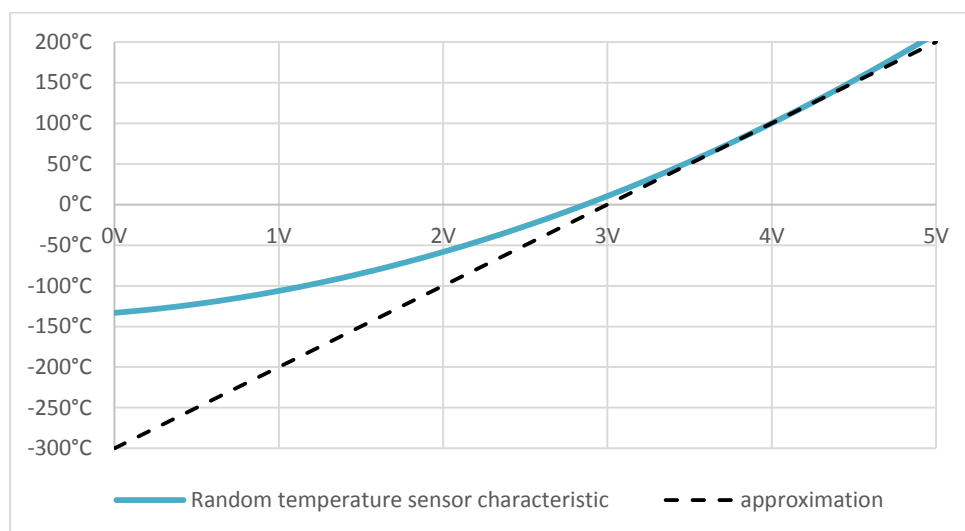
HM04 characteris		
KL03 PS tmp.sens		
PT01 temperat@0V	0,0°C	temperature value at 5V temperature sensor output voltage (control board input voltage)
PT02 temperat@5V	250,0°C	temperature value at 5V temperature sensor output voltage (control board input voltage)

Temperature sensors with series resistance

For temperature sensors with internal series resistance, an analogue input resistance of 45k Ω on chassis A/B/C and 45.23k Ω in chassis D has to be considered.

Configuration example for a random temperature sensor

The diagram below shows a random temperature characteristic that was approximated around the point 4V/100°C. The correct configuration in this example would be: PT01 = -300°C; PT02 = 200°C



Over-temperature detection

Over-temperature switch off

Each temperature sensor (1 to 8) has its own temperature threshold at which an error switch-off is triggered:

Sensor	Error threshold (HM07 monitoring->...)	Error message
1	UE11 HS-tmp1 dis	<i>PS overtmp.Sens1</i>
2	UE12 HS-tmp2 dis	<i>PS overtmp.Sens2</i>
3	UE13 HS-tmp3 dis	<i>PS overtmp.Sens3</i>
4	UE14 HS-tmp4 dis	<i>PS overtmp.Sens4</i>
5	UE15 HS-tmp5 dis	<i>PS overtmp.Sens5</i>
6	UE16 HS-tmp6 dis	<i>PS overtmp.Sens6</i>

7	UE17 HS-tmp7 dis	<i>PS overtmp.Sens7</i>
8	UE18 HS-tmp8 dis	<i>PS overtmp.Sens8</i>

Figure 2 illustrates the over-temperature switch-off function for temperature sensor 1

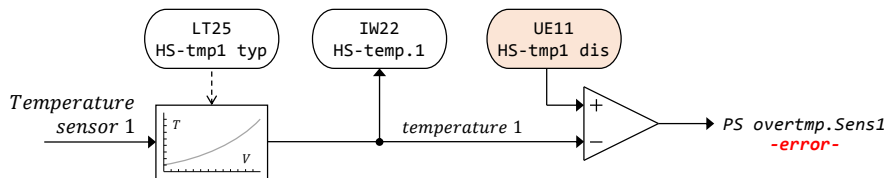


Figure 2: Over-temperature switch-off diagram temperature sensor 1

Over-temperature warning

All evaluated temperature sensor values (1 to 8) are combined to generate the warning message **PS overtmp.Sens1** if any of the temperature values is higher than the warning threshold **UE19 HS-tempWarn**.

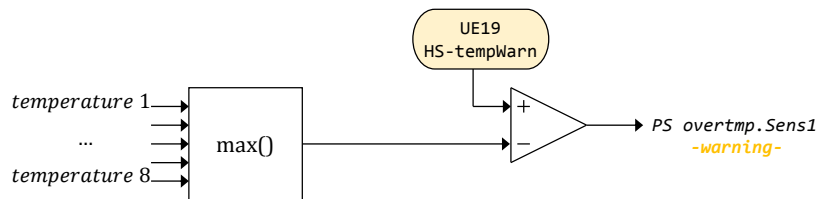


Figure 3: Over-temperature warning diagram

Fan control

All evaluated temperature sensor values (1 to 8) are combined to control the internal cooling fan in the power stack. The highest temperature value is used to switch-on the fan at the threshold **SL08 fan on temp** and to switch it off at the threshold **SL09 fan off tmp**.

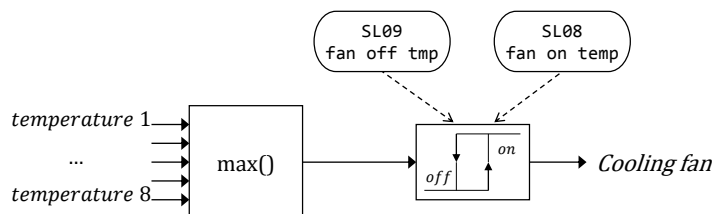


Figure 4: Cooling fan control

Motor Lsq / Lsd measurement

Procedure

Isolate the motor terminals from all other circuits.

Select two motor terminals for this test. The third terminal is not used and should be left open-circuit.

Connect a convenient DC source (for example a laboratory adjustable DC power supply) to the motor terminals and increase the current to align the PMSM rotor at a stable point. There will be one stable point for each pair of motor poles. For example, a 4-pole motor will have two stable points. These shaft angles correspond to the d-direction. Note the angles of the stable points.

Disconnect the current source and measure the inductance at one of the stable points. This value is Lsd.

Rotate the shaft (direction not important) to a position that is 25% of the distance to the next stable point. This is the q-direction. Measure the inductance at this point to obtain Lsq.

Note: The two inductances, Lsd and Lsq will correspond to the highest and lowest inductance measured as the shaft is rotated slowly. Normally Lsq is the higher value. For some motor constructions, Lsd will be the higher value.

Connect a suitable inductance meter between two motor phase terminals. The third phase terminal is unconnected.

Rotate the motor shaft slowly to find the minimum and maximum inductance values.

The associated parameter values should be set as follows:

“MP13 Lsq UV (op)” = Lsq

“MP11 Lsd/Lsq” = Lsd / Lsq

“SR03 prop.g.corr” should be left at 100 % (no Ls-identification necessary)

Alternative measurement

It is also possible to conduct a test with two motor phase joined together. The procedure is the same as above, however, with this connection the measured Lsq need to be converted as follows:

“MP13 Lsq UV (op)” = Lsq * 4/3

ASD - Modbus documentation for software version v10r020b

1 ASD – Modbus

1.1 Hardware (termination resistor) setup

DIP-switch	State	Description
SW2	0	Modbus termination resistor of 120 Ω is NOT used. Recommended if this ASD system is NOT the last point of a Modbus cable run
	1	Modbus termination resistor of 120 Ω is used. Recommended if this ASD system is the last point of a Modbus cable run

1.2 Modbus functions

The ASD Modbus functionality was adapted from MSC-3. Therefore, ASD supports the same function codes. Existing differences are described in this document.

Detailed information about the Modbus standard can also be downloaded from modbus.org

In the actual software version, all accessible values are stored in holding registers.

Input registers, discrete inputs or coils are supported by software but currently not used.

can be done with the following commands:

Function	Function code	description
Read Holding Registers	03	Read operation of a continuous block of data
Write Single Register	06	Write operation to one single register. This works only for 16bit parameters. Longer

		parameters need the command 'Write Multiple Registers'
Write Multiple Registers	16	Write operation to a continuous block of data

Parameter modifications that are done via MODBUS are only temporary, that means that the values that have previously been stored in the EEPROM will be loaded again after a restart. If the modifications have to be non-volatile then set 'SO30 EPROM save' to 1 and **all** temporary parameter values will stored in the EEPROM **once**.

1.3 Modbus addresses

1.3.1 Address range 0 to 999

The Modbus address range from 0 to 999 is not directly assigned to parameters in the ASD menu structure (parameter list). All these values are volatile and will be reset to their default values after a restart. There is also no way of storing them in a volatile memory section. Not all of this data range is actually used. The Modbus error ILLEGAL DATA ADDRESS is generated if a not available address was received.

Memory map

Holding Registers (read and write)

Address	Register ID	Initial value	description
0	Write violat.cnt	0	Total write violation counter for events of Limitations¹¹ + incomplete write requests¹² + write denials¹¹
1	Limitation count	0	Total Limitations¹¹ counter (number of parameters that have been limited after they have been received)
2	Addr.of Limitat.	0xFFFF	Modbus address of last Limitation during a write command
3	Limitat.cnt.temp	0	Number of Limitations during the last transmission where a limitation occurred
4	IncmlptWrt count	0	Total incomplete write request¹² counter (number of (32 bit) parameters that have been rejected because they have not been received completely)
5	Addr.IncmlptWrt	0xFFFF	Modbus address of parameter during the last

¹¹ More information in chapter 1.4: Write restrictions and value limitations (Address range 1000 to 9999 only)

¹² More information in chapter 1.5.2: 32-bit write-access

			incomplete write request
6	IncmlptWrtCntTmp	0	Number of incomplete write requests during the last transmission where incomplete write requests occurred
7	wDenial count	0	Total write denial ¹¹ counter (inaccessible Modbus addresses)
8	Addr.of wDenial	0xFFFF	Modbus address of to last write denial
9	wDenial.cnt.temp	0	Number of write denials during the last transmission where write denials occurred

Table 1: Holding Registers (read and write) within address range 0 to 999

1.3.2 Address range 1000 to 9999

The Modbus address range from 1000 to 9999 is used to access to the ASD menu structure for parameters, setpoints and actual values. This address range is like a continuous block of accessible data. Therefore there will be no Modbus error of ILLEGAL DATA ADDRESS generated - neither during a read nor during a write access within this section.

Memory map

Each parameter/ setpoint/ actual value in the ASD menu structure has its own Modbus-address somewhere in this memory section. Information to the associated **addresses for each parameter can be found in the parameter list.**

1.4 Write restrictions and value limitations (Address range 1000 to 9999 only)

Values that are written to addresses in the range of 1000 to 9999 that are not used in the ASD menu or that are write-protected are ignored. It is recommended not to write to unused addresses to avoid malfunctions if these addresses are going to be used in other ASD software versions.

The same problematic also applies to values that are internally limited after they have been received.

ASD provides various error registers to detect such invalid write requests (**Limitations** and **write denials**). See therefore holding registers 0 to 9 (Table 1).

1.5 Parameters with 32-bit length (Address range 1000 to 9999 only)

32-bit parameters are spread over two (16 bit) Modbus addresses. Therefore only the multiple read/write functions Read-Holding-Registers and Write-Multiple-Registers can provide full access to them.

The address order how the 32-bit values are organized is Little-Endian that means, the least significant 16-bit are stored at the lowest address whereas the highest significant 16-bit are stored at the highest address.

Modbus address order example of a 32-bit parameter:

Parameter value @ Modbus address 1150: *0x87654321*

Value in Modbus register 1150: *0x4321*

Value in Modbus register 1151: *0x8765*

1.5.1 32-bit read-access

During a continuous read procedure with the command Read-Holding-Registers (function code 3), the associated parameter value is latched to make sure that there are no problems with data consistency.

If necessary, 32-bit values can also be read-in piece by pieces. Note that the associated data is not latched between two separate read commands.

Continuous read example

Parameter data from address 1150 to 1155:

Parameter	A		B		C	
Modbus address	1150		1152		1154	
Parameter value	0x22221111		0x44443333		0x66665555	
Modbus register	1150	1151	1152	1153	1154	1155
Value in Modbus register	0x1111	0x2222	0x3333	0x4444	0x5555	0x6666

Table 2: Continuous read example

A continuous read command of 4 Modbus addresses starting at address 1151 will return the data 0x2222, 0x3333, 0x4444, 0x5555

1.5.2 32-bit write-access

Only parameter values that have been transmitted completely within one message will be processed.

Parameter values are latched until they are received completely and then modified at once to make sure that there are no problems with data consistency.

Note that the associated data is not latched between two separate write commands.

ASD provides various error register to detect such **incomplete write** requests. See therefore holding registers 0, 4, 5 and 6 (Table 1).

Continuous write example

See the data content of the continuous read example Table 2: Continuous read example.

A Write-Multiple-Registers-command to 4 addresses starting at address 1151 with the data 0x7777, 0x8888, 0x9999, 0xAAAA will not result in a modification of parameter A and C because the data for these parameters are not transmitted completely.

Read-back Modbus data from address 1150 to 1155 after the write command:

Parameter	A		B		C	
Modbus address	1150		1152		1154	
Parameter value	0x22221111		0x99998888		0x66665555	
Modbus register	1150	1151	1152	1153	1154	1155
Value in Modbus register	0x1111	0x2222	0x8888	0x9999	0x5555	0x6666

Table 3: Continuous write example

General purpose PID controller

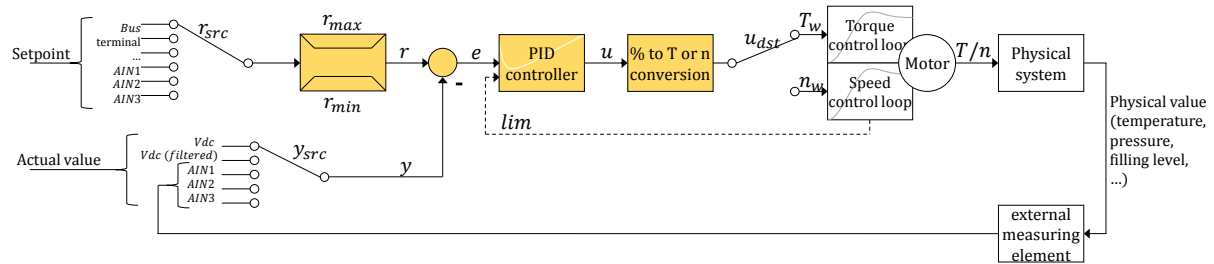


Figure 5: Complete control loop with general purpose PID controller

The built-in general purpose PID controller can be used to control physical values like temperature or pressure. The actuating value for the physical system can either be torque T or speed n .

The scaling of the PID controller inputs and outputs are unitless in % what makes it very flexible to use.

1.1. Parameter overview with example values

1.1.1. Configuration parameters

Main menu: HM02 contr.param			
Sub menu: RP05 field weak.			
PI01	PID SP src.	<i>analog in2</i>	Setpoint source (r_{src}) for PID controller
PI02	PID AV src.	<i>analog in3</i>	Actual value source (y_{src}) for PID controller
PI03	gain Kp	10.000	Normalized proportional gain K_p of PID controller
PI04	int.time Ti	4000.0 ms	Integration time T_i of PID controller
PI05	drv.time Td	10.0 ms	Derivative time T_D of PID controller
PI06	PID SP max.	50.00 %	PID setpoint limitation max (r_{max})
PI07	PID SP min.	-50.00 %	PID setpoint limitation min (r_{min})
PI11	PID setp. 1	0.00%	PID controller setpoint 1
PI12	PID setp. 2	0.00%	PID controller setpoint 2

1.1.2. Bus-Setpoints

Setpoint menu

SO05	PID setpnt	25.00 %	PID controller setpoint if setpoint source (r_{src}) is 'bus'
------	------------	---------	---

1.1.3. Actual values

Actual value menu

IW42	PID act.val	25.00%	Actual value (y) in the PID control loop
IW43	PID setpnt	25.00%	Setpoint value (r) in the PID control loop

Control parameters and structure

The PID controller can be tuned by modification of three control parameters:

Proportional Gain K_P

Integration time T_I

Derivative time T_D

The complete transition functions of the PID controller is:

$$F_R(s) = K_P \cdot \frac{1 + \frac{1}{T_I s} + T_D s}{1 + T_F s}$$

The additional 1st-order filter element with the time constant T_F is used to limit the output signal in case of a step signal on the controller input.

The filter time T_F has a fix relation to the derivative time: $T_F = 0.1 \cdot T_D$

Control structure

The control structure can be modified from PID to PI, PD or P control structure by setting the integration and/or the derivative time to zero.

The resulting control structures are illustrated in *Figure 6*

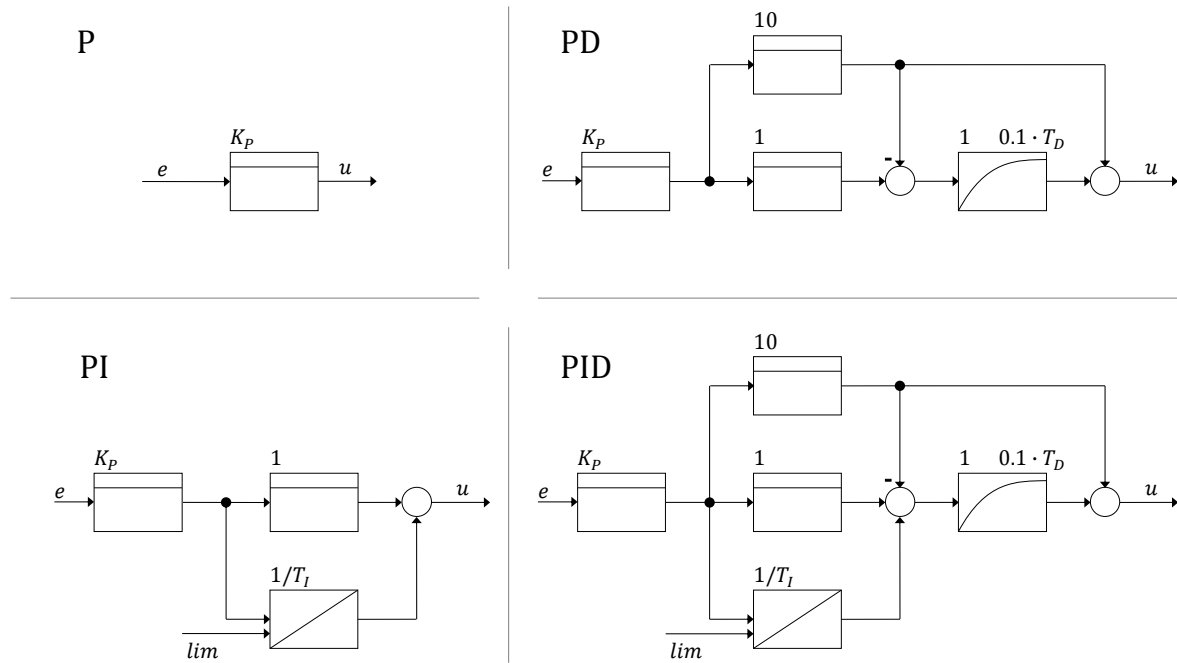


Figure 6: various control structures of the general purpose PID controller

1.2. PID controller inputs and outputs

The general-purpose-PID-controller internally uses a unitless scaling in % on both inputs and outputs.

Depending on the selections 1.2.1 and 1.2.2.2, the physical input and output scaling must be considered in the closed control loop. For example:

$$IN_{scale} = \frac{100\%}{808\text{ V}}; \quad OUT_{scale} = \frac{30\text{ Nm}}{100\%}; \quad PID_{scale} = IN_{scale} \cdot OUT_{scale} = \frac{30\text{ Nm}}{808\text{ V}}$$

1.2.1. Controller output

The actuation element inside the control loop of **Error! Reference source not found.** is either the motor torque or the motor speed. For this purpose, the controller output value in % will be converted into torque setpoint T_w or a speed setpoint n_w depending on the control mode **BF01 contrl mode**. To connect the PID controller output to the torque or speed controller, the respective setpoint source must be set to *PID controller (DP01 trq.SP src. or FR01 spd SP src.)*.

Torque scaling with controller output destination u_{dst} = torque setpoint T_w

If the control mode setting is *torque control*, 100% on the controller output will represent the external torque reference value:

Main menu: HM06 setp.proces			
Sub menu: SW01 torq. proc			
DP05	ext.trq.ref	100.00Nm	Torque setpoint in Nm per 100% PID output

Speed scaling with controller output destination $u_{dst} = \text{speed setpoint } n_w$

If the control mode setting is *speed control* any other control mode with a speed setpoint, 100% will represent the external speed reference value:

Main menu: HM06 setp.proces			
Sub menu: SW03 speed proc.			
FR05	ext.spd.ref	5000rpm	Speed setpoint in rpm per 100% PID output

Controller output limitations

The controller output is internally limited to +/- 100%. Further limitations can be achieved by limitation of the physical representation like the motor torque or speed directly.

A detection of a torque or speed limitation automatically prevents the PID controller from winding up.

1.2.2. Controller Inputs

1.2.2.1. Setpoint source (r_{src})

The Parameter **PI01 PID SP src.** defines the setpoint source (r_{src}). The following settings are possible:

PI01 PID SP src. value	description
<i>disabled</i>	PID controller is disabled
<i>terminal</i>	Setpoint selection out of 8 predefined setpoints PI11 PID setp. 1 to PI18 PID setp. 8 via terminals
<i>terminals binary</i>	Setpoint selection out of 8 predefined setpoints

	PI11 PID setp. 1 to PI18 PID setp. 8 via terminals, binary coded
<i>bus</i>	Setpoint source is SO05 PID setpnt
<i>fix value</i>	Setpoint source is PI11 PID setp. 1
<i>analog in1</i>	Setpoint source is analog input 1. Scaling and filtering according to the analog input setup. (100 % $\hat{=}$ full-scale)
<i>analog in2</i>	Setpoint source is analog input 2. Scaling and filtering according to the analog input setup. (100 % $\hat{=}$ full-scale)
<i>analog in3</i>	Setpoint source is analog input 3. Scaling and filtering according to the analog input setup. (100 % $\hat{=}$ full-scale)

1.2.2.2. Actual value source (y_{src})

The Parameter **PI02 PID AV src.** defines the actual value source. The following settings are possible:

PI02 PID AV src. value	description
<i>analog in1</i>	Actual value source is analog input 1. Scaling and filtering according to the analog input setup. (100 % $\hat{=}$ full-scale)
<i>analog in2</i>	Actual value source is analog input 2. Scaling and filtering according to the analog input setup. (100 % $\hat{=}$ full-scale)
<i>analog in3</i>	Actual value source is analog input 3. Scaling and filtering according to the analog input setup. (100 % $\hat{=}$ full-scale)
<i>Vdc</i>	Internally measured, non-filtered DC-link voltage. Scaling: 100% $\hat{=}$ LT05 Vdc ref.vol
<i>Vdc (filtered)</i>	Same as setting <i>Vdc</i> but filtered by the filter time constant SR08 timecon.Vdc
<i>speed</i>	Measured motor speed value Scaling: 100% $\hat{=}$ RF01 int.frq.ref
<i>speed (filtered)</i>	Same as setting <i>speed</i> but filtered by the filter time constant DR04 speed fil.

1.3. PI-controller gain selection by symmetrical optimum

If it is not sure what controller gains need to be used, a simple rule can be applied called symmetrical optimum:

PI-controller gain selection by symmetrical optimum	
$K_P = \frac{\tau_{dom}}{2 \cdot K_S \cdot \tau_\sigma}$ $T_I = 4 \cdot \tau_\sigma$	τ_σ : Sum of small time constants τ_{dom} : Dominating time constant K_S : System gain
Examples	
(A) Second- or higher-order behaviour of control system like $\frac{K}{(1 + \tau_1 s)(1 + \tau_2 s)(1 + \tau_3 s)}$ $\tau_1 \gg \tau_2; \tau_1 \gg \tau_3$ → $\tau_\sigma = \tau_2 + \tau_3$ $\tau_{dom} = \tau_1$ $K_S = K$ → $K_P = \frac{\tau_1}{2 \cdot K \cdot (\tau_2 + \tau_3)}$ $T_I = 4 \cdot (\tau_2 + \tau_3)$	(B) Integrative plus first- or higher-order behaviour of control system like $\frac{K}{(1 + \tau_1 s)(1 + \tau_2 s)s}$ → $\tau_\sigma = \tau_1 + \tau_2$ $\frac{K_S}{\tau_{dom}} = K$ → $K_P = \frac{1}{2 \cdot K \cdot (\tau_1 + \tau_2)}$ $T_I = 4 \cdot (\tau_1 + \tau_2)$

The formulas above does not directly show the PID controller normalization. As also described in 1.2, it is important to consider this when calculation the controller P-Gain (K_P'). The easiest way is to add this values to the physical system gain. For example:

$$K' = PID_{scale} \cdot K$$

1.4. Configuration example

1.4.1. Temperature control application: P-controlled fan

Requirements

Control variable: temperature τ in degree

Physical temperature sensor output scaling:

$$V_{ensor} = (\tau - 20^{\circ}C) \cdot 0.05 \text{ /}^{\circ}C$$

Analog input voltage range: 0V to 10V.

Expected control behaviour (motor speed setpoint in relation to the temperature):

$$n_{motor} = \begin{cases} 0 \text{ rpm}, & T < 50^{\circ}C \\ (\tau - 50^{\circ}C) \cdot 10 \text{ rpm/}^{\circ}C, & 100^{\circ}C > T \geq 50^{\circ}C \\ 500 \text{ rpm}, & 100^{\circ}C \geq T \end{cases}$$

Actual value source (y_{src}) should be analog input 2.

Setpoint source (r_{src}) should be a fix value.

ASD Configuration

Set the output scaling n_{scale} to allow a sensible value range for example 1000rpm/100%.

The analog input scaling $V_{scale,AIN2}$ is typically 10V/100% (default value).

Calculation of the temperature scaling τ_{scale} :

$$\tau_{scale} = \frac{^{\circ}C}{0.05} \cdot \quad \quad \quad le,AIN2 = \frac{^{\circ}C}{0.05} \cdot \frac{10V}{100\%} = 200^{\circ}C/100\%$$

Calculation of the controller gain K_P :

$$K_P = - \frac{dn_{w,motor}}{d\tau} \cdot \frac{\tau_{scale}}{n_{scale}} = \quad 1) \cdot 10 \frac{rpm}{^{\circ}C} \cdot \frac{200^{\circ}C/100\%}{1000rpm \ 100\%} =$$

Calculation of the controller setpoint value r in % (temperature sensor value @ 50°C):

$$= \frac{V_{ensor@50^{\circ}C}}{cale,AIN2} = \frac{4.5}{10V \ 100\%} = 45\%$$

Main menu: HM02 contr.param

Sub menu: RP09 PID control

PI01	PID SP src.	<i>fix value</i>	Setpoint source (r_{src}) is PI11 PID setp. 1
PI02	PID AV src.	<i>analog in2</i>	Actual value source (y_{src}) is analog input 2
PI03	gain Kp	-2.000	Normalized Proportional Gain K_p' of PID controller
PI04	int.time Ti	0.0ms	Integration part of the PID controller is off
PI05	drv.time Td	0.0ms	Derivative part of the PID controller is off
PI06	PID SP max.	50.00%	PID setpoint limitation max (r_{max}) is 50% (500rpm)
PI07	PID SP min.	0.00%	PID setpoint limitation min (r_{min}) is 0% (0rpm)
PI11	PID setp. 1	45.00%	PID controller setpoint 1 is 45% (50°C)

Main menu: HM03 acquisition

Sub menu: AQ05 analog inpt

AI22	scale ana2	10.00V	Analog input voltage scaling $V_{scale,AIN2}$ is 10V/100%
------	------------	--------	---

Main menu: HM06 setp.proces

Sub menu: SW03 speed proc.

FR05	ext.spd.ref	1000rpm	Speed scaling n_{scale} . 1000rpm at a PID output of 100%
------	-------------	---------	--

1.4.2. Flywheel application

The general purpose PID controller can be used to control the DC-Link voltage by using the DC-link voltage as an actual value source (See 1.2.2.2).

In this application, the PID controller output needs be used to generate a torque setpoint (1.2.1).

The setpoint of the DC-Link voltage can be configured according to 1.2.2.1. If the setpoint-source is **bus** then care must be taken after a reset because the reset of all bus vale will be zero. The motor might be powered with full torque and the DC-link voltage might break down. It is recommended to use a fix setpoint instead.

How to set sensible setpoints

To allow a power flow into the flywheel, a DC-link setpoint below the actual dc-link value must be used. If the DC-link voltage drops below this setpoint, the controller draws power out of the flywheel and into the DC-link in order to maintain the setpoint voltage.

The scaling is in % of **LT05 Vdc ref.vol** must be considered to calculate the correct setpoint y' in %.
For example:

$$y' = y \cdot y_{scale} = 550 \text{ V} \cdot \frac{100 \%}{808 \text{ V}} \approx 68.07 \%$$

How to set the controller gains – background information

Theoretical background: Mathematical description of the controlled system	
$p_M = T \cdot \omega$ $\omega = \frac{1}{I} \int T \cdot dt$ $\dot{V}_{dc} = -\frac{p_G + p_M}{C_{dc} \cdot V_{dc}} = -\frac{p_G + T\omega}{C_{dc} \cdot V_{dc}}$	p_G: Power flow into the grid by a line side converter p_M: Power flow into the flywheel C_{dc}: DC-link capacity V_{dc}: DC-link voltage T: Torque of the motor that is driving the flywheel ω: Angular frequency of the flywheel I: Inertia of the system

As it can seen from the equations above, the system-gain is not constant, the power flow into the grid by a line side converter (p_G) might be even unknown. If ω and V_{dc} are seen as slow changing value and p_G as an independent disturbance signal, the system gain can be approximated by:

$$\frac{K_s}{\tau_{dom}} = K \approx - \frac{\omega_{max}}{C_{dc} \cdot V_{dc,min}}$$

The higher the control loop gain, the higher the risk of instability, therefore the system must be capable of running stable at **maximum** rotational speed at **minimum** DC-link voltage.

To calculate the maximum controller gain, the signal delays in the closed loop system must be known:

Current controller delay $\tau_{i,q}$ (**SR02 setl.time q** divided by 4) and V_{dc} filtering $\tau_{Vdc,fil}$ (**SR08 timecon.Vdc**).

The controller gains according to the symmetrical optimum (1.3, example B) are:

$$\left(K' = K \cdot PID_{scale} = -\frac{\omega_{max}}{C_{dc} \cdot V_{dc,min}} \cdot PID_{scale} \right)$$

PI-controller gain equations	
Physical proportional gain	$K_P = \frac{C_{dc} \cdot V_{dc,min}}{2 \cdot \omega_{max} \cdot (\tau_{i,q} + \tau_{Vdc,fil})}$
Normalized proportional gain	$K'_P = \frac{C_{dc} \cdot V_{dc,min}}{2 \cdot \omega_{max} \cdot (\tau_{i,q} + \tau_{Vdc,fil}) \cdot PID_{scale}}$
Integration time	$T_I = 4 \cdot \tau_{\sigma} = 4 \cdot (\tau_{i,q} + \tau_{Vdc,fil})$

Example application

Specifications	
DC-link capacity	$C_{dc} = 0.66mF$
Minimum DC-link voltage during operation	$V_{dc,min} = 400V$
Various drive settings	
FR06 max. speed = 10000 rpm FR07 min. speed = 0 rpm	$\omega_{max} = 2\pi \cdot 10000 \text{ rpm} \cdot 1 \text{ Hz}/60 \text{ rpm}$
DP05 ext.trq.ref = 30 Nm LT05 Vdc ref.vol = 808 V	$PID_{scale} = \frac{30 \text{ Nm}}{808 \text{ V}}$
SR08 timecon.Vdc = 5ms SR02 setl.time q = 2ms (4 times $\tau_{i,q}$)	$\tau_{Vdc,fil} = 5 \text{ ms}$ $\tau_{i,q} = 0.5 \text{ ms}$
Results for the controller gain	
$K_P = -0.023 \frac{Nm}{V}; \quad K'_P = -0.617$	PI03 gain Kp = -0.617
$T_I = 22 \text{ ms}$	PI04 int.time Ti = 22 ms

These values for K'_P and T_I are only calculated by using approximations but can be seen as maximum sensible values (or minimum in case of T_I).

If a faster controller response is necessary, this can simply be achieved by reducing the DC-link filter time or by using the unfiltered DC-link voltage (**Vdc**) as an actual value source (**PI02** **PID AV src.**). According to the PI-controller gain equations it will then be possible to increase K_p' and to reduce T_I .

Depending on the application, even a P-controller can be sufficient. If this is the case, T_I can simply be set to zero.

Example configuration of all associated parameters:

Main menu: HM02 contr.param			
Sub menu: RP01 curr. ctrl.			
SR02	setl.time q	2.0 ms	Settling time q
SR08	timecon.Vdc	5.0 ms	Time constant of Vdc measuring
Sub menu: RP09 PID control			
PI01	PID SP src.	<i>fix value</i>	Setpoint source (r_{src}) is PI11 PID setp. 1
PI02	PID AV src.	<i>Vdc (filtered)</i>	Actual value source (y_{src}) is analog input 2
PI03	gain Kp	- 0.617	Proportional Gain K_p' of PID controller
PI04	int.time Ti	22.0 ms	Integration time T_I of the PID controller
PI05	drv.time Td	0.0 ms	Derivative part of the PID controller is off
PI06	PID SP max.	74.26 %	PID setpoint limitation max (r_{max}) is 74.26 % (600 V)
PI07	PID SP min.	61.88 %	PID setpoint limitation min (r_{min}) is 61.88 % (500 V)
PI11	PID setp. 1	68.07 %	PID controller setpoint 1 is 68.07 % (550 V)
Main menu: HM06 setp.proces			
Sub menu: AS01 oper. manag			
BF01	ctrl mode	torque control	Basic control mode = torque control
Main menu: HM06 setp.proces			
Sub menu: SW01 torq. proc.			
DP01	trq.SP src.	PID controller	Torque setpoint comes from the PID controller

DP04	trq.lim.typ	percentage	Motor torque limitation is in % of the rated torque
DP05	ext.trq.ref	30.00 Nm	PID output scaling / torque scaling: 30 Nm at a PID output of 100%
DP06	max. torque	100.0	Maximum motor torque limitation to 100 % of the rated torque
DP07	min. torque	-100.0	Minimum motor torque limitation to -100 % of the rated torque

Sub menu: SW03 torq. proc.

FR04	spd.lim.typ	percentage	Motor speed limitation is in % of the rated speed
FR06	max. speed	100.0	Maximum motor speed limitation to 100 % of the rated motor speed
FR07	min. speed	0	Motor must not be allowed to turn backwards!

V/Hz operation mode

Standard V/Hz parameters

UF01 boost (in %): Boost voltage in % of the rated voltage

UF02 corner freq (in Hz) : Stator corner frequency between base speed range and field weakening. Can be set to 0 if voltage is allowed to raise up to the maximum output voltage

UF03 gradient (in %): Modification of the V/Hz gradient in %. For example: UF03 = 90% means 90% of the rated voltage @ rated frequency

V/Hz parameters for voltage and current limitation

To avoid overvoltage and overcurrent trips, special parameters are available to activate limitation features. Voltage and Current limitation do work during acceleration and deceleration. There is no protection if the load changes in a steady speed point.

UF04 I-lim.accel: Enable current limitation in V/Hz mode during acceleration.

UF05 I-lim.decel: Enable current limitation in V/Hz mode during deceleration

UF06 VDC lim.: Enable DC-link voltage limitation in V/Hz mode during deceleration.

UF09 t-min break (in s): Period of time from rated speed to zero speed to allow a minimum deceleration in case of limitation during deceleration. This is to make sure, that an invalid limitation will not prevent a motor from decelerate at all. Low values will reduce the ability of current and voltage limitation during deceleration.

The following parameters are to adjust the thresholds for the above mentioned limitation functions:

LT10 max.current/ MP06 max.current (in A): current limitation (lowest value of both parameters). This value should be much lower than the over current trip value *LT11 overcurrent*. If a closer trip point is needed, I2T features should be used.

LT14 trq.lim.Vdc (in V): size of limitation range because of Vdc (DC link voltage)

LT15 t0-thrs.Vmx (in V): upper limitation range because of Vdc (DC link voltage). This value should be reasonable lower than *LT06 Vdcmax dis*.

LT14/ LT15 configuration example:

LT14 = 20V; LT15 = 760V → Vdc limitation starts smoothly at 740V with full impact at 760V

ASD – IM and PMSM field weakening functions

Parameter overview

Main menu: HM02 contr.param			
Sub menu: RP09 PID control			
FS01	f.weak.IM	<i>off</i>	Field weakening on/off for induction motors
FS02	f.weak.PMSM	<i>off</i>	Field weakening procedure selection for synchronous motors
FS03	fw char.low	10.0 %	Field weakening characteristic lowering <i>Field weakening characteristic starts by this factor earlier as the theoretically optimum value.</i>
FS04	isdmax corr	100.0 %	Correction of the field weakening current For weak/strong/constant-FW operation (PMSM only)
FS05	volt-cntrl.	<i>on</i>	Additional output-voltage controller in case of dynamic field weakening (PMSM only)
FS06	VcontrlGain	100.0 %	Additional output-voltage controller gain correction value. (PMSM only)
FS07	Rs-comp.	<i>on</i>	Additional stator resistance compensation algorithm in case of dynamic field weakening (PMSM only)

Field weakening operation for induction motors

The field weakening characteristic for induction motors (**FS01 f.weak.IM = on**) weakens the flux to a value at which the motor can generate its maximum torque (this does not apply to the basic speed range).

The expected available voltage (**HM08->LT04 Input volt.**) as well as the maximum motor current (**HM01->MP06 max.current**) have both direct impact to the magnetizing current in the field weakening area.

Parameter **FS03 fw char.low** can be used to reduce to voltage consumption by starting to reduce the magnetizing current at lower frequencies as normal.

Note that the voltage consumption in the basic speed range can be modified by the magnetizing current correction factor **HM02→RP06→MG01 imag corBSR**.

Field weakening operation for permanent magnet synchronous motors

4 different algorithms are available for field weakening in combination with PMSM.

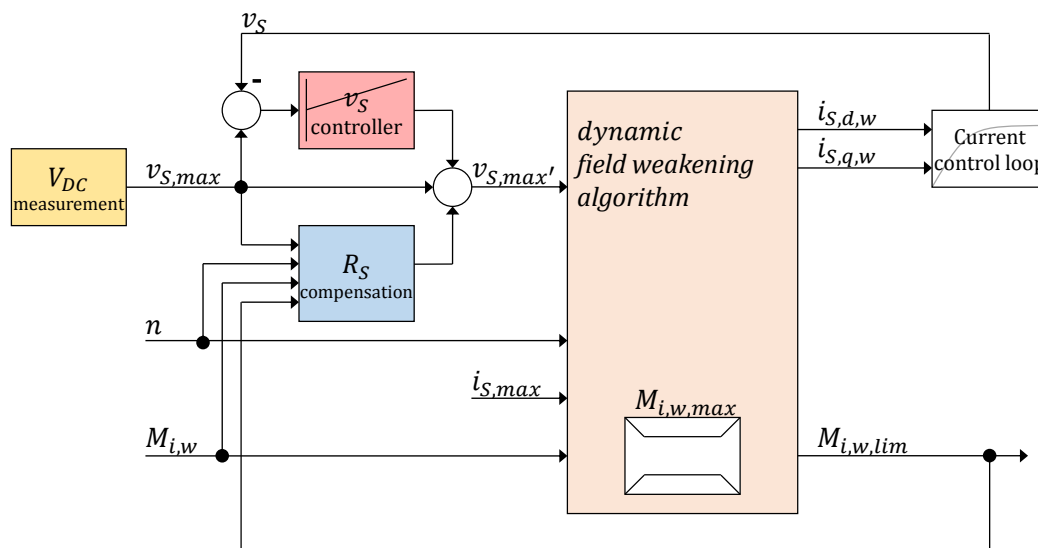
The most sophisticated algorithm is dynamic field-weakening (dynamic-FW).

1.1.1 Dynamic field weakening (FS02 f.weak.PMSM = *dynamic-FW*)

In the basic speed range, field weakening current is used to allow maximum possible torque per ampere.

Above basic speed range, this algorithm produces dynamically the nearly optimal field weakening current for machines with small stator resistances. To do so, the maximum allowed motor current $i_{s,max}$ is considered as well as the maximum output voltage $v_{s,max}$ that is calculated from the measured DC-link voltage.

Parameter **FS03 fw char.low** can be used to reduce to voltage consumption by increasing the field weakening current at lower frequencies as normal. Values other than 0 are only sensible if the voltage controller is switched off and if the drive works in voltage limitation.



Additional (**recommended!**) options are:

Output voltage controller (FS05 volt-cntrl. = on)

Machine data is never 100% accurate. Also machines with high stator resistances might reduce the accuracy of the dynamic field-weakening algorithm. Under this circumstances, the drive dynamic might suffer because of a voltage limitation or the maximum torque might be reduced because not the full available output voltage is used during field weakening.

All this can be compensated by an optional voltage controller. The output voltage is kept close to its maximum value in the field weakening area. The systems stays at a point of maximum torque and efficiency.

Parameter **FS06 VcontrlGain** can be used to modify the speed of the voltage controller in relation to a recommended internal value.

Rs compensation algorithm (FS07 Rs-comp. = on)

This optional algorithm compensates most of the impact of the machine stator resistances. The output voltage controller could do this job as well but only with a delay in response of a change of the requested torque. The Rs compensation algorithm does this immediately and the output voltage controller does not need to be stimulated.

1.1.2 Weak field weakening (FS02 f.weak.PMSM = *weak-FW*)

A frequency dependent field weakening current is applied to the machine to allow higher speeds than rated speed.

This algorithm is less accurate and slower than *1.1.1 Dynamic field weakening* but needs much less computing time.

The field weakening are starts automatically at rated frequency. Parameter **FS03 fw char.low** can be used to shift this point to lower frequencies if requested.

Parameter **FS04 isdmax corr** can be used to adapt the field weakening current by a constant factor. This parameter can be used either to increase or to decrease the voltage consumption.

1.1.3 Strong field weakening (FS02 f.weak.PMSM = *strong-FW*)

A similar field weakening algorithm as in *1.1.2 Weak field weakening* but with a higher field weakening current at high frequencies.

1.1.4 Constant filed weakening (FS02 f.weak.PMSM = *constant-FW*)

The field weakening current in this operation mode is a frequency and torque **independent** value.

It is defined by the parameter **FS04 isdmax corr** as well as by the current $i_{s,d,\psi,comp}$ that is necessary to fully compensate the rotor flux caused by the permanent magnets:

$$i_{s,d,w} = (\text{FS04 isdmax corr}) \cdot i_{s,d,\psi,comp}$$

For example: if FS04 isdmax corr is set 50%, the field weakening current would compensate the rotor flux by 50% at any speed.

Pulse train inputs

The pulse train input functions are mostly the same as for the incremental encoder inputs. For the new extended functions there are additional parameters.

The pulse train input functions can work with any incremental encoder interface as well as with the pulse train interface board. The Z-output is only available with pulse train interface board.

The configuration example below shows how to configure the functions with a pulse train interface board on expansion slot EXP1:

HM03 acquisition			
AQ01 gernal			
ME02	load.en.src	EXP1	

HM03 acquisition			
AQ02 EXP1 settng			
EA01	expans.type	pulseTrainInterf	expansion board type
EA02	IE sig.type	<i>quadrature signl</i>	Quadrature-mode for the signal type of inputs A and B are quadrature type
		<i>step + direction</i>	step-and-direction-mode A = pulse input B = direction input
EA03	IE line cnt	1000	Number of pulses per motor revolution if in step-and-direction-mode or number of quadrature periods if in quadrature-mode
EA04	IE timrMeas	on	Use time measurement to increase the quality of the speed signal. Only important if speed feed forward is used.
EA05	IE TtimMeas	5ms	Time constant to adapt the speed of the time measurement to the incremental speed. Normally no need to modify this value
EA08	Z-out.Tmin	0.1 ms	Minimum output time for index output 'high'. Automatically delays a high-to-low transition of the Z-output if pulse length was too short. Value 0 is allowed.
EA09	Z-out.width	0.001000 rev	Minimum position range in revolutions

			where Z-output is 'high'. Range is symmetrical around the zero position.
--	--	--	--

HM05 drive ctrl.**AS01 oper. manag**

BF01	ctrl mode	pos.diff.control	Control mode position-difference-control to let the motor follow the pulse signals
------	-----------	------------------	--

HM06 setp.proces**SW05 posit.proc**

LM06	sync.posdif	on	Should be set to 'on', so that the position setpoint from the pulse input signal will automatically be synchronized with the motor position when the drive is disabled.
LM09	ratio motor	1 Rev	An additional electronic gear ratio can be applied here if necessary
LM10	ratio load	1 Rev	

Actual values update rates and filter time constants

(document version v03 - applies to ASD software version v10r020)

Update time intervals

Possible time values for update intervals are:

- 125µs - closed loop controller sampling time. This time value is not fix and can be higher or lower
- 2.5µs - open loop control interrupt. This time value is fix.
- 100ms - main loop cycle time. This time value is not fix and can be higher or lower.

Display filtering

Actual values are filtered up to two times until they are displayed:

- filter 1 has an internal filter time that is not meant to be modified for display purpose
- filter 2 is a further 1st order filter with variable time. The variable filter time can be modified with the parameter HM03->AQ01->ME08 dpl.val.fil

actual value		unit	update rate	filter 1	filter 2
IW01	out.current	A	100ms	2.5ms averaging interval	variable filter time
IW02	out.voltage	V	2.5ms	2.5ms averaging interval	variable filter time

IW03	stator freq	Hz	100ms	-	variable filter time
IW04	el.rot.freq	Hz	100ms	n-filter time DR04	variable filter time
IW05	motor speed	rpm	2.5ms	n-filter time DR04	variable filter time
IW06	motor pos	stps	100ms	-	-
IW07	motor pos	incr	125μs	-	-
IW08	load speed	rpm	2.5ms	-	variable filter time
IW09	load pos	stps	100ms	-	-
IW10	load pos	incr	125μs	-	-
IW11	nLd.mot.rel	rpm	2.5ms	-	variable filter time
IW12	torque	Nm	2.5ms	2.5ms averaging interval	variable filter time
IW13	torq.transd	Nm	100ms	2.5ms averaging interval	variable filter time
IW14	ext.speedSP	rpm	100ms	-	-
IW15	ext.pos.SP	stps	100ms	-	-
IW16	ext.pos.SP	incr	100ms	-	-
IW17	pos.ramp.SP	stps	100ms	-	-
IW18	pos.ramp.SP	incr	100ms	-	-
IW19	ext.torq.SP	Nm	100ms	-	-
IW20	dc-link vol	V	2.5ms	Vdc filter time SR08	-
IW21	dc-link cur	A	2.5ms	-	-
IW22	HS-temp.1	*C	multiple of 2.5ms	-	-
IW23	HS-temp.2	*C	multiple of 2.5ms	-	-
IW24	HS-temp.3	*C	multiple of 2.5ms	-	-
IW25	HS-temp.4	*C	multiple of 2.5ms	-	-
IW26	HS-temp.5	*C	multiple of 2.5ms	-	-
IW27	HS-temp.6	*C	multiple of 2.5ms	-	-
IW28	HS-temp.7	*C	multiple of 2.5ms	-	-
IW29	HS-temp.8	*C	multiple of 2.5ms	-	-
IW30	motor temp.	*C	multiple of 2.5ms	-	-
IW31	PCB temp.	*C	multiple of 2.5ms	-	-
IW32	drive state		-	-	-
IW33	limit.state		-	-	-
IW34	int.ctrl.md		-	-	-
IW35	brake state		-	-	-
IW36	error state		-	-	-

IW37	warning		-	-	-
IW38	Ls-id.state		-	-	-
IW39	Kpi-corLsld	%	-	-	-
IW40	RPR id.stat		-	-	-
IW41	RPR id.angl	incr	-	-	-
IW42	PID act.val	%	100ms	-	variable filter time
IW43	PID setpnt	%	100ms	-	variable filter time
IW44	Lsq corr UV	mH	-	-	-
IW45	Rs corr UV	Ohm	-	-	-
IW46	tot.m.inert	kgm2	-	-	-
IW47	isd Psi-cmp	A	-	-	-
IW48	iSdw	A	100ms	-	variable filter time
IW49	iSdwcorr	A	100ms	-	variable filter time
IW50	iSd	A	100ms	-	variable filter time
IW51	iSqw	A	100ms	-	variable filter time
IW52	iSqwcorr	A	100ms	-	variable filter time
IW53	iSq	A	100ms	-	variable filter time
IW54	contr.int.	us	100ms	-	-
IW55	timer.int.	us	100ms	-	-
IW56	sat.id.stat		-	-	-
IW57	sat75	%	-	-	-
IW58	sat100	%	-	-	-
IW59	sat125	%	-	-	-
IW60	no load cur	A	-	-	-
IW61	rotor timec	ms	2.5ms	-	-
IW62	contrl.freq	Hz	-	-	-
IW63	resolv.freq	Hz	-	-	-
IW64	REScur.exp1	mA	-	-	-
IW65	REScur.exp2	mA	2.5ms	-	-
IW66	R1 cos.Gain		2.5ms	-	-
IW67	R1 cosOffst		-	-	-
IW68	R1 sinOffst		-	-	-
IW69	R1 cosPhase		-	-	-
IW70	R2 cos.Gain		-	-	-
IW71	R2 cosOffst		-	-	-
IW72	R2 sinOffst		-	-	-
IW73	R2 cosPhase		-	-	-
IW74	EXP1 Vin	mV	100ms	-	-
IW75	EXP2 Vin	mV	100ms	-	-
IW76	line volt.	Vrms	2.5ms	1ms	variable filter time
IW77	CRLY ON LIM	V	2.5ms	-	variable filter time
IW78	Operat.Mode		-	-	-
IW79	Td-id.state		-	-	-

ASD - Parameters for torque control with external torque transducer

The torque controller is a superimposed PI controller that makes sure that the real actual torque signal matches the torque setpoint signal.

Before closed loop torque control operation (operation without transducer) is tested, It should be made sure that open loop torque control is working fine (**BF01 ctrl mode** = "torque control").

Configuration can be done in the following steps:

-Set the correct operation mode: **BF01 ctrl mode** = "torque cntr.ext"

-Select the analogue input at which the torque transducer is connected:

TC01 input src	<i>no selection</i>	Input for external torque transducer. This defines the input to be used for the connection of an optional torque transducer used to provide a physically measured feedback signal for the closed loop torque controller
	<i>analog in1</i>	
	<i>analog in2</i>	
	<i>analog in3</i>	

-Configure the analog input by adjusting the analog input parameters that are associated with **TC01 input src** like **AI22 scale ana2** = 10 V or **AI26 timec.ana2** = 5 ms. A higher filter time will result in less dynamic but also in less noise.

-Set the scaling and the offset compensation of the torque transducer signal:

TC06 trq.scaling	<i>631.11 Nm</i>	External torque transducer scale factor. This value is the torque for 100% of the analogue selected input range. Note that the analog input range itself must be set correctly (parameter AI22 scale ana2 for example)
-------------------------	------------------	--

-Set the correct external torque filtering / signals delay. This is to internally adjust the controller gain. The internal analog input filtering (like **AI26 timec.ana2**) is automatically considered and must not be added to this value.

TC08 torque fil.	<i>0.5 ms</i>	External torque transducer filter time constant. This is the time constant of the external torque transducer or a filter in the analogue signal path external to the ASD control board. It is used to optimise the closed loop torque controller
-------------------------	---------------	--

-Offset calibration and elimination of polarity errors. The offset can be adjusted easily by adjusting the actual measured torque value **IW13 torq.transd** to 0 Nm during a no load situation:

TC07 trq.offset	0.11 Nm	External torque transducer signal offset adjustment
TC02 invert trq.	on / off	Torque transducer signal inversion. This allows a negative going signal to be treated as positive torque if required.

-Adjust the maximum allowed impact of the torque controller:

TC05 trq.imp.abs	50 %	Closed loop torque controller impact limit. This is the maximum amount of torque that the closed loop torque controller can add to the internal open loop torque controller reference. 100% is equivalent to motor rated torque.
-------------------------	------	--

-Adjust the controller torque controller gains. The controller gains are internally calculated and can be set to 100%. If the motor noise is too big, this values can be simply reduced.

TC03 prop.g.corr	100 %	Closed loop torque controller proportional gain correction factor. This is a correction factor applied to the theoretically optimum proportional gain value.
TC04 int.g.corr	100 %	Closed loop torque controller integrator gain correction factor. This is a correction factor applied to the theoretically optimum integrator gain value.

-the scaling and overall function of the torque transducer can roughly be checked if open loop torque control is used and **IW13 torq.transd** is compared with the expected torque value of **IW12 torque**.

-All other relevant parameters like maximum / minimum torque and speed etc. are identical to the operation without torque transducer.

Encoder less operation with ASD and PMSM. Revision 0.003

This document gives a short explanation of how to use the encoder less features of ASD in combination with a PMSM.

The parameter list (within this file) has to be studied carefully for a basic understanding.

1. Techniques

There are 4 different techniques that go together with encoder less operation with ASD and PMSM.

1.1 **EMF** (electromagnetic force) monitoring

EMF monitoring works only in the middle and high speed range. Rotor position estimation by monitoring the EMK

1.2 **HF** (high frequency) voltage injection

HF voltage injection works only in the low and middle speed range and only if $L_d \ll L_q$. Rotor position estimation by evaluation of position dependent inductances.

1.3 **MM** (mechanical model) driven position estimation

MM driven position estimation works in the full speed range but with poor performance. A high stabilization current is applied to the machine all the time so there are high losses with that technique.

1.4 **LF** (low frequency) current injection for Saturation direction detection

HF voltage injection only works between 0° and 180° . To get the position value in the range of 0° to 360° a low frequency direct-axis-current is injected to measure the direction of the saturation caused by the permanent magnets. LF current injection starts automatically in HF mode and is active for a very short time after enabling the drive.

2. Operation modes

There are 3 basic operation modes and various test modes.

Operation modes:

no encod.less op	No encoder less operation with PMSM
mode 1 (EMF+HF)	Use EMF monitoring at high speeds and HF injection at low speeds. Standard operation mode for Motors with $L_d \ll L_q$.
mode 2 (EMF+MM)	Use EMF monitoring at high speeds and mechanical model at low speeds. Standard operation mode for Motors with $L_d = L_q$.
mode 3 (EMF)	Use EMF monitoring always. Not recommended. Only to use under very certain conditions where zero speed can be avoided.

In Listen modes, torque is always zero so it can be used during first-time operation and to test the encoder less functions:

listen HF zeroSP	HF voltage signal (without LF current signal) is injected. Measured speed signal is forced to zero. To test the ability to use HF voltage injection.
listen HF only	HF voltage signal (without LF current signal) is injected. To test HF signal quality.
listen LF	HF voltage signal and a constant LF current signal is injected. To test LF signal quality.
listen HF (+LF)	HF voltage signal and a short time LF current signal is injected. To test combination of HF + LF signal.
listen EMF	To test EMF monitoring
listen EMF+HF	To test combination of EMF monitoring + HF injection

3. Parameter description

EL01	EncLesOpMod	Selection of encoder less operation modes in case of PMSM and encoder source = no encoder.
EL02	Hspeed freq	Frequency (in % of rated frequency) at which the high speed operation mode (like EMK evaluation) is active.
EL03	Lspeed freq	Frequency (in % of rated frequency) at which the low speed operation mode (like HF injection if configured in EL01) is active.
EL04	trans.time	Transition time between low and high-speed operation in % of an internal value.
EL05	tran.errors	Defines whether an invalid transitions between encoder less operation modes should generate an error. Strictly Recommended if using HF injection!
EL06	deadtimecmp	Uses the dead time compensation table to increase the quality of encoder less operation. Can work in parallel to SR09 deadtimecmp.
EL07	HspeedStart	Make it possible to enable the drive if the motor is already running
EL08	EncL.Backup	Use pure encoderless operation mode as a backup if the main encoder has an error (error reset will be required)
EL09	EncL.DualMd	Use position information of encoderless to increase the accuracy of encoder signals
EL10	DM.RPR ID	In dual mode: Position reference angle identified by encoderless operation
EL11	DM.PosImpct	In dual mode: Maximal position impact of the encoderless position estimation in 0.1°
EL12	DM.frqImpct	In dual mode: Maximal encoderless frequency impact in 0.1%
EL13	DM.trackTim	In dual mode: time constant of the tracking of the encoder signals with an error that is bigger EL11 DM.PosImpct
EL20	EMF filter	Major filter time in case of encoder less operation in EMF monitoring mode.
EL21	EMF damp.	Damping factor in EMF monitoring mode.
EL22	EMF a-track	Modification of the acceleration-tracking rate of position estimation in EMF monitoring mode.
EL30	HF freq.	Frequency of injected HF signal. Automatically limited to the highest possible value.

EL31	HF voltage	Injected HF signal voltage (RMS phase-to-phase).
EL32	HF filter.	Major filter time in case of encoder less operation in HF injection mode.
EL33	HF damping	Damping factor in HF injection mode.
EL34	HF a-track.	Modification of the acceleration-tracking rate of position estimation in HF injection mode.
EL35	HF iq delay	Additional Filtering of the current setpoint in % of EL22 HF n-filt.
EL36	HF monitor.	Signal-strength-low-threshold of HF injection feedback to generate an error. In % of the internal voltage normalization.
EL37	HF sStrenth	Actual signal strength of HF injection feedback in % of the internal voltage In % of the internal voltage normalization.
EL40	LF freq.	Injected LF signal frequency to detect the d-axis direction during HF injection startup
EL41	LF Current	LF signal current in % of the rated motor current.
EL42	LF monitor.	Signal-strength-low-threshold of LF injection, In % of the saturation difference between +d and -d axis.
EL43	LF sStrenth	Detected saturation difference between +d and -d axis in % during LF injection
EL50	MM max accl	Limitation of the maximum acceleration in mechanical model operation mode due to limitation of the q-current in % of the rated q-current/torque.
EL51	MM stabCurr	Static stabilization current in % of the rated current.
EL52	MM dynStabC	Dynamic stabilization current added to the static. in % of the q-current
EL53	MM J-correc	Moment of inertia correction in the mechanical model.

4. Encoder less operation of a salient pole PMSM (first-time operation)

4.1 Preparations

- ME01 mot.enc.src must be set to no encoder.
- For best encoder less operation behavior, the motor parameters should be as accurate as possible. Therefore Use the parameters *MP20* to *MP24* if available in the motor datasheet. Execute Ls-identification to measure the inductance value AFTER modifications on the motor parameters.
- RPR-identification is not needed
- Fast current sensor response is essential. Make sure that the power stack already has the hardware modifications to increase the current sensor bandwidth (other resistors and capacitors)
- For best behavior, the power stack rating should match the rated motor current as good as possible. For example: do not use a 10A motor on a 125A power stack.
- HF injection mode runs best with high control frequencies. To allow this, set the switching frequency to a high value and allow to control twice per switching frequency (*RE01 double.cntr = on*).

4.2 Procedure during first-time operation step 1: Test EMF monitoring

Set parameters to:

EL01	EncLesOpMod	mode 2 (EMF+MM)
EL02	Hspeed freq	25%
EL03	Lspeed freq	20%
EL04	trans.time	100%
EL05	tran.withSP	off
EL06	tran.errors	on
EL20	EMF filter	4ms
EL21	EMF damp.	1
EL22	EMF a-track	200%
EL50	MM max accl	50%
EL51	MM stabCurr	40%

EL52	MM dynStabC	100%
EL53	MM J-correc	100%

- Set operation mode to speed control. And run the motor at very low speeds. The current consumption should be approximately 40% of the rated current.
- Accelerate slowly. At 25% of the rated speed the motor the Operation mote should change to EMF mode. The current consumption should be close to zero.
- Accelerate fast between the two operation modes to check if transition is working properly.

4.3 Procedure during first-time operation step 2: Test HF injection

Do the Tests below in the following order. **Each test has to be finished successfully before executing the next!!!**

If all test of are done successfully, encoder less operation is ready to use with the motor in torque, speed and position control operation modes. Recommended encoder less operation mode during operation is mode 1 (EMF+HF)

Set parameters to:

EL01	EncLesOpMod	<i>listen HF zeroSp</i>
EL02	Hspeed freq	25%
EL03	Lspeed freq	20%
EL04	trans.time	100%
EL05	tran.withSP	off
EL06	tran.errors	on
EL20	EMF filter	4ms
EL21	EMF damp.	1
EL22	EMF a-track	200%
EL30	HF freq.	1234Hz

EL31	HF voltage	50V
EL32	HF filter.	4ms
EL33	HF damping	0.5
EL34	HF a-track.	200%
EL35	HF iq delay	0%
EL36	HF monitor.	0.2%
EL40	LF freq.	100Hz
EL41	LF Current	50%
EL42	LF monitor.	1%

4.3.1 Test of HF feedback signal strength

HF feedback signal strength is depending on the strength of the HF voltage and the relative induction difference $((L_q - L_d) / L_q)$ of the machine.

Set the parameter 'MP11 Lsd/Lsq' to the induction ratio L_d/L_q . The exact value is not important, only whether it is bigger or smaller than 1. If the ratio is completely unknown, it is recommended to set it to a value slightly below 1 to start with.

- Set *EL01 EncLesOpMod = listen HF zeroSp*
- Enable the drive in speed control. The motor will not turn because torque is always zero in listen modes.
- There must be an audible HF signal.
- *EL29 HF sStrenth* should be stable and display a value of 1% or more. If not, adjust *EL21 HF Voltage*.
- MP12 Lsd/Lsq ID displays an identified induction ratio whereas MP14 Lsq UV ID is displaying the stator inductance Lsq. These values can optionally be used as motor parameters MP11 Lsd/Lsq and MP13 Lsq UV
- Disable the drive and enable it with the setting *EL01 EncLesOpMod = listen HF only*. No error message must occur.

Try different values for EL20 HF freq. and EL21 HF Voltage if test is not successful.

4.3.2 Test of LF feedback signal strength

LF feedback signal strength is depending on the strength of the LF current and the physical pre-saturation level of the motor. High LF frequencies *EL30 LF freq.* can reduce the signal strength slightly.

- Set *EL01 EncLesOpMod = listen LF*
- Enable the drive in speed control. The motor will not turn because torque is always zero in listen modes.
- There must be an additional audible 100Hz signal as well as the HF signal.
- *EL39 LF sStrenth* must display a value of typically 2% (or more). If not, adjust *EL31 LF Current*.

If it is not possible to get a reasonable signal strength *EL39 LF sStrenth*, it is possible that the induction ratio L_d/L_q was configured inverse to the actual value. Set *MP11 Lsd/Lsq* to its reciprocal value (for example 1.25 instead of 0.8) and go back to test 4.3.1.

4.3.3 Test of combination of HF voltage injection and LF current injection

- Set *EL01 EncLesOpMod = listen HF (+LF)*
- Enable the drive in speed control. The motor will not turn because torque is always zero in listen modes.
- Turn the shaft by hand and see if speed and position information changes. No error must occur.

4.3.4 Test of encoder less operation with enabled drive

- Set *EL01 EncLesOpMod = mode 1 (EMF+HF)*
- -set operation mode to speed control and run the motor at a very low speed. Speed fluctuations because of the cogging torque are normal.
- -accelerate slowly. At 25% of the rated speed the motor the inter operation mode should change from HF injection to EMF evaluation. The audible noise from the HF injection will disappear.
- -accelerate fast between the two operation modes to check if transition is working properly.

5. Encoder less operation of a non-salient pole PMSM (first-time operation)

Follow 4.1 and 7. If the tests are done successfully, encoder less operation is ready to use with the motor in torque, speed and position control operation modes.

Recommended encoder less operation mode: mode 2 (EMF+MM)

6. Extended Functions

Normally encoderless operation is only active if there is no motor encoder configured. Extended functions allowing to use encoderless operation if there is an encoder error (encoderless backup modes) and/or in parallel to the motor encoder evaluation (dual modes).

6.1 Encoderless backup

The Encoderless backup functions allowing encoderless operation in case of an encoder error without the need of changing the parameterization. There are 3 different presets that only differ in the way how the transition between encoder based operation and encoderless operation is working.

During encoderless backup mode, the drive operates in the encoderless mode that is parameterized in 'EL01 EncLesOpMod'. The warning message 'Enc.less backup' appears.

The input function 'backup mode rst' (edge triggered) can be used to reset the encoderless operation mode and go back to encoder based operation if the encoder evaluation is free of errors. This behavior is the same for all encoderless backup presets.

Encoderless backup modes are:

EL 08 EncL.Backup	description
<i>off</i>	Encoderless backup is not used
<i>after err. reset</i>	Standard backup mode. An encoder error causes the drive to trip. The drive automatically operated in encoderless backup mode after an error reset.
<i>auto transition</i>	Automatic transition mode. There is no trip of the drive after an encoder error. Setpoints are automatically blocked and released again after encoderless operation is initialized.
<i>Direct transmis.</i>	Direct transition mode - not recommended - A direct transition between encoder and encoderless operation is

	executed after an encoder error. This is only possible if the dual-mode configuration (see below) allows simultaneous encoderless and encoder based operation without tracking (EL09 Encl.DualMd = <i>on</i> or EL09 Encl.DualMd = <i>Encoder only</i>) if not, automatic transition mode is used automatically.
--	---

6.2 Dual modes

Dual modes allowing to use encoderless operation in parallel to encoder based operation. Benefits of this combination can be:

- Encoder signals can be used for safety reasons. The accuracy can be increased with the help of encoderless operation.
- Faster encoderless start-up in HF-injection operation mode by eliminating a possible 180° offset with the help of the encoder signals instead of using low frequency current injection.
- Incremental encoder positions can be initialized with the help of encoderless operation.
- Identification of the rotor position reference angle of the motor encoder without the need of the RPR identification procedure.

Encoderless dual modes are:

EL09 Encl.DualMd	description
<i>off</i>	Dual mode is switched off
<i>on</i>	Dual mode is switched on without tracking - NOT recommended if sensorless signals are not absolutely stable - Encoder and encoderless signals are constantly evaluated. Encoderless evaluation runs independent to encoder. The position value is taken from the encoderless evaluation but are limited to the encoder position $\pm 0.5 \cdot \text{'EL11 DM.PosImpct'}$. Even so the encoderless evaluation runs independent to encoder, the encoder position is used to define a possible 180° offset in HF injection mode to avoid LF current injection.
<i>on + tracking</i>	Dual mode is switched on with tracking

	- recommended dual mode setup for most applications - Encoder and encoderless signals are constantly evaluated. Encoderless evaluation internally follows the encoder position with a maximum allowed difference of $\pm 0.5 \times \text{'EL11 DM.PosImpct'}$.
<i>Encoderless only</i>	Pure encoderless operation. Encoder evaluation runs in background only. - this is mainly a maintenance feature to test encoderless operation while a motor encoder is configured or to verify the rotor reference angle - Rotor position reference angle of the motor encoder is displayed at 'EL10 DM.RPR ID'
<i>Encoder only</i>	Pure encoder based operation. Encoderless evaluation runs in background only. - this is mainly a maintenance to verify the rotor reference angle - Rotor position reference angle of the motor encoder is displayed at 'EL10 DM.RPR ID'
<i>Enc.less startup</i>	Pure encoder based operation. Encoderless evaluation runs only for a very short time during first enable of the drive and only in combination with an incremental encoder. - recommended dual mode if a PMSM is used in combination with an incremental encoder - If an encoder index signal has occurred before the first drive enable, encoderless operation is not executed at all.

Parameter Identification (Auto-tuning)

Induction motor systems

There are two separate functions to automatically identify motor parameters.

Controller gain ('Ls-identification')

This function determines the value for parameter SR03 Current controller proportional gain correction. The motor shaft does not rotate during this identification process.

Saturation/No-Load-current identification ('satur. Identific')

This function determines values for the following parameters

MP07 No-load current

SA01 Saturation degree at 50 % rated magnetizing current

SA02 Saturation degree at 75 % rated magnetizing current

SA03 Saturation degree at 100 % rated magnetizing current

SA04 Saturation degree at 125 % rated magnetizing current

WARNING During this identification process the motor will operate in speed control mode with the motor shaft rotating at approximately half of rated speed. Before beginning this process, disconnect any mechanical load from the motor shaft and ensure that the motor shaft is free to turn without risk of injury to persons or damage to property. Remove or secure any key fitted to the motor shaft.

The system must have been configured and tested to operate in closed loop speed control mode before commencing this identification process.

Synchronous motor (PMSM) systems

RP-identification

What else?

Activating the identification processes

Each of the identification processes is controlled by assigning the function to a digital input and then operating that input.

Choose an unused digital input and assign the required to identification function to it. See parameters TI05 – T147 regarding the assignment of functions to digital inputs. The available identification functions are:

Ls-identification

RP-identification

satur. identific

Do not assign an identification function to the enable terminal.

To start and end an identification process:

1. En-input must be 0.

2. The digital input terminal that has the required identification function assigned to it must change from 0 to 1
3. Change En-input to 1.
4. After process is finished set both En-input and associated input terminal to 0

Note that the above description is based on the various inputs being configured as active high (default). The identification process will terminate immediately if the En-input becomes inactive.

The status of the identification process can be checked as an *Actual Value* on the HMI.

Process	Actual value for process status	Display for identification in progress	Display for identification complete
LS-identification	IW38 Ls-id.state	??	LS-ident accomp
RP-identification	IW40 RPR id.stat	??	??
satur. Identific	IW56 sat100	??	LshSatID accomp

[PN] For the next software version I might implement that you can see it on the HMI LEDs if the identification process is still going and I might add the function to start identification with a HMI soft button.

Note that for Saturation/No-Load-current identification the motor will automatically run in speed control at half the rated speed. Because of that do not start this identification if speed control is not working or the motor is connected to a machine like in the lead screw application!

Display Messages

Error Messages

Error message	Error description
overspeed	over speed
time-out br.open	time-out-error during opening the brake
time-out br.close	time-out-error during closing the brake
overcurrent	overcurrent
overcurrent ext.	overcurrent measured by external current sensors
overvoltage dc	Vdc overvoltage
ADC override	overdrive of the ADCs
hardware error	hardware-detected error
proc.sync.mism.	hardware error synchronisation
n-tracking error	Speed following error
emergency stop	emergency stop completed
power supply 5V	Power supply supervision error at 5V-supply (power stack)
pwr.sply 12V/24V	Power supply supervision error at +/-12V or 24V-supply
HWoverc.Isum ext	Sum current error on external current sensor interface
HW overc. I1 ext	over current phase U on external current sensor interface
HW overc. I2 ext	over current phase V on external current sensor interface
HW overc. I3 ext	over current phase W on external current sensor interface
overvoltage PS	over voltage (power stack)
PS Desat error	error DESAT (power stack)
PS overtmp.Sens1	Power stack over temperature detected by sensor 1
PS overtmp.Sens2	Power stack over temperature detected by sensor 2
PS overtmp.Sens3	Power stack over temperature detected by sensor 3
PS overtmp.Sens4	Power stack over temperature detected by sensor 4
PS overtmp.Sens5	Power stack over temperature detected by sensor 5
PS overtmp.Sens6	Power stack over temperature detected by sensor 6
PS overtmp.Sens7	Power stack over temperature detected by sensor 7
PS overtmp.Sens8	Power stack over temperature detected by sensor 8
undervoltage dc	Vdc voltage too low
NTC-overtmp.mot	over temperature motor
pos.tracking err	position following error
CAN comm.	CAN-communication error
PTC-overtmp.mot	PTC over temperature motor
dc link unchargd	DC link not charged. Can be caused by different parametrization errors like setup of the charging contactor thresholds
earthfault PS	earth fault detected (power stack)
dynmc brake err	fault in the brake resistor control unit detected (power stack)
charge relay err	fault of the slow charging contactor detected (power stack)
overtmp. PS HW	over temperature power stack (hardware error)
wrong powerstack	wrong power stack detected during power up
wrong exp1 board	wrong expansion board 1 detected during power up
wrong exp2 board	wrong expansion board 2 detected during power up
Vab too low	Line voltage Vab under voltage
Vbc too low	Line voltage Vbc under voltage
Vab no AC	Line voltage Vab zero crossing error detected
Vbc no AC	Line voltage Vbc zero crossing error detected

ENC Alarm	Alarm Signal received from the on board incremental encoder evaluation
EXP1 Alarm	Alarm Signal received from expansion slot 1
EXP2 Alarm	Alarm Signal received from expansion slot 2
restart required	Changes in configuration that requires a restart
parametrError EXP	Invalid parameterization for expansion slot configuration
no currInterface	External current sensors are selected but not available
EXP1 resolver OC	Resolver excitation current limit reached on expansion slot 1
EXP2 resolver OC	Resolver excitation current limit reached on expansion slot 1
EXP1 resol.NoSig	resolver signals lost on expansion slot 1
EXP2 resol.NoSig	resolver signals lost on expansion slot 2
EXP1 HIPER.NoSig	Hiperface signals lost on expansion slot 1
EXP2 HIPER.NoSig	Hiperface signals lost on expansion slot 2
EXP1 SINCOSnoSig	sin/cos encoder signals lost on expansion slot 1
EXP2 SINCOSnoSig	sin/cos encoder signals lost on expansion slot 2
EXP1 Z1track Err	overcurrent on an external current sensor
EXP2 Z1track Err	invalid I2C configuration or communication
I2t motor overc.	error external current sensor voltage supply
mul.trm.fct.aloc	a terminal function is allocation to multiple terminals
EEPROM error	EEPROM not detected during start-up
I2C error	Communications error on I2C bus
CurrSensorSupply	Power supply error for external current sensors
HMI connection	HMI connection error
HF inject. error	Error during encoderless operation: HF signal strength too low
LF sat.direc.err	Error during encoderless operation: not able to detect change in saturation during LF signal injection
encles.inval.tra	Invalid transition between different encoderless operation modes. One out of the two modes might not work properly
I2t power stack	I2t of power stack caused over current
Save Guards act	Save guards have been activated
VdcProt.SW.Estop	Motor terminal short circuit is induced by software Vdc threshold (extreme emergency stop)
VdcProt.HW.Estop	Motor terminal short circuit is induced by hardware Vdc threshold (extreme emergency stop)
AFE feedback	active front end signal lost during operation
main cont.feedb.	(not implemented) main contactor feedback signal lost during operation
mot.contac.feedb	(not implemented) motor contactor feedback signal lost during operation
invalid ID-mode	Identification mode is not allowed in this configuration
inverter startup	Inverter is not ready because start-up time has not elapsed yet
err-reset locked	Error reset is locked temporary (for a predefined time)
err-reset allow.	Error reset is allowed again
J1939_TimeOutSet	J1939 timeout of setpoint messages detected
J1939_TimeOutCon	J1939 timeout of control messages detected

Warning Messages

Warning message	Warning description
overspeed	over speed
overvoltage dc	Vdc overvoltage
PS overtmp.Sens1	overtemperature power stack
undervoltage dc	Vdc voltage too low
NTC-overtemp.mot	over temperature motor
inaccu.RPRresult	inaccurate result of the rotor position reference identification
Enc.less backup	emergency mode: Encoderless mode instead of encoder based operation because of a previous encoder error (can be resettled by a terminal function)

Frequently Asked Questions - FAQ

How do you change the user level?

Parameter

HM15 user contr

UC03 passwrd.HMI = password for level required

How do you set the ASD to operate in the simulator mode?

Parameter

HM20 SIMULATOR

SM01 motor simul =1

SM02 HW simulat. = 1

(software restart required)

Reference diagrams

Speed control loop refM

Position control loop

Position control loop nFF, TFF

Position control loop nFF, TFF, n from ramp

Parameter list

[Put full list of parameters here]

See separate document for parameter list

Document change history

Instruction manual revision date	Change
16 th November 2015	Motor Lsq / Lsd measurement section rewritten
17 th November 2015	Added Pulse train input section Added V/Hz mode parameter section Added Actual value display filtering section
18 th February 2022	Display messages section added