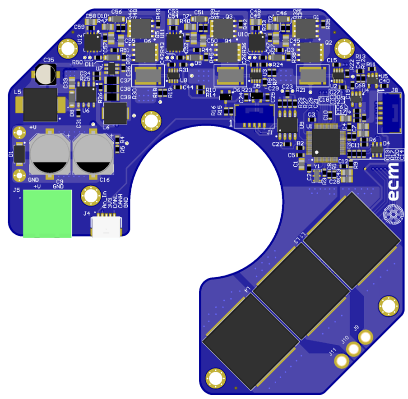
Please read all instructions thoroughly before proceeding to ensure that all steps are understood. Please contact us with any questions at: **techsupport@pcbstator.com**

Ensure that the motor is mounted securely and rigidly before operating.

1 Overview



The shelfstock servomotor is capable of either 0-10V speed control or CANopen-based control. The following guide is intended for a motor configured for CANopen-based control. To operate this motor, you will need a 36V motor power supply and a CANopen capable interface to control the motor.

2 Absolute Maximum Electrical Ratings*

	Minimum	Maximum	Unit
Motor Supply Voltage		38	V
Analog Input Voltage		10.3	V

**Stresses beyond those listed under Absolute Maximum Ratings can cause permanent damage to the device. These stress ratings do not imply functional operation of the device under these or any other conditions beyond those indicated under Recommended Operating Conditions. Exposure to absolute maximum-rated conditions for extended periods may affect device reliability.*

3 Recommended Electrical Operating Conditions*

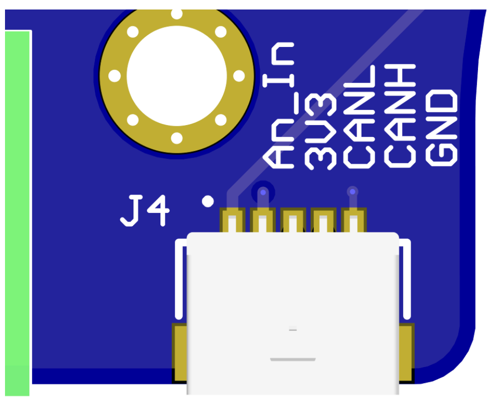
	Minimum	Maximum	Unit
Motor Supply Voltage	10	36	V
Analog Input Voltage	0	10	V

**Operating at a motor supply voltage below 36 V may reduce the maximum achievable motor speed.*

4 Hardware Setup

1. Mount the motor securely and rigidly using the holes in the rear or front of the motor.
2. Supply the motor with 36 VDC using the removable 2 pin screw connector (color and markings may vary). V+ = left side. V- = right side, as shown.

3. Use the control connector, J4, to connect to the CAN bus. Connect CANL to Pin 3. Connect CANH to Pin 4. Connect the ground reference to Pin 5 (GND). If needed, 3.3VDC is provided on Pin 2 (3V3).
   - By default, the board terminates the CAN bus with two 60 Ω resistors in series (R27 and R28, located near J1).
   - If needed, J1 has the same pinout as J4 and may be accessed with the rear cover removed for daisy-chaining purposes.



5 CANopen Communication Parameters

The CANopen interface is configured to operate at a baudrate of 1Mbit/s. By default, the node ID is 0x28.

6 CANopen Object Dictionary

The CANopen object dictionary has been provided to you via the file named "servo_profile.eds". The object dictionary may be viewed graphically with EDSEditor, an open-source program that can be found at `https://github.com/CANopenNode/CANopenEditor`. Below is a short description of some of the key parameters in the dictionary.

- 0x2006: Current Integral Gain
- 0x2007: Current Proportional Gain
- 0x2008: Current Reference Ramp Rate
- 0x2009: Current Reference Filter Pole
- 0x2020: Position Integral Gain
- 0x2022: Position Proportional Gain
- 0x2026: Velocity Integral Gain
- 0x2027: Velocity Proportional Gain
- 0x2044: Drive Temperature
- 0x20A1: Over-voltage Threshold
- 0x20B2: PWM Frequency
- 0x20BE: Encoder Calibration Command
- 0x20C0: Encoder Calibration Status
- 0x20CF: Under-voltage Threshold
- 0x20F2: Analog Input Voltage
- 0x603F: Fault Register
- 0x6040: Controlword
- 0x6041: Statusword
- 0x6060: Mode of Operation
- 0x6061: Mode of Operation Display
- 0x6062: Position Demand Value
- 0x6064: Position Actual Value
- 0x606B: Velocity Demand Value
- 0x606C: Velocity Actual Value

- 0x6071: Torque Target Value
- 0x6074: Torque Demand Value
- 0x6075: Motor Rated Current
- 0x6077: Torque Actual Value
- 0x6079: DC Bus Voltage
- 0x607A: Position Target Value
- 0x6080: Max Motor Speed
- 0x60FF: Velocity Target Value

6.1 Current Integral Gain (0x2006)

- **Access**: RW
- **Bytes**: 4
- **Type**: FLOAT
- **Unit**: N/A
- **Description**: The integral gain for the current controllers. The gain will be appropriately scaled by the sample rate. That is, the register value should **not** be scaled. Ensure that the value is appropriately chosen for pole cancellation ($K_{I,i} = \frac{R}{L}$ where R and L are the motor phase-to-neutral resistance and inductance, respectively, **including** the controller series inductor resistance and inductance).

6.2 Current Proportional Gain (0x2007)

- **Access**: RW
- **Bytes**: 4
- **Type**: FLOAT
- **Unit**: N/A
- **Description**: The proportional gain for the current controllers. Ensure that the value is appropriately chosen for pole cancellation ($K_{P,i} = \omega L$ where L is the motor phase-to-neutral inductance, **including** the controller series inductor inductance).

6.3 Current Reference Ramp Rate (0x2008)

- **Access**: RW
- **Bytes**: 4
- **Type**: FLOAT
- **Unit**: $\mathrm{A\,s^{-1}}$
- **Description**: The current reference ramp rate. The rate will be appropriately scaled by the sample rate. That is, the register value should **not** be scaled.

6.4 Current Reference Filter Pole (0x2009)

- **Access**: RW
- **Bytes**: 4
- **Type**: FLOAT
- **Unit**: $\text{rad}\,\text{s}^{-1}$
- **Description**: The current reference filter pole.

6.5 Position Integral Gain (0x2020)

- **Access**: RW
- **Bytes**: 4
- **Type**: FLOAT
- **Unit**: N/A
- **Description**: The integral gain for the controller used in position control modes. The gain will be appropriately scaled by the sample rate. That is, the register value should **not** be scaled.

6.6 Position Proportional Gain (0x2022)

- **Access**: RW
- **Bytes**: 4
- **Type**: FLOAT
- **Unit**: N/A
- **Description**: The proportional gain for the controller used in position control modes.

6.7 Velocity Integral Gain (0x2026)

- **Access**: RW
- **Bytes**: 4
- **Type**: FLOAT
- **Unit**: N/A
- **Description**: The integral gain for the controller used in velocity control modes. The gain will be appropriately scaled by the sample rate. That is, the register value should **not** be scaled.

6.8 Velocity Proportional Gain (0x2027)

- **Access**: RW
- **Bytes**: 4
- **Type**: FLOAT
- **Unit**: N/A
- **Description**: The proportional gain for the controller used in velocity control modes.

6.9 Drive Temperature (0x2044)

- **Access**: RO
- **Bytes**: 2
- **Type**: INT16
- **Unit**: °C
- **Description**: This drive does not feature a drive temperature sensor so this register will always report 0.

6.10 Over-voltage Threshold (0x20A1)

- **Access**: RO
- **Bytes**: 4
- **Type**: FLOAT
- **Unit**: V
- **Description**: The threshold at which an over-voltage fault will occur.

6.11 PWM Frequency (0x20B2)

- **Access**: RO
- **Bytes**: 4
- **Type**: UINT32
- **Unit**: Hz
- **Description**: The PWM frequency used by the drive. Because the PWM generation is center-aligned, the current ripple will be at twice this frequency.

6.12 Encoder Calibration Command (0x20BE)

- **Access**: RW
- **Bytes**: 2
- **Type**: UINT16
- **Unit**: N/A
- **Description**: Used to send commands to the encoder calibration state machine. Valid commands are 0 - None, 1 - Calibrate Encoder, 2 - Align Encoder.

6.13 Encoder Calibration Status (0x20C0)

- **Access**: RO
- **Bytes**: 2
- **Type**: UINT16
- **Unit**: N/A
- **Description**: Used to read the status of the encoder calibration state machine.

The bit names for the calibration status register are

15	14	13	12	11	10	9	8
				Alignment Error	Digital Calibration Error	Analog Calibration Error	EEPROM Error

7	6	5	4	3	2	1	0
Configuration Error	Communication Error	Aligning	Calibrating Digital Subsystem	Calibrating Analog Subsytem	Configuring	Rebooting	Waiting

6.14 Under-voltage Threshold (0x20CF)

- **Access**: RO
- **Bytes**: 4
- **Type**: FLOAT
- **Unit**: V
- **Description**: The threshold at which an under-voltage fault will be triggered.

6.15 Analog Input Voltage (0x20F2)

- **Access**: RO
- **Bytes**: 2
- **Type**: UINT16
- **Unit**: mV
- **Description**: The most recently measured voltage at the analog input (J4 Pin 1).

6.16 Fault Register (0x603F)

- **Access**: RO
- **Bytes**: 2
- **Type**: UINT16
- **Unit**: N/A
- **Description**: Present faults.

The bit names for the fault register are

15	14	13	12	11	10	9	8
-	-	-	-	-	-	Encoder Fault	-

7	6	5	4	3	2	1	0
Over Current	Under Voltage	Over Voltage	-	-	-	-	-

6.17 Controlword (0x6040)

- **Access**: RW
- **Bytes**: 2
- **Type**: UINT16
- **Unit**: N/A
- **Description**: Used to send commands to the motor controller state machine.

The bit names for the Controlword register are

15	14	13	12	11	10	9	8
-	-	-	-	-	-	-	-

7	6	5	4	3	2	1	0
Fault Reset	-	-	-	Enable Operation	-	Enable Voltage	Switch On

Please see Section 7 for more information on using the controlword register to control the state of the motor controller.

6.18 Statusword (0x6041)

- **Access**: RO
- **Bytes**: 2
- **Type**: UINT16
- **Unit**: N/A
- **Description**: The motor controller state.

The bit names for the Statusword register are

15	14	13	12	11	10	9	8
-	-	-	-	-	-	-	Calibrating ADC

7	6	5	4	3	2	1	0
-	Switch On Disabled	-	Voltage Enabled	Fault	Operation Enabled	Switched On	Ready to Switch On

This motor controller does not feature the ability to disable the voltage to the drive, so the "Voltage Enabled" bit is always set.

The "Calibrating ADC" bit indicates that the controller is calibrating against voltage and current measurement offsets. During calibration, PWM is enabled, but the motor will not receive electrical power.

6.19 Mode of Operation (0x6060)

- **Access**: RW
- **Bytes**: 1
- **Type**: UINT8
- **Unit**: N/A
- **Description**: The desired mode of operation of the motor controller. The valid options are free-spinning (0x00), position control (0x01), velocity control (0x03), torque control (0x04), synchronous position mode (0x08)

In synchronous position mode, the drive will wait for the reception of a SYNC message before updating the internal reference. This allows for synchronization between multiple motors.

6.20 Mode of Operation Display (0x6061)

- **Access**: RO
- **Bytes**: 1
- **Type**: UINT8
- **Unit**: N/A
- **Description**: The current mode of operation.

6.21 Position Demand Value (0x6062)

- **Access**: RO
- **Bytes**: 4
- **Type**: FLOAT
- **Unit**: radians
- **Description**: The internal position reference in mechanical radians. Demand values are rate-limited and low-pass filtered target values.

6.22 Position Actual Value (0x6064)

- **Access**: RO
- **Bytes**: 4
- **Type**: FLOAT
- **Unit**: radians
- **Description**: The actual position of the motor in mechanical radians.

6.23 Velocity Demand Value (0x606B)

- **Access**: RO
- **Bytes**: 4
- **Type**: FLOAT
- **Unit**: RPM
- **Description**: The internal velocity reference in mechanical rotations per minute. Demand values are rate-limited and low-pass filtered target values.

6.24 Velocity Actual Value (0x606C)

- **Access**: RO
- **Bytes**: 4
- **Type**: FLOAT
- **Unit**: RPM
- **Description**: The actual velocity of the motor in mechanical rotations per minute.

6.25 Torque Target Value (0x6071)

- **Access**: RW
- **Bytes**: 2
- **Type**: INT16
- **Unit**: N/A
- **Description**: The target current in torque control mode. The unit is a fraction of the rated motor current.

The equation for the setting the target torque register, REG[0x6071], with a target current, i_{target}, in Amps is

$$\text{REG[0x6071]} = 32768\frac{i_{\text{target}}}{i_{\text{max}}},$$

where i_{max}, is the rated motor current in Amps.

6.26 Torque Demand Value (0x6074)

- **Access**: RO
- **Bytes**: 2
- **Type**: INT16
- **Unit**: N/A
- **Description**: The internal motor current reference. The unit is a fraction of the rated motor current. Demand values are rate-limited and low-pass filtered target values.

The equation for the retrieving the internal motor current reference, i_{demand}, in amps from the torque demand register, REG[0x6074], is

$$i_{\text{demand}} = i_{\text{max}}\frac{\text{REG[0x6074]}}{32768},$$

where i_{max}, is the rated motor current in Amps.

6.27 Motor Rated Current (0x6075)

- **Access**: RO
- **Bytes**: 4
- **Type**: FLOAT
- **Unit**: A
- **Description**: The rated motor current in Amps.

6.28 Torque Actual Value (0x6077)

- **Access**: RO
- **Bytes**: 2
- **Type**: INT16
- **Unit**: N/A
- **Description**: The measured motor current. The unit is a fraction of the rated motor current.

The equation for the retrieving the measured motor current, i_{actual}, in amps from the actual torque register, REG[0x6077], is

$$i_{\text{actual}} = i_{\text{max}} \frac{\text{REG[0x6077]}}{32768},$$

where i_{max}, is the rated motor current in Amps.

6.29 DC Bus Voltage (0x6079)

- **Access**: RO
- **Bytes**: 2
- **Type**: UINT16
- **Unit**: mV
- **Description**: The measured DC bus voltage in millivolts.

6.30 Position Target Value (0x607A)

- **Access**: RW
- **Bytes**: 4
- **Type**: FLOAT
- **Unit**: radians
- **Description**: The target position of the motor in position control mode in mechanical radians.

6.31 Max Motor Speed (0x6080)

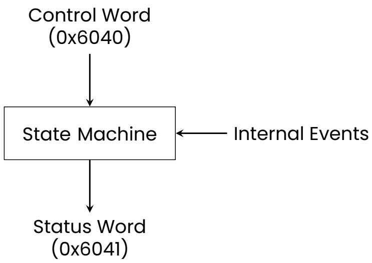
- **Access**: RO
- **Bytes**: 4
- **Type**: FLOAT
- **Unit**: RPM
- **Description**: The maximum allowable motor speed in mechanical rotations per minute.

6.32 Velocity Target Value (0x60FF)

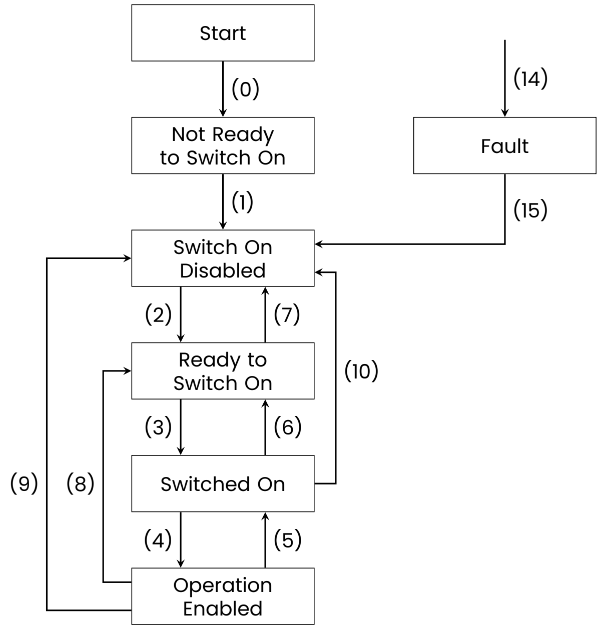
- **Access**: RW
- **Bytes**: 4
- **Type**: FLOAT
- **Unit**: RPM
- **Description**: The target velocity of the motor in velocity control mode in mechanical rotations per minute.

7 Motor Controller State

The motor controller state is determined by the Controlword register (0x6040) and internal events. The state is read via the Statusword register (0x6041).



The state machine describes the state evolution of the device with respect to the control of power electronics due to user commands and internal drive faults.



7.1 States

By reading the Statusword, the current motor controller state may be determined:

Statusword Bit					State	Description
6	**3**	**2**	**1**	**0**		
0	0	0	0	0	Not Ready to Switch On	The drive is being initialized.
1	0	0	0	0	Switch On Disabled	The drive is disabled, e.g., for safety reasons.
0	0	0	0	1	Ready to Switch On	The drive is disabled and ready to be enabled.
0	0	0	1	1	Switched On	The drive is enabled, but PWM is disabled.
0	0	1	1	1	Operation Enabled	The drive is enabled and PWM is enabled.
0	1	0	0	0	Fault	A fault has occurred. The drive is disabled.

The cause of a transition to the Fault state is recorded in the Fault Register (0x603F).

7.2 State Transitions

The possible state transitions are as follows:

	Starting State	Ending State	Description
(0)	Start	Not Ready to Switch On	Device Reset.
(1)	Not Ready to Switch On	Switch on Disabled	The drive has self-initialized successfully.
(2)	Switch On Disabled	Ready to Switch On	"Shutdown" command received from the host.
(3)	Ready to Switch On	Switched On	"Switch On" command received from the host.
(4)	Switched On	Operation Enabled	"Enable Operation" command received from the host.
(5)	Operation Enabled	Switched On	"Disable Operation" command received from the host.
(6)	Switched On	Ready to Switch On	"Shutdown" command received from the host.
(7)	Ready to Switch On	Switch On Disabled	"Disable Voltage' command received from the host.
(8)	Operation Enabled	Ready to Switch On	"Shutdown" command received from the host.
(9)	Operation Enabled	Switch On Disabled	"Disable Voltage" command received from the host.
(10)	Switched On	Switch On Disabled	"Disable Voltage" command received from the host.
(14)	All states	Fault	A fault has occurred in the drive.
(15)	Fault	Switch On Disabled	"Fault Reset" command received from the host.

7.3 Commands

Using the Controlword, the commands that may be sent are as follows:

	Controlword Bit				
Command	**7**	**3**	**1**	**0**	**Transitions**
"Shutdown"	0	X	1	0	(2), (6), (8)
"Switch On"	0	X	1	1	(3)
"Disable Voltage"	0	X	0	X	(7), (9), (10)
"Disable Operation"	0	0	1	1	(5)
"Enable Operation"	0	1	1	1	(3)/(4)*, (4)
"Fault Reset"	1	X	X	X	(15)

**The "Enable Operation" command given in state Ready to Switch On will perform transitions (3) and (4) in succession.*

7.4 Example Power Up Sequence

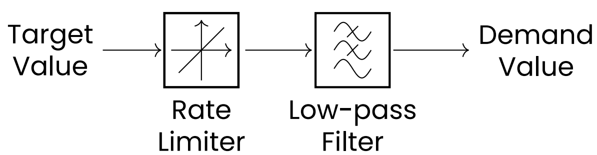
Below is a step-by-step example of how to enter the “Operation Enabled” state via CANopen.

1. Apply power to the motor controller.
2. Configure any desired PDO configuration objects via the SDO service.
3. Configure the desired target values and mode of operation.
4. Configure the NMT state to operational.
5. Send the “Shutdown” command and wait for the motor controller to transition to the “Ready to Switch On” state.
6. Send the “Enable Operation” command and wait for the controller to transition to the “Operation Enabled” state. The motor will begin to perform the requested action.

8 Controllers

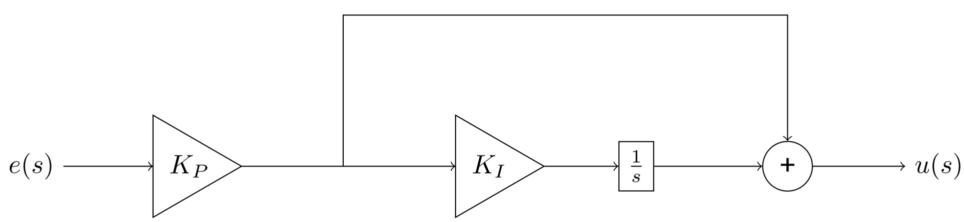
8.1 Controller Reference Values

The reference for a controller can be modified with it’s associated target value, e.g. CANopen register 0x6071 (Torque Target Value) for torque control mode. The actual reference used in the control loop can be read via it’s associated demand value, e.g. CANopen register 0x6074 (Torque Demand Value) for torque control mode. A demand value is a rate-limited and low-pass filtered target value, as shown below



8.2 PI Controller

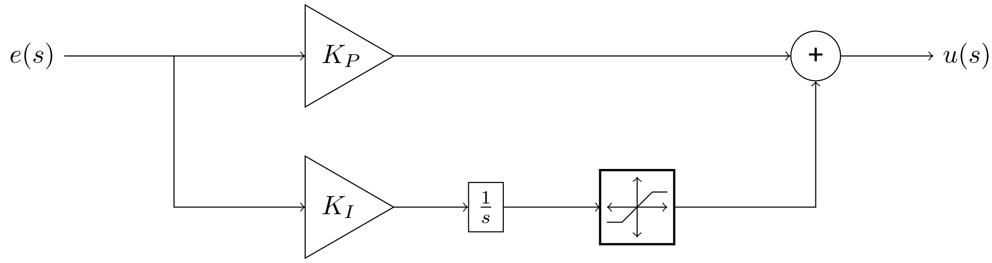
All of the controllers use one or more PI controllers in their operation. There are two forms of the PI controller used in this drive. The first is in the series configuration, shown below,



and has a transfer function of

$$\frac{u(s)}{e(s)} = K_P \left(1 + \frac{K_I}{s}\right). \tag{1}$$

The second is an anti-windup parallel configuration, shown below,



and has a transfer function of

$$\frac{u(s)}{e(s)} = K_P + \frac{K_I}{s}, \quad (2)$$

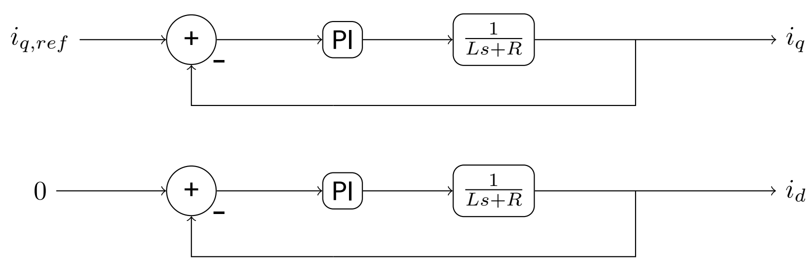
assuming the integral is not saturated.

Integral gains set by the CANopen registers will be appropriately scaled by the sample rate. That is, register values should **not** be scaled.

8.3 Current (Torque) Controller

All controllers feature both an i_q and i_d series PI controller. When the motor is running, the reference for the i_d controller is set to zero.

The control loops, including the stator's electrical transfer function, are shown below



The stator's electrical transfer function, $i(s)/v(s) = 1/(Ls + R)$, where L and R are the phase-to-neutral inductance and resistance, respectively, includes the extra series inductance present on the drive.

The PI controller gains are, by default, chosen to cancel the poles in the stator's transfer function, or

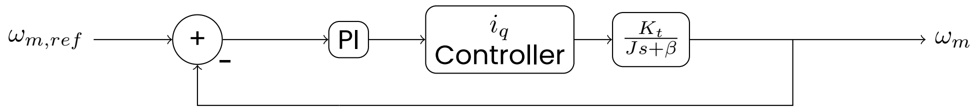
$$K_{P,i} = \omega L$$
$$K_{I,i} = \frac{R}{L}$$

where ω is the desired current controller bandwidth. A higher controller bandwidth results in a faster transient response at the expense of increased noise in the current loop. Increased noise in the current loop may result in audible noise emanating from the motor and/or increased motor losses.

In torque control mode, the reference for i_q can be modified with CANopen register 0x6071 (Torque Target Value). The actual reference used in the control loop can be read via CANopen register 0x6074 (Torque Demand Value).

8.4 Velocity Controller

In velocity or position control modes, a velocity controller is used that sets the i_q controller reference. The control loop uses an anti-windup parallel PI controller. The system, including a model for the motor's mechanical transfer function, is shown below

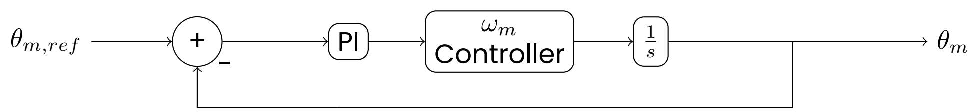


where J is the rotor inertia, β is the drag coefficient, and K_t is the torque constant. If the bandwidth of the i_q controller is high enough and is properly tuned, its dynamics can be ignored when analyzing this system.

In velocity control modes, the reference, $\omega_{m,ref}$, can be modified with CANopen register 0x60FF (Velocity Target Value). The actual reference used in the control loop can be read via CANopen register 0x606B (Velocity Demand Value).

8.5 Position Controller

In position control modes, a position controller that sets the velocity reference is used. The control loop, including the relevant motor mechanical transfer function component, is shown below



If the bandwidth of the ω_m controller is high enough and it is properly tuned, its dynamics can be ignored when analyzing this system.

In position control modes, the reference, $\theta_{m,ref}$, can be modified with CANopen register 0x607A (Position Target Value). The actual reference used in the control loop can be read via CANopen register 0x6062 (Position Demand Value).

9 Encoder Calibration

The internal motor encoder has been calibrated and tested during assembly. In rare cases, a motor that experiences encoder faults may be recovered by recalibrating the encoder. The CANopen interface provides a means to do this through the encoder calibration state machine.

There are two types of encoder calibration: calibration and alignment. Calibration performs analog and digital adjustment of the sensor and can only take place while the motor is spinning. Alignment aligns the encoder counts and the motor's electrical angle and can only take place while the motor is at a standstill.

Please contact us for further information on calibrating the encoder.

10 Thermal Information

This model motor was determined to have a thermal resistance of approximately 1.85 W °C^{-1}. That is, for each 1.85 W of power loss, there should be a 1 °C delta between the housing temperature and the peak stator temperature at steady state. Performance at the expected operating point, based on a sample motor in the lab, is shown in the table below. Individual motor performance may vary. The housing temperature must be monitored during testing.

Speed [RPM]	**Torque [Nm]**	**Mechanical Power [W]**	**System Efficiency [%]**	**Power Loss [W]**	**Expected Delta [°C]**	**Maximum Housing Temperature for 100 °C Stator Limit**
3000	0.5	157.05	73.71	56.03	30.3	69.7

The motor can run at increased temperatures, but the lifetime of the machine will be affected by an unknown amount. ECM cannot predict the lifetime effect of increased temperatures, as this is largely dependent on the final installation and the environmental conditions experienced while in use. The motor should be able to run up to 120 °C stator temperature without too much risk to the motor; however, as with all motors, when running at elevated temperatures, the performance of the motor will change and should be monitored. It is necessary to monitor performance to ensure that it is not in a thermal runaway situation. Preventing a thermal runaway is not something unique to ECM. This means that there are already a variety of established solutions. Solutions such as a thermal shut off switch mounted to housing, software limits on current, directly monitoring stator temp with sensors, etc.