BORGWARNER

CM Software Users Manual

Document: 0A-0163-02

BorgWarner Portland (formerly Cascadia Motion) Software User Manual

Contents

List of Figures

List of Tables

1 Firmware

This manual is directed primarily at the latest inverter family offering from BorgWarner Portland (formerly Cascadia Motion). The CM series inverters all run the same Gen 5 firmware with the same programming target. Other inverter families such as the PM and RM series inverters will need to reference the legacy manuals.

Table 1: Inverter/Firmware Versions

Inverter Family	Processor Target	Inverters	Firmware Family
Gen 5	28277D CPU 1 / 20 MHz	PMxxx-G5 / CM200 / CM350	Version: 65XX

There are firmware versions that are customer/application specific that do not conform the above firmware version numbers.

The firmware is a single file in hexadecimal format that can be downloaded and programmed into the inverter over the serial port. The title of the firmware file follows the date versioning scheme. This scheme uses the year followed by the month and then day. The format is GEN5_CC_CM_yyyymmdd_nnnn_option.hex.

In addition to the date code, the filename contains a software release number (nnnn).

The 'option' refers to specific features. As noted below the main firmware is labelled as Group_1 or Group_2.

An example of a released firmware file would be GEN5_CC_CM_20240829_6531_Group_1.hex

Where,

'20240829' is the date code, August 29, 2024.

'65' is the major release number.

'31' is the minor release number.

Important: The hex file with the tag 'Group_1' in the filename should be used for motor types between 0 and 59. The hex file with the tag 'Group_2' in the filename should be used for motor types starting from 60 and onward.

New firmware is released on a continuous basis. The time to release firmware depends on the new feature requests, change requests, and bug reports discovered internally or by the external customers.

Each firmware release has an accompanying 'Firmware Release Notes' document that provides the following information:

(a) Important notices regarding the new firmware

(b) New features and/or change requests

(c) Bug fixes

1.1 Firmware Release Package

In addition to the Release Notes, the firmware release package contains the following:

(a) Firmware

(b) Tools, typically the RMS GUI and the defsyms files associated with the RMS GUI

(c) Other documentation as needed

The firmware package is uploaded to an online repository. To access the repository navigate to the Cascadia Motion website, www.cascadiamotion.com. Go to the Documentation/Support page.

Previous releases are also made available in the same repository. If a specific release cannot be found, please contact BorgWarner Portland support.

1.1.1 Firmware

The firmware file can be uploaded to the inverter over the serial port (RS-232). The program C2Prog is used to upload the firmware to the inverter. Please see section 1.1.2.2, ‘‘C2Prog’’, for more details.

The serial communication interface is used for three purposes. It is used for firmware download, graphical user interface (GUI) communication, and for serial data acquisition.

At normal power up the inverter enters the serial data acquisition mode. Serial data is transmitted in hexadecimal format. This data can be captured on a PC by using any standard communications software such as Hyper-Terminal or Real-Term. The data can also be captured on any type of device that has a standard serial port. The data can be used to plot specific graphs to understand vehicle performance.

Upon starting the RMS GUI, the controller will automatically transition from serial data acquisition mode to a mode that communicates with the GUI. The RMS GUI program, designed for the Windows platform, is utilized to reprogram EEPROM parameters and monitor data. Upon initiation, the RMS GUI will automatically scan all available COM ports for an inverter, provided the COM port is not already occupied by another program on the PC. The RMS GUI will then attempt to establish communication with the inverter, a process that may take a few seconds. Once communication is established, the RMS GUI will display all parameters that can be monitored and reprogrammed. Additionally, the RMS GUI will display the firmware version and date code of the firmware installed in the inverter. If the GUI cannot locate a defsyms (a symbol defining active parameters for the firmware) with a date code matching the inverter firmware date code, it will flag an error. Please see section 3.2, ‘EEPROM Parameters Guide’, for more details on programming EEPROM parameters into the inverter.

1.1.2 Tools

The RMS GUI and C2Prog applications are provided for interacting with the inverter. Both of these tools can be found on the Cascadia Motion website.

1.1.2.1 RMS GUI

The Release Package includes the RMS GUI application and all necessary defsyms files for it to operate with the firmware in the Release Package. The instructions provided here apply to the current public

version of the RMS GUI (1.4.8). For first-time use, the RMS GUI must be installed. This can be accomplished by downloading and installing the GTK environment (gtk+-2.8.9.-setup-1.exe) from the Cascadia Motion website. After the GTK environment is installed, a PC reboot is required for the changes to take effect.

The GUI program enables the user to monitor various variables and reprogram EEPROM parameters. It's important to note that EEPROM parameters must be programmed before operating the controller. Please see section 3.2, 'EEPROM Parameters Guide' for more information.

It is not necessary to configure the COM port baud rate or other settings for use with the RMS GUI as it will automatically adjust the settings correctly. Please note that the RMS GUI may encounter issues with certain folder names. Therefore, when loading or saving a file using the RMS GUI, ensure that the file is located in the same folder as the RMS GUI.exe.

The following files are related to the use of the RMS GUI:

- **RMS GUI.exe:** This is the main application used to monitor data and reprogram EEPROM parameters.There is no setup file; simply copy this application to a suitable location.
- **defsyms_yyyymmdd.txt:** This default symbols file includes the parameters to be monitored and reprogrammed, depending on the version of hardware in use. It is firmware-specific, meaning each firmware has its own default symbols file. The file can be matched through the date code in yyyymmdd format. The RMS GUI will automatically select the corresponding file if multiple are present.
- **gtk+-2.8.9-setup-1.exe:** This is a one-time installed library file. The computer must be rebooted after the installation. This file can be obtained from the Cascadia Motion website.
- **Conf folder:** The RMS GUI will automatically create a folder named "conf". This folder will contain a record of any EEPROM changes that have been made. Each time the EEPROM is programmed a new file is created. The files are text files that can be easily viewed in a test viewer such as Notepad.

1.1.2.2 C2Prog

C2Prog is a flash programming tool specifically designed for TI C2000™ MCUs. Rather than using JTAG as the communication interface between the programming tool and the MCU, C2Prog utilizes RS-232, RS-485 and CAN (Controller Area Network). This makes the program particularly suitable for field deployment where the JTAG port is typically inaccessible.

The C2Prog flash programmer utilizes the stock boot-loader feature of the MCU for rapid flash programming over serial. The software can be downloaded from the Cascadia Motion website.

BorgWarner Portland inverters are designed to be reflashed using C2prog and the serial (RS-232) port. The inverters will switch to a boot mode if the Program Enable input is grounded when low voltage power is applied to the inverter.

Reflashing the inverter should not alter any EEPROM settings if the firmware belongs to the same family. However, there are non-standard firmware files that will modify the EEPROM. Therefore, it is always a good idea to save your existing EEPROM settings before reflashing an inverter.

The Release Notes provide details on how C2prog should be configured for that particular firmware package.

For the latest version of the application and more details, please visit http://www.codeskin.com

1.1.2.3 Realterm

Realterm is a terminal program specially designed for capturing, controlling and debugging binary and other data streams. It has more features for debugging communication ports than a Hyper-terminal. This application can be used for collecting the streaming data during bench testing.

If it is desired to capture data from the inverter using Realterm then Realterm should be configured for 57600 baud, 8 data bits, No parity.

Some of the features of this application include command line control, ability to capture to file, arbitrary baud rates, etc. For more details, please refer to http://realterm.sourceforge.net/

2 C2Prog: Firmware Programming Guide

2.1 Required Hardware

To successfully program a BorgWarner Portland (formerly Cascadia Motion) inverter, an RS232 cable or RS232-USB Adapter (based on PC's port availability) will be needed.

2.2 Required Software

(a) New Firmware Package, available from www.cascadiamotion.com

(b) C2Prog Software, available here: http://www.codeskin.com/c2prog-download

2.3 Programming Steps

(a) Launch C2Prog and click the '. . . ' button in the Firmware Image box to select the desired firmware image to install onto the inverter. The file will have a .hex extension.

(b) Make sure the proper COM port is selected. Click the 'Select Port. . . ' button, then the 'Scan Ports' button.

(c) Using the 'Target:' pull-down menu select the following target:

- **28379,378,377,375D-CPU01** Using: **SCI**

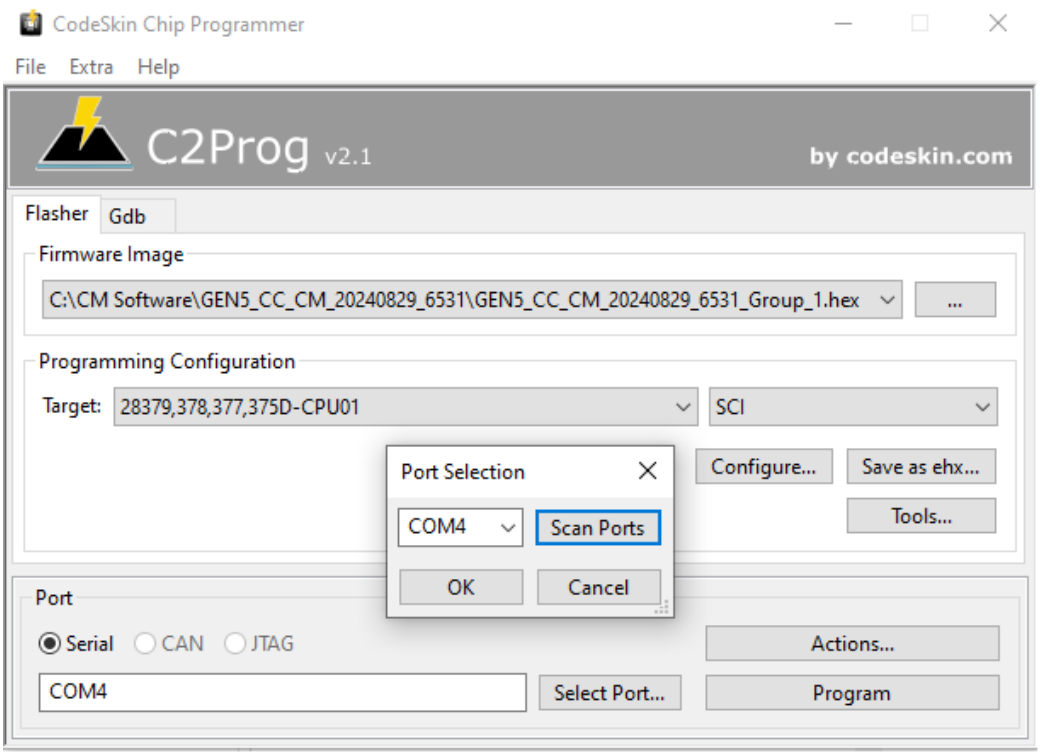


Figure 1: C2Prog Settings

(d) Verify the Target Configuration under the 'Configure...' button is set to the default, as shown in figure 2. Note the hidden password fields should be defaulted to '0xFFFF' and 'Smart Sector Selection' should be checked.

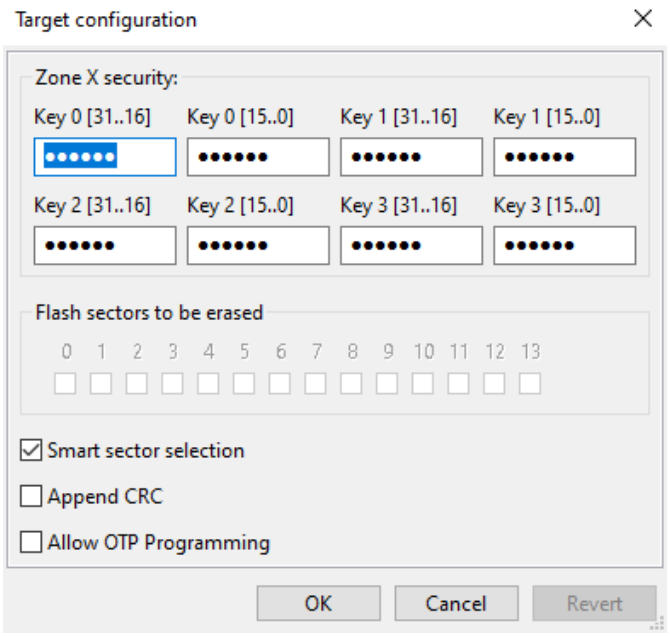


Figure 2: Target Configuration Settings

(e) Turn off the low voltage power to the inverter (typically BATT+ signal). Ground the Program Enable input to the inverter. Then turn on the low voltage power to the inverter.

(f) Now click the 'Program' button near the bottom.

(g) Programming will then begin. The C2Prog software will show the status of the programming.

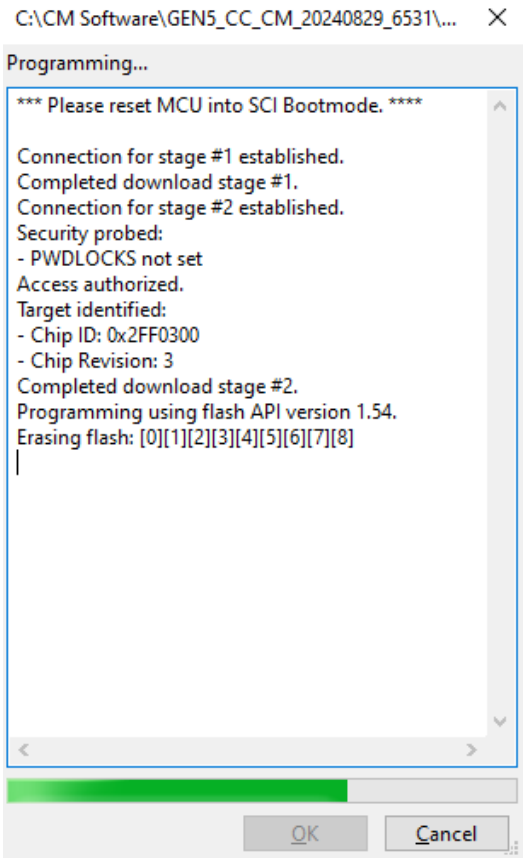


Figure 3: C2Prog Status Window

(h) When the programming is completed, click the 'OK' button to close the Status screen. Turn off the low voltage power to the inverter. Disconnect the Program Enable input from ground. Now when the inverter is repowered the new firmware will be operational.

2.3.1 Troubleshooting

The scenarios listed below are various error states the the C2Prog utility can indicate. With the errors listed are common causes of the error in an attempt to aide troubleshooting. If problems persist please contact BorgWarner Portland support.

C2Prog Status Message:

- **Cannot connect with autobaud:** If this error appears after 5 seconds or so, C2Prog has given up trying to connect, indicating either low voltage power is likely not on or the serial connection is broken. If this error appears immediately it is likely that the Program Enable is not grounded and the inverter is writing to the serial port.
- **Transmit Failed:** Low voltage power to the inverter likely lost during the programming sequence.

3 RMS GUI Usage

3.1 Overview

3.1.1 Required Hardware

An RS232 cable or RS232-USB Adapter (based on PC's port availability) will be needed in order to communicate with a BorgWarner Portland (formerly Cascadia Motion) inverter via RMS GUI.

3.1.2 Required Software

The following software applications/files are needed to program EEPROM parameters:

- RMS GUI Application: This application is part of the Firmware Release Package and can be downloaded using the link provided in the above section, 'Firmware Release Package'.
- Default symbols file (e.g. defsyms_yyyymmdd.txt): Each released firmware requires a specific default symbols file. Please refer to section 1.1.2.1 'RMS GUI' for more details.
- Firmware file: Please refer to section 1 'Firmware' for more details.

3.1.3 Data Formats

RMS GUI parameters will adhere to the data formats as listed in table 2.

The column, Variable Type follows the standard computer programming data types:

- Byte (char): an 8-bit value ranging from 0 – 255 if unsigned and -128 – 127 if signed.
- Integer (int): a 16-bit value ranging from 0 – 65535 if unsigned and -32768 – 32767 if signed.
- Long Integer (long): a 32-bit value ranging from $-(2^{31}+1)$– 2^{31}.

All EEPROM data is broadcast with a multiplication factor. In order to get the actual value, divide it by the value in the column 'Multiplier' (may also be referred to as 'Prescalar').

Format	Variable Type	Range	Unit	Multiplier
Temperature	Signed Integer	± 3000.0	° C	10
Low Voltage	Signed Integer	± 300.00	Volts	100
High Voltage	Signed Integer	± 3000.0	Volts	10
Torque	Signed Integer	± 3000.0	N.m.	10
Current	Signed Integer	± 3000.0	Amps	10
Angle	Signed Integer	0 – 359.9	Degrees	10
Angular Velocity	Signed Integer	±30000	RPM	N.A.
Boolean	Unsigned Byte	0 OR 1	Binary	N.A.
Frequency	Signed Integer	± 3000.0	Hz	10
Power	Signed Integer	± 3000.0	kW	10
Flux	Signed Integer	0 to 30.000	Webers	1000
Proportional Gain	Unsigned Integer	0 – 655.00 OR 0 – 6.5535	N.A.	100 OR 10000
Integral Gain	Unsigned Integer	0 – 6.5535	N.A.	10000
Derivative Gain	Unsigned Integer	0 – 655.35	N.A.	100
Low-pass Filter Gain	Unsigned Integer	0 – 6.5535	N.A.	10000
Time	Unsigned Long Integer OR Unsigned Integer	See Parameter Description	See Parameter Description	See Parameter Description
Per-unit Value	See Parameter Description	See Parameter Description	See Parameter Description	See Parameter Description

Table 2: Data Formats

3.2 EEPROM Parameters Guide

RMS GUI is a Windows application developed by BorgWarner Portland (formerly Cascadia Motion). This application communicates over a RS232 port. The primary purpose of this application is to be able to monitor a specific set of parameters in real time. However, the application also provides the ability to program certain EEPROM parameters. The set of EEPROM parameters need to be modified based on each motor and other system set up by the customer. EEPROM parameters must be programmed correctly before the controller is operated.

This section provides the user with a process of updating EEPROM parameters using the GUI application.

3.2.1 Programming Steps

(a) Start the GUI. Firmware date code and firmware version on the title of the GUI window. The RMS GUI application also displays the COM port information in the title of the window.

(b) Click on the EEPROM View tab (labeled 'Tab 2' in figure 5). This will display all EEPROM parameters that can be programmed by the user.

(c) In order to change any value, click on the value under the VALUE column. Enter a new value and then click ENTER key on your keyboard.

(d) When all values are changed, click on the Program EEPROM button (labeled 'Button 3' in figure 5).

(e) A status message will confirm whether the programming was successful or not. Follow the on-screen instructions.

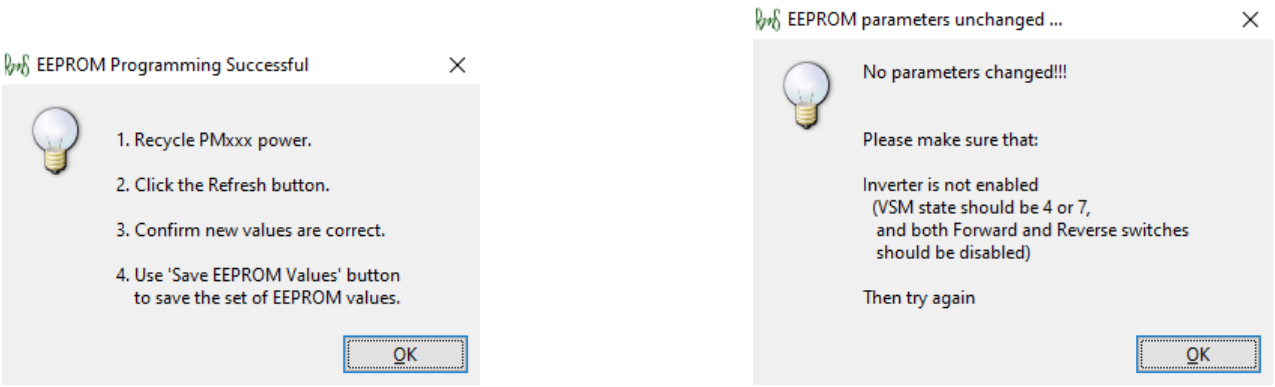


Figure 4: EEPROM Pop-ups

3.2.2 Saving EEPROM Values

EEPROM values can be saved by using the Save button (labeled 'Button 4' in 5). You will be prompted for a filename to save the data to. After selecting the file you will be prompted to press "OK" to start the download.

3.2.3 Uploading EEPROM Values

You can also load a predefined set of values by using the Load EEPROM Values button (labeled 'Button 2' in figure 5).

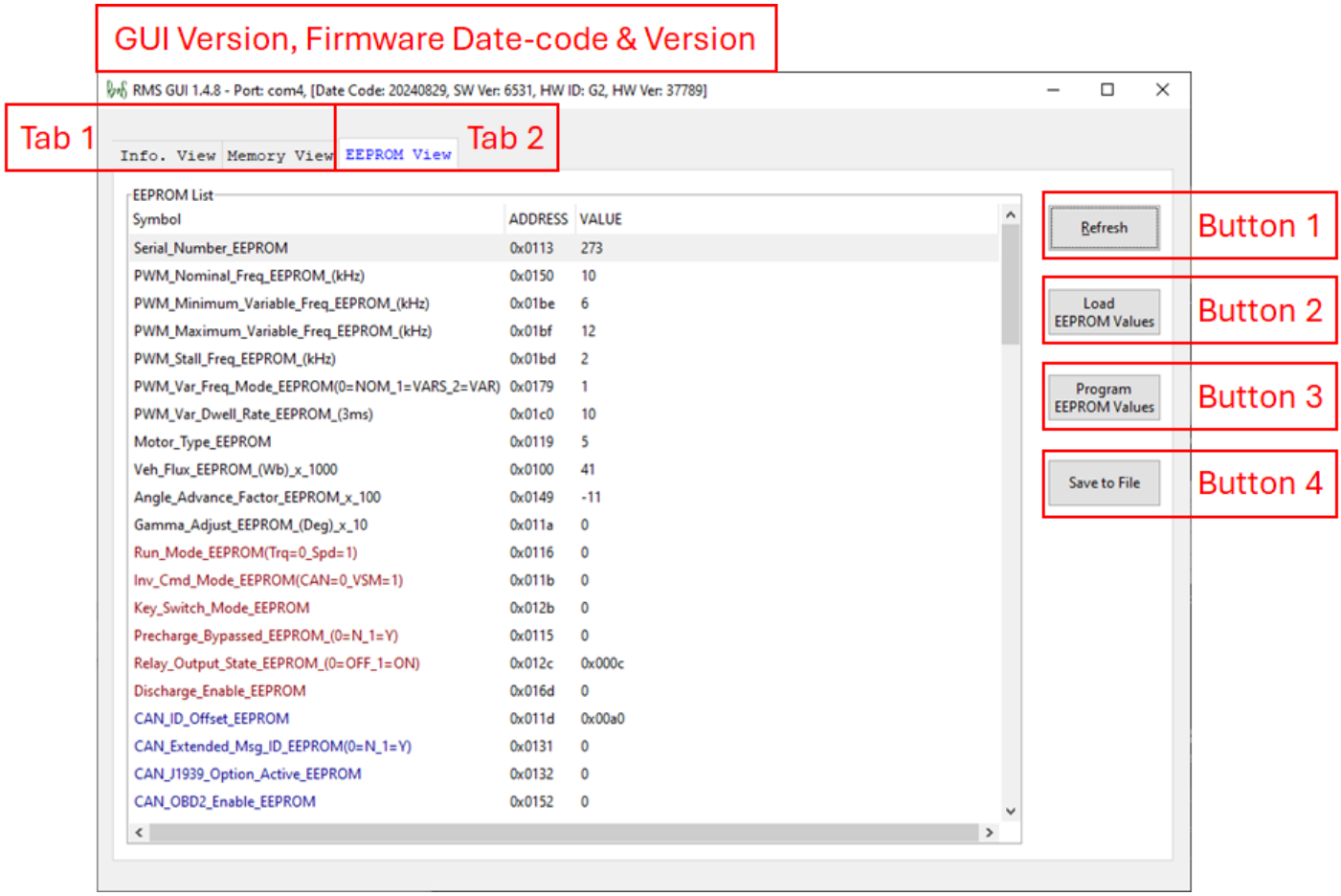


Figure 5: Uploading EEPROM Values

3.2.4 EEPROM Parameter Setup (via GUI EEPROM View)

There are a number of internal parameters (may be considered as "calibrations") that must be set in the controller before it is ready to operate a vehicle. All of these values must be adjusted to suit the vehicle and motor you are using. These adjustments are part of personalizing the drivability and vehicle dynamics to suit the final application of the vehicle.

EEPROM Parameter setup is accomplished using the RMS GUI or by using the Parameter message of CAN. Refer to section 3.2, 'EEPROM Parameters Guide' for more information on how to update and program these parameters in non-volatile memory.

Refer to the following appendices for different categories of EEPROM parameters (each appendix is hyper-linked, press CTRL-CLICK to go to a specific table):

A.1: Motor Configuration Parameters

A.2: System Configuration Parameters

A.3: Continuous Variable PWM Operation

A.4: CAN Configuration Parameters

A.5: Current Configuration Parameters

A.6: Voltage & Flux Parameters

A.7: Temperature Parameters

A.8: Accelerator & Torque Parameters

A.9: Speed Configuration Parameters

A.11: Shudder Compensation Parameters

A.12: Brake Parameters (applies to VSM Mode)

3.3 Monitored Parameters View (via GUI Memory View)

The GUI provides the ability to monitor several operation parameters of the controller. It is also helpful for checking connections to the controller. Items can be added or removed from the Memory Window to the Watch window to view the parameter.

Refer to the following appendix for a complete list of parameters that can be monitored through RMS GUI (each appendix is hyper-linked, press CTRL-CLICK to go to a specific table):

GUI Watch Parameters

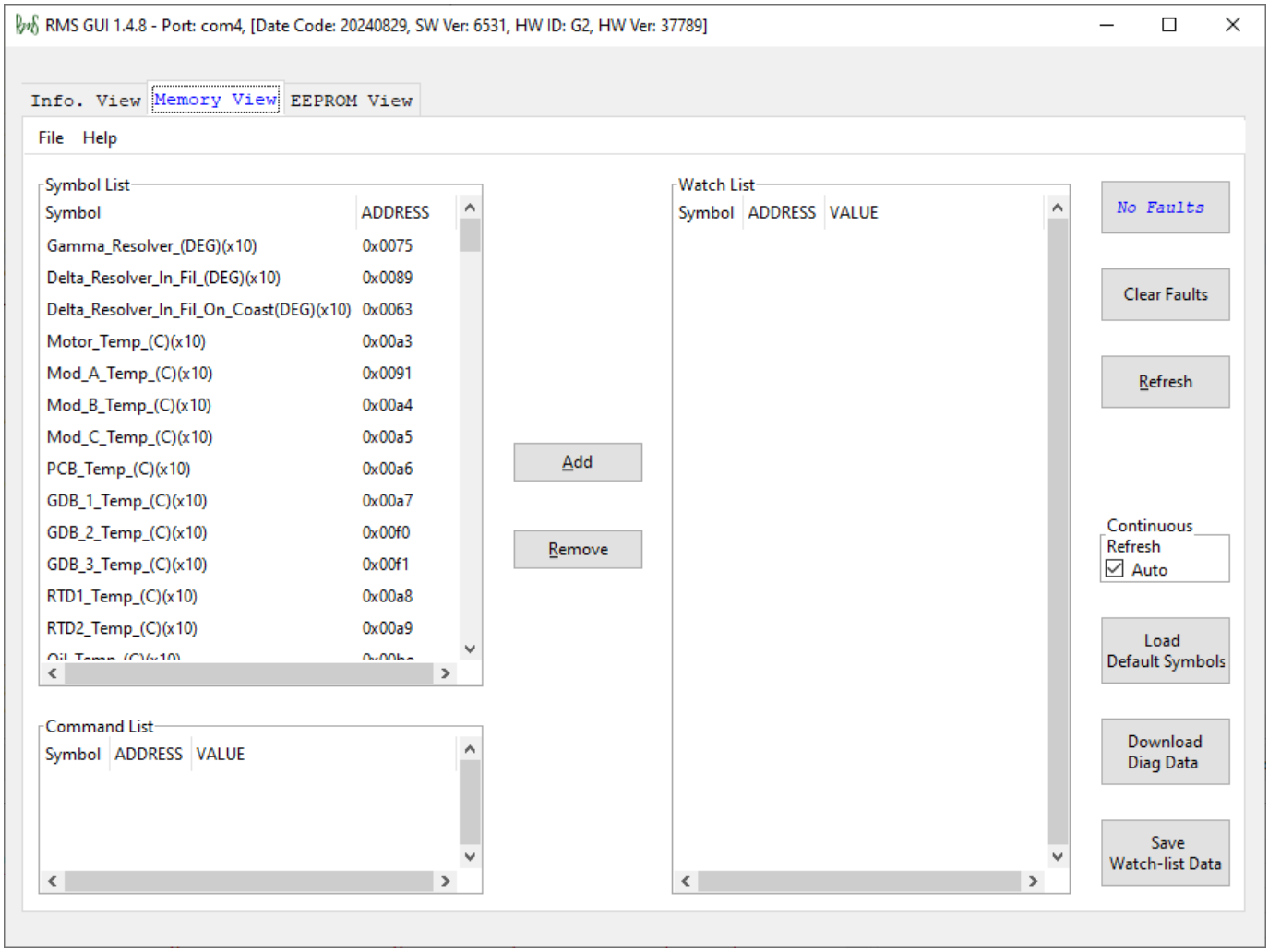


Figure 6: RMS GUI

No Faults/Check Faults button: This button allows the user to check the fault status when the 'Auto' box is check for 'Continuous Refresh' or by clicking on this button. 'No Faults' status in blue indicates that there are no faults currently present. 'Check Faults' status indicates the presence of one or more faults. To check which faults are present, click on this button.

Clear Faults button: This button allows the user to clear all faults with the exception of a few

mentioned in Appendix D.2 (table of Run Faults).

Download Diagnostic Data button: This button allows the user to download Diagnostic Data. Please refer to section 6.8, 'Download Diagnostic Data', for details. Please note that on some PCs it has been found that the RMS GUI will appear like it has stopped responding during the middle of downloading of the Diag Data. Please wait several minutes for the RMS GUI to attempt to complete the download.

Load Default Symbols button: This button allows the user to load the default symbols file for the firmware in the PM unit.

3.4 Switching Back To DAQ Mode

Once in the GUI mode, the user has the option to switch back to the serial data acquisition mode. However, it requires the RMS GUI application to be completely shut down. In other words, the GUI application must release the serial port.

Once the serial port is released, another terminal application such as Realterm can be started. Open the serial port and click anywhere in the window where the serial data appears. Press '+' and then <Enter>. The serial broadcast data should start to appear again.

Realterm, in particular, also has an option to send out the ASCII characters as shown in figure 7. You can enter '+' in the first box and check the +CR and +LF options for carriage return and linefeed respectively. Then press the "Send ASCII" button. The serial broadcast data should start to appear again.

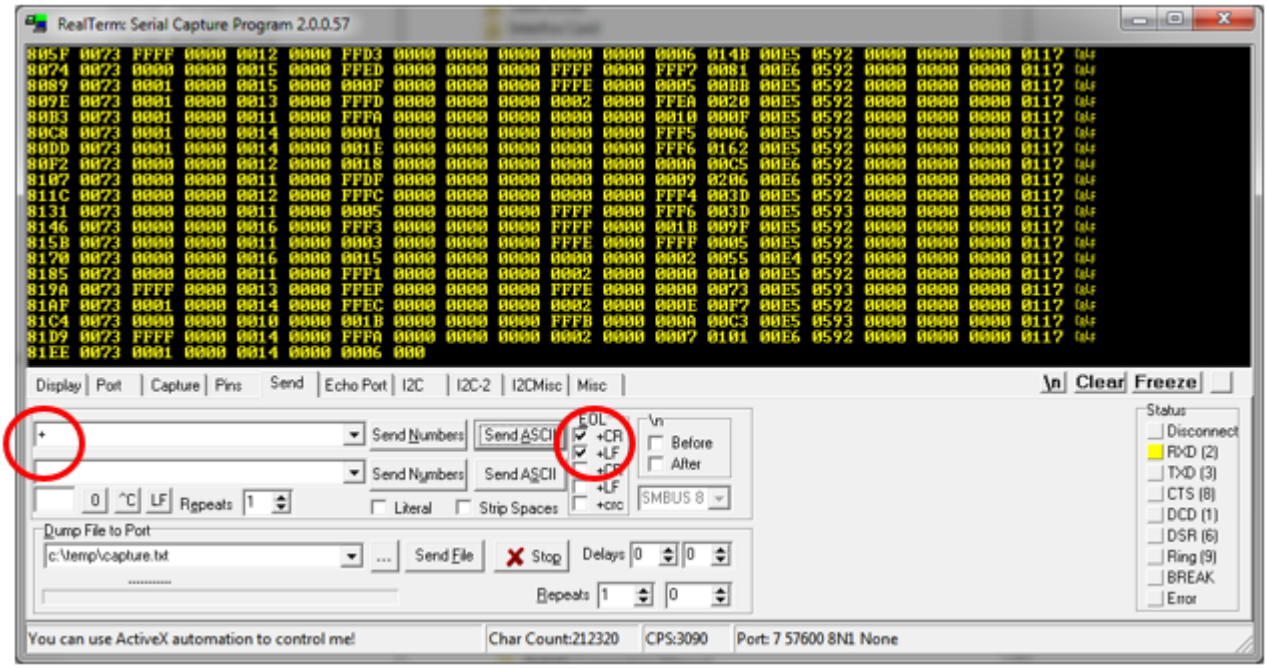


Figure 7: Realterm

4 Calibration Processes

Before the inverter can be used successfully, it is very important to make sure that it is calibrated properly. There are several calibrations that are performed before each unit is shipped to the customer. However, some of these calibrations depend on the specific environment in which the unit is used.

User Manuals for following calibration processes are provided to customers. The calibrations can be performed as many times as needed.

Table 3: Calibration Process

Calibration Process	**User Manual (PDF format)**	**Factory Calibrated?**
Current Offset	BorgWarner Portland Calibration Manual	No, not necessary
DC Voltage	BorgWarner Portland Calibration Manual (factory calibrated thus not normally needed)	Yes
Hall Sensor Encoder	BorgWarner Portland Calibration Manual (not normally needed)	No
SIN/COS Encoder	See BorgWarner Portland Calibration Manual (only necessary with certain motors that have a sin/cos encoder)	No
Resolver	Section 4.1, 'Resolver Calibration' (this process is necessary for all motors that use a resolver or SIN/COS encoder)	No
RTD	BorgWarner Portland Calibration Manual (factory calibrated thus not normally needed)	Yes

4.1 Resolver Calibration

4.1.1 Overview

Resolver calibration is implemented by an adjusting an angle offset (Gamma Adjust) between the resolver and the magnetic field of the motor (rotor position).

Important Note: It is not necessary for the high-voltage DC to be connected to the inverter. However, spinning the motor will generate a DC voltage in the inverter, even if the inverter is off. This voltage is dangerous, take proper precautions. If a high-voltage DC is connected to the inverter make sure that it is high enough in value that the voltage generated by the motor will be less than this battery voltage.

In addition, the Motor Type EEPROM parameter must already be set for the motor that you are using. You can usually find the Motor Type number for the motor in the motor specific manual provided by BorgWarner Portland. If you do not know the number for your motor, please contact BorgWarner Portland for more information.

4.1.2 Verifying Resolver and Motor Direction

Operation of the resolver should be checked prior to calibration. The goal of this section is to verify that the resolver is working correctly.

The following parameters associated with the Resolver Direction can be accessed by the RMS GUI software or CAN application:

- `Gamma_Resolver_(Deg)_x_10`
- `Feedback_Speed_(RPM)`
- `Voltage_Feedback_Speed_(RPM)`
- `Sin_corr_(V)_x_100`
- `Cos_corr_(V)_x_100`

`Gamma_Resolver_(Deg)_x_10` will increase as the motor shaft is slowly turned by hand in the forward direction. When the value reaches 3600 it will reset back to 0. Verify that when the shaft is turned in the forward direction than the value of `Gamma_Resolver_(Deg)_x_10` increases. The values of `Sin_corr_(V)_x_100` and `Cos_corr_(V)_x_100` should also adjust as if they were running through a circle. Imagine `Cos_corr_(V)_x_100` on the x-axis and `Sin_corr_(V)_x_100` on the y-axis (for PM Gen 3 family only).

If the value of `Gamma_Resolver_(Deg)_x_10` decreases or does not change when turning forward then there is wiring issue with the resolver.

When spinning the motor the Feedback Speed parameter should indicate either a positive number for forward rotation or a negative number for reverse rotation.

The next step is to verify the motor direction as determined by the motor phase connections to the inverter. The only way to do this is to spin the motor from an external source at a speed sufficient for the controller to measure the back EMF of the motor. Monitor the Voltage Feedback Speed parameter to determine the direction of rotation as being calculated by the controller.

The direction of rotation as measured by Voltage Feedback Speed must match the direction as determined by the resolver. The magnitude of the Voltage Feedback Speed should match the Feedback Speed. If the magnitudes do not match there could be an error in the setup of the resolver (number of resolver poles does not match the motor type setup).

Note that if the motor speed is not high enough the back EMF will be too small for a proper measurement. Typically a motor speed of about 25% the base speed of the motor should be enough to get a stable measurement.

4.1.3 Resolver Angle Offset Adjustment (Gamma Adjust)

The resolver angle offset setting is the most important part of the resolver calibration process. The motor controller should not be operated in speed mode until the resolver calibration process has been completed.

The following parameters associated with the Gamma Adjust can be accessed by the RMS GUI software or CAN application:

- `Gamma_Adjust_EEPROM_(Deg)_x_10`
- `Gamma_Adjust_(Deg)_x_10`
- `Delta_Resolver_In_Fil_(Deg)_x_10`
- `Delta_Resolver_In_Fil_On_Coast`
- `Feedback_Speed_(RPM)`

The next step is to align the magnetic field of the motor with the resolver. The resolver rotational position on the shaft is not necessarily always in the same place due to manufacturing tolerances, nor is it always aligned with the magnetic field.

To adjust this angle it is necessary for the motor to be connected to the inverter.

To run this calibration it is necessary to spin the motor at a speed that provides sufficient back EMF for the controller to measure. Generally 1000 rpm is sufficient. However for some motors this speed will be too low. For some motors the speed will be too high. A good rule of thumb would be about 25% to 33% of the no-load base speed of the motor.

The motor must either be spinning from an external mechanism or it can be coasting down. In either case the PWM must be OFF (motor not enabled to run) when perform the calibration readings.

If it is necessary to try and operate the motor to allow it to coast down it is recommended that torque mode be used for control and not speed mode. Speed mode can result in high motor currents in an uncalibrated motor.

The controller is not very sensitive to the exact gamma adjust value when operated at no-load. Run the motor with as little load as possible. Use a small torque command to bring the motor speed up to some value above the calibration speed. Then command the inverter off. As the motor coasts down the values can be read as described below.

The internal voltage sensors in the inverter are used measure the back EMF of the motor and thus determine the alignment. The internal sensors that measure back EMF can only measure the voltage when the motor is NOT enabled. Thus it is necessary to have the motor be disabled when monitoring the back EMF.

Put the following variables into the watch window of the RMS GUI or CAN application:

- `Gamma_Adjust_(Deg)_x_10`
- `Delta_Resolver_In_Fil_(Deg)_x_10`
- `Feedback_Speed_(RPM)`

Click the continuous refresh button so that the values will continuously update.

The `Delta_Resolver_In_Fil_(Deg)_x_10` will now display an angle (3600 = 360 degrees). The goal is to get this angle to be 900 (90 degrees) if the motor is running in forward direction or -900 (-90 degrees) if the motor is running in reverse direction. If the resolver is properly connected and the motor is spinning at a sufficient speed the displayed angle should remain relatively constant (less than ±2 degrees) as the motor is spun. If the value is constantly changing or staying at a value near zero degrees then it is likely that the motor phasing is not correct. Try swapping two of the motor leads.

While the motor is spinning in forward direction at the selected calibration speed (NOT enabled) record the angle `Delta_Resolver_In_Fil_(Deg)_x_10`. Next determine the angle represented by `Delta_Resolver_In_Fil_(Deg)_x_10`. Divide the value by 10 to convert it to degrees. Now determine the amount of adjustment necessary to make `Delta_Resolver_In_Fil_(Deg)_x_10` be plus or minus 90 degrees as determined by direction of rotation. Increasing `Gamma_Adjust_(Deg)_x_10` will decrease `Delta_Resolver_In_Fil_(Deg)_x_10`.

Repeat the process until `Delta_Resolver_In_Fil_(Deg)_x_10` = 90 degrees (900) for positive rotation, or `Delta_Resolver_In_Fil_(Deg)_x_10` = -90 degrees (-900) for negative rotation. The goal would be to adjust the `Gamma_Adjust_(Deg)_x_10` parameter until `Delta_Resolver_In_Fil_(Deg)_x_10` reads within ± 0.7 degrees. Once this goal is achieved, program the value in `Gamma_Adjust_(Deg)_x_10` into its EEPROM equivalent, `Gamma_Adjust_EEPROM_(Deg)_x_10` to save it permanently as a calibration parameter.

`Gamma_Adjust_(Deg)_x_10` can be either positive or negative as needed.

Example:

1. While spinning the motor at 1000 RPM (when it is NOT enabled), the `Delta_Resolver_In_Fil_(Deg)_x_10` parameter reads a value of 828.
2. Convert the value to degrees by dividing by 10. 828/10 = 82.8 degrees.
3. To get to 90 degrees we need to change by $90 - 82.8 = 7.2$ degrees.
4. The value of `Gamma_Adjust_EEPROM_(Deg)_x_10` while running the test was 2.9 degrees. So to increase `Delta_Resolver_In_Fil_(Deg)_x_10` we need to decrease `Gamma_Adjust_EEPROM_(Deg)_x_10`. So we calculate the adjustment as $2.9 degrees - 7.2 degrees = -4.3$ degrees.

5. Enter the new value into `Gamma_Adjust_(Deg)_x_10` and repeat the process to verify that `Delta_Resolver_In_Fil_(Deg)_x_10` now reads 900.
6. Program the new calibration value for `Gamma_Adjust_EEPROM_(Deg)_x_10` as -43. Reset the power to the controller. Repeat the test to confirm the change.

Helpful hints:

- If it is not possible to spin the motor from an external source, then it is possible to spin the motor from the controller by guessing at various values of gamma adjust that make the motor spin in the desired direction. Make changes in gamma adjust (large changes like 45 degrees are acceptable). An unloaded should not take much torque to get the motor to spin. Once the motor is spinning allow the speed to increase above the 25% to 33% target and then disable the inverter. Watch delta resolver value while the motor is coasting down.
- If using the method of coasting down after the inverter has been enabled, then it can be quite helpful to record the CAN data be sent by the inverter. It is easier to see the exact value of delta_resolver during the coast down.
- A motor that will not spin with any value of gamma adjust likely means that the resolver direction doesn't match the motor phase direction. Either swap both SIN with both COS or swap two of the motor phases.

4.1.4 Motor Core Delta Resolver Speeds

The speed at which `Delta_Resolver_In_Fil_(Deg)_x_10` will be displayed isa function of the motor poles. For the common BorgWarner Portland catalog motors this speed is listed below.

Table 4: Motor Core Delta Resolver Speeds

Motor Core / CM Common Name GUI Parameter	**Delta_Resolver_In_Fil_On_Coast_Speed [RPM]**
SS250-115-DOM iDM-190DX-D iM-225DX iM-375DZ-D, iM-375SiC-D, iDM-375SiC-D	1050
SS250-115-SOM iM-225DZ-S	1050
SS410-150-DOM iM-425DZ-D, iM-425SiC-D	525

The coast value is calculated via the motor poles and the common PU frequency in the firmware. The equation is listed below:

`Delta_Resolver_In_Fil_On_Coast` Speed [RPM] = 10500 / Motor Poles

i.e. For a 10 pole motor like the SS250-115-DOM,

`Delta_Resolver_In_Fil_On_Coast` Speed [RPM] = 10500/10 = 1050 [RPM]

5 Vehicle State Machine

The inverter software has an internal state machine that steps through a series of actions at startup, shutdown, and generally whenever operation "transitions" from one mode or state to another. Note that this state machine is used in both CAN mode and VSM operation modes.

The particular state that the inverter is in can be tracked via the RMS GUI parameter `VSM_State` or via CAN signal `INV_VSM_State`. Data for both of these signals will correspond to the states outlined in table 5.

Table 5: Vehicle State Machine

VSM_State	Name
0	Start State.
1	Pre-charge sequence initial state-Turn on the pre-charge relay.
2	Pre-charge sequence active state-Waiting for capacitor to finish charging.
3	Pre-charge sequence finish state-Completes the final checks before proceeding to Wait State.
4	Wait State-waiting for activation of forward or reverse.
5	Ready state-Activates the inverter state machine to begin energizing the motor.
6	Motor Running State-Normal motor running.
7	Fault State-The controller has faulted.
14	Shutdown in Process-In key switch mode 1, user has turned key switch to off position.
15	Recycle Power State-This indicates that the power to the controller needs to be recycled after EEPROM Programming is complete.

5.1 Start State (VSM_State = 0):

5.1.1 12V Power-up:

When the vehicle is powered up, this is the default state. If the program enable input is held low at power up it will not execute the software in the inverter and will not proceed into the Vehicle State Machine.

5.1.2 Default Initialization:

This is the processor setup and initialization process, including setting all I/O pins to the correct state (in/out, pull-up or pull-down, weak or strong, etc). At this point, the initialization process sets up a default list of parameters with pre-assigned default values.

5.1.3 Load from EEPROM:

This state will load the application parameters to configure the unit for the actual application. This also loads FACTORY CALIBRATIONS from memory, as these are just a class of EEPROM parameters.

5.1.4 Power on Self-Test (POST):

A number of tests are to be performed in this state. Each test will have an associated fault flag.

The following is a list of parameters checked:

Table 6: POST Checks

Test Area	Description
Current sensors	Check current sensors reading to be within a valid range
Accelerator input	Check accelerator input data is within a valid range
PCB Temperature Sensor	Check PC temperature is in valid range
GDB Temperature Sensor	Check gate drive board temperatures in range
Module Temperature Sensors	Check substrate temperatures for module A, module B, and module C in range
5V power	Check internal 5V and external transducer power in range
12V power	Check 12V power in range
2.5V power	Check internal 2.5V reference voltage in range
1.5V power	Check internal 1.5V reference voltage in range
HW Faults (Saturation and over current)	If exist, attempt to clear faults and then report

If a power-on self-test fault occurs it will blink the fault indicator followed by two quick blinks to differentiate POST faults from RUN faults. The number of blinks gives a general indication of the particular fault.

A particular fault code can be found by clicking on the “Check Faults” button on the “Memory View” page of RMS GUI. For a complete list of power-on self-test faults please refer to section D.1, ‘POST Faults’.

5.2 Pre-Charge Sequence:

5.2.1 Pre-charge initialization (VSM_State=1)

This state declared VDC Out-of-range high fault if DC voltage is above the software over-voltage threshold. The value of software over-voltage threshold is not adjustable the user and is set based on the particular type of inverter being used.

If DC voltage is below the software over-voltage threshold, Pre-charge output is activated. State machine goes to Pre-charge Active State.

5.2.2 Pre-charge Active (VSM_State=2)

This state controls the charging of the capacitors internal to the controllers. If the rate of charge stays within range, Main output is activated and Pre-charge output is deactivated. During the pre-charge process:

If DC voltage exceeds software over-voltage threshold, VDC Out-of-range high fault is declared.

After 3 seconds that is, the maximum pre-charge time,

1. If DC voltage is less than the value of EEPROM parameter, DC Under-voltage threshold VDC Out-of-range low fault is declared.
2. If DC voltage is still charging, pre-charge timeout fault is declared.

5.2.3 Pre-charge Complete (VSM_State=3)

This state checks if the capacitor charge is stable, that is, it is not over-charged or under-charged, or there is no quick change in voltage since the pre-charge output was deactivated. If any of the conditions is true, a relevant fault is declared.

5.3 Wait State (VSM_State=4)

This state checks for the Key Switch Mode. Based on that value, the inverter can be powered to run the motor as follows:

5.3.1 Key Switch Mode 0

This mode allows for a simple on/off ignition switch functionality. To power up the PM unit, turn the ignition to ON position. This state then checks to see that the brake switch is active and only one of /FORWARD and /REVERSE switches is active. If both switches, /FORWARD and /REVERSE, are active, the state shall declare a FWD_RVS_INVALID_STATE_FAULT. If a correct direction and the brake are active then the motor will be enabled.

5.3.2 Key Switch Mode 1

This mode allows for traditional ignition switch functionality. To power up the PM unit, turn the ignition to ON position. This state then checks to see that the brake switch has been active and start

signal pulse has been received. While keeping the brakes on, only one of /FORWARD and /REVERSE switches needs to be activated. If both switches, /FORWARD and /REVERSE, are active, the state shall declare a FWD_RVS_INVALID_STATE_FAULT.

5.4 Ready State (VSM_State=5)

The READY state shall send out the Enable Inverter Command and wait for Inverter Ready Flag to be set. The Inverter Ready Flag will be set if the inverter successfully performs a series of actions necessary to stat the motor. If inverter does not enable the motor within a specific amount of time, the state shall declare an inverter state timeout fault.

This state automatically transitions to the next state if there are not faults.

The following table lists several inverter states:

Table 7: Inverter State Machine

VSM_State	Name
0	Precharge, power-up state
1	Stop-Inverter is not running and is in “STOP” state.
2	Open Loop State-for testing purposes
3	Closed Loop state-normal state
4	Start Time Delay-small delay before starting the inverter
5	Current Sensor Test-flux ramp and flux regulators enabled
6	Closed Loop Torque-torque regulator is enabled
7	Torque Ramp-start torque ramp
8	Idle Run-inverter running normally
9	Idle Stop-inverter is stopped
10	Ramp Off Torque-ramps down the torque command
11	Ramp Off Flux-ramps down the flux command
12	All Ramps Off-shutoff inverter
15	Default-Stop state

5.5 Motor Running State (VSM_State=6)

This is the normal motor running operation of the vehicle state machine. While running the drive can be switched from torque command to speed command mode, and may be exercised within the full operating envelope of the machine / drive combination.

5.6 Fault State (VSM_State=7)

If a fault occurs either during power-On self-test, or while the drive is running, the drive will go to the fault state.

If the drive has a fault during the running state a fault code will be set and the fault indicator will begin blinking. At any given time, the fault indicator will blink only one fault.

A particular fault code can be found by clicking on the "Check Faults" button on the "Memory View" page of RMS GUI. For a complete list of run faults please refer to section D.2, 'Run Faults'.

5.6.1 Fault Priority:

Fault indicator will blink faults in the following priority:

POST Faults (Higher Priority)

RUN Faults (Lower Priority)

POST faults are followed by two quick blinks to distinguish from RUN faults. For each type of fault (POST or RUN), the highest priority of a fault is based on the number of blinks. The fault with 1 blink is the highest priority and the fault with the highest number of blinks is the lowest priority fault. The fault blinking will occur such that if the highest priority fault goes away, the lower priority fault will start blinking and this pattern will continue till all faults are removed.

5.6.2 Clear Faults Command:

Once a fault is acknowledged, it can be cleared using the Clear Faults Command from the GUI. In order to clear a fault, set the Clear Faults Command to 0.

This command clears all active faults including POST Faults. The only exception is the POST Fault, EEPROM Update Required. This fault is set after programming a new firmware in the PM controller. The purpose of this fault is to have the user accept all previous EEPROM parameters and update the new ones. If there are no EEPROM parameters to update, user should still enter the Access Code and Program EEPROM Command to accept all EEPROM parameters. Please refer to section 3.2 EEPROM Parameters Guide for more details on how to program EEPROM parameters.

In CAN mode, before sending out the Clear Faults Command, make sure that the inverter is disabled. If inverter is enabled and the command is sent out, the motor may start running based on the mode and commanded Torque/Speed.

5.7 Shutdown in Process State (VSM_State=14)

This state indicates that the inverter "Shutdown in Process". In key switch mode 1, user has turned key switch to off position by holding the ignition input low.

5.8 Recycle Power State (VSM_State=15)

This state indicates that the EEPROM Programming has been successfully completed. For new EEPROM values to take effect, the controller must be re-powered.

6 Functionalities

6.1 Torque Capability

Each execution of the inverter's 3ms loop, a new torque capability calculation is run for both motoring and regeneration cases. The torque capability calculation first considers the maximum torque allowed for the motor type as well as the maximum torque allowed by user specification in the EEPROM. Next, the active current limit for the inverter, which is influenced by active PWM frequency, bus voltage, stall model status, hot spot model temperature, coolant temperature and motor temperature, is used to back-lookup the defined torque set-point from an IPM motor control table. This is done for both motoring and regeneration cases.

The continuous torque capability calculations are available to be read via CAN broadcast message, 0xB1, see table 71.

6.1.1 Use Cases

On BorgWarner Portland (formerly Cascadia Motion) catalog products (iM-225, iM-375, iM-425, etc.) the torque capability feature limits the commanded torque from a VCU to facilitate additional Iq/Id command accuracy. Note that there is a slight offset added to the torque capability command limited this way, 2% of the max torque allowed via EEPROM settings `Motor_Torque_Limit_EEPROM_(Nm)_x_10` and
`Regen_Torque_Limit_EEPROM_(Nm)_x_10`, which provide a bit of calculation error buffer. See figure 8 for an example of this.

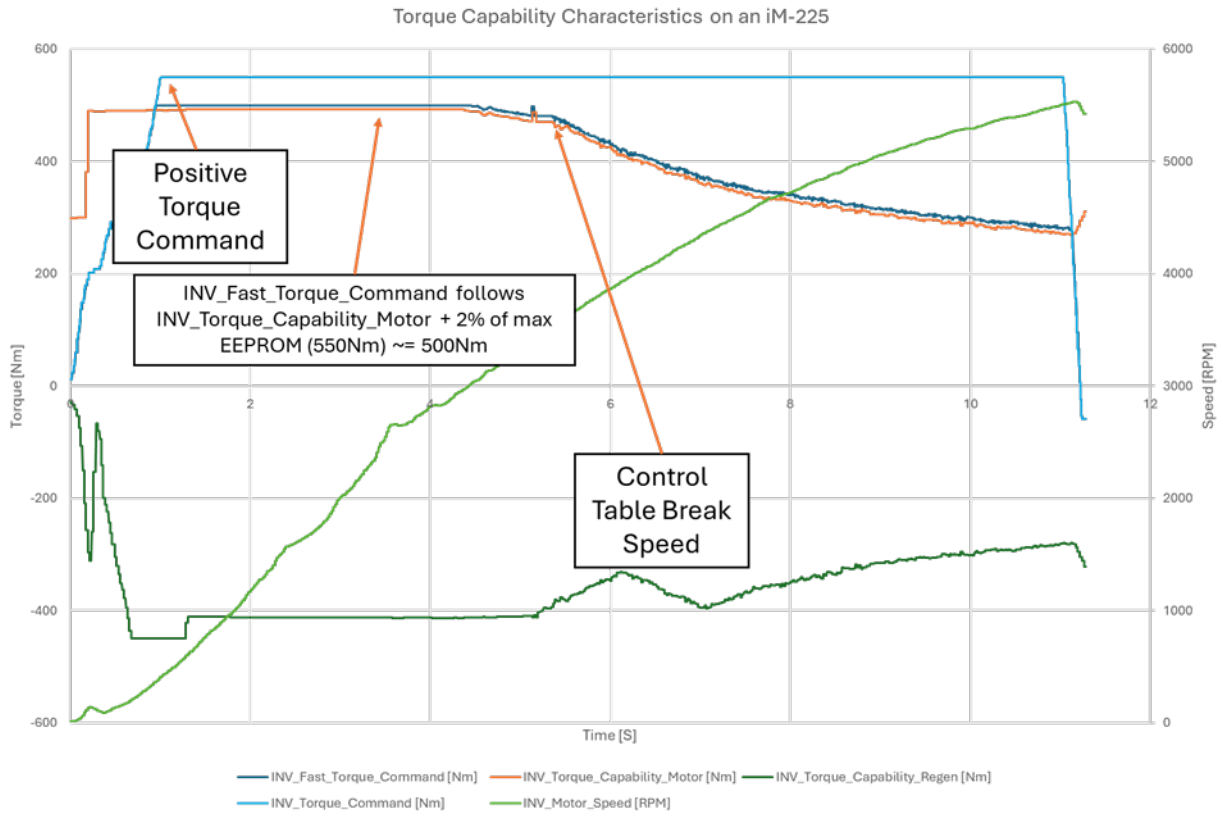


Figure 8: Torque Capability Example

6.2 Field Weakening

The Gen 5 and CMxxx inverters support multiple forms of Field Weakening for higher speed operation of motors. Field Weakening refers to the application/modification of the control current to maintain the motor voltage generated via the Back EMF (BEMF) when operating past the connected motor corner point. Thus this applies more specifically to all permanent magnet motors connected to a Gen 5or CMxxx inverter.Each Motor Types as defined int he code has a Field Weakening control strategy enabled by default, and in general cannot be changed. However, the catalog motors offered by BorgWarner Portland (formerly Cascadia Motion) may have the option to try one of two different control strategies. The three types of Field Weakening control are defined in the table below.

Field Weakening Type	General Control Description
Surface PM Motor Field Weakening	Applicable for surface permanent magnet motors and Motor Type 65, this methodology applies negative D axis current to maintain the set max modulation index.
Internal PM Motor Field Weakening	The default field weakening procedure for all internal permanent magnet motors that use Id and Iq control tables. This methodology reduces Q axis current to maintain the set max modulation index and assumes the tables are properly applying the correct D axis current
Internal PM Motor DC Link Field Weakening	A new feature for the catalog BorgWarner Portland integrated units which modifies the effective DC voltage used in the internal Id and Iq control tables to maintain the set max modulation index. This methodology should hold torque better when field weakening is required. This will be default for all catalog BorgWarner Portland integrated units moving forward and will be offered as a feature via EEPROM enable for the existing BorgWarner Portland catalog products.

The catalog integrated unit default Field Weakening method and further options are defined the table below:

Catalog Integrated Unit	Motor Type	Default Field Weakening Method	Options
iM-225DX-D / DW	5, 47	Internal PM Motor Field Weakening	Internal PM Motor DC Link Field Weakening available via `DCLink_FW_Reg_Use_EEPROM`.
iM-225DZ-S / SW	24, 22	Internal PM Motor Field Weakening	Internal PM Motor DC Link Field Weakening available via `DCLink_FW_Reg_Use_EEPROM`.
iM-375DZ-D / SiC-D	10	Internal PM Motor Field Weakening	Internal PM Motor DC Link Field Weakening available via `DCLink_FW_Reg_Use_EEPROM`.
iDM-190SiC-D / 375SiC-D	48, 10	Internal PM Motor Field Weakening	Internal PM Motor DC Link Field Weakening available via `DCLink_FW_Reg_Use_EEPROM`.
iM-425DZ-D / SiC-D	243	Internal PM Motor Field Weakening	Internal PM Motor DC Link Field Weakening available via `DCLink_FW_Reg_Use_EEPROM`.
iM-125DX-DW	289	Internal PM Motor DC Link Field Weakening	NA

To enable the DC Link Field Weakening change the EEPROM `DCLink_FW_Reg_Use_EEPROM` to a 1. The Internal PM Motor DC Link Field Weakening Regulator is not available for non-catalog motor types.

6.3 Stall Burst

To improve functionality at stall and low speed, a thermal model has been implemented to allow for real time estimation of inverter switch junction temperature. This model allows for higher bursts of current from stall in a wide array of conditions while continuing to protect the inverter for long term use.

This methodology is implemented in the form of an inverter switch Junction Temperature model fed into a PI regulator targeting a max switch junction temperature. This regulator allows the inverter to have the full max burst current, as found by testing at BorgWarner Portland (formerly Cascadia Motion), then the Junction Temperature Model and PI regulator de-rate that max burst current as necessary to maintain a max junction temperature.

Inverter	Target Max Switch Junction Temperature
CM200DX	210 ° C
CM200DZ	149 ° C
CM350DZ	149 ° C

The Junction Thermal Model takes in: DC voltage in V_{DC}, switching frequency in Hz, motor speed in RPM, motor current in A_{pk}, and coolant temperature in ° C, (with a minimum value of 45 ° C in the model), to predict the hottest inverter switch junction temperature in ° C. Note: the model will change over time, but time is not an inherent input to the system other than the switching period.

The general architecture of control is presented below in Figure 1.

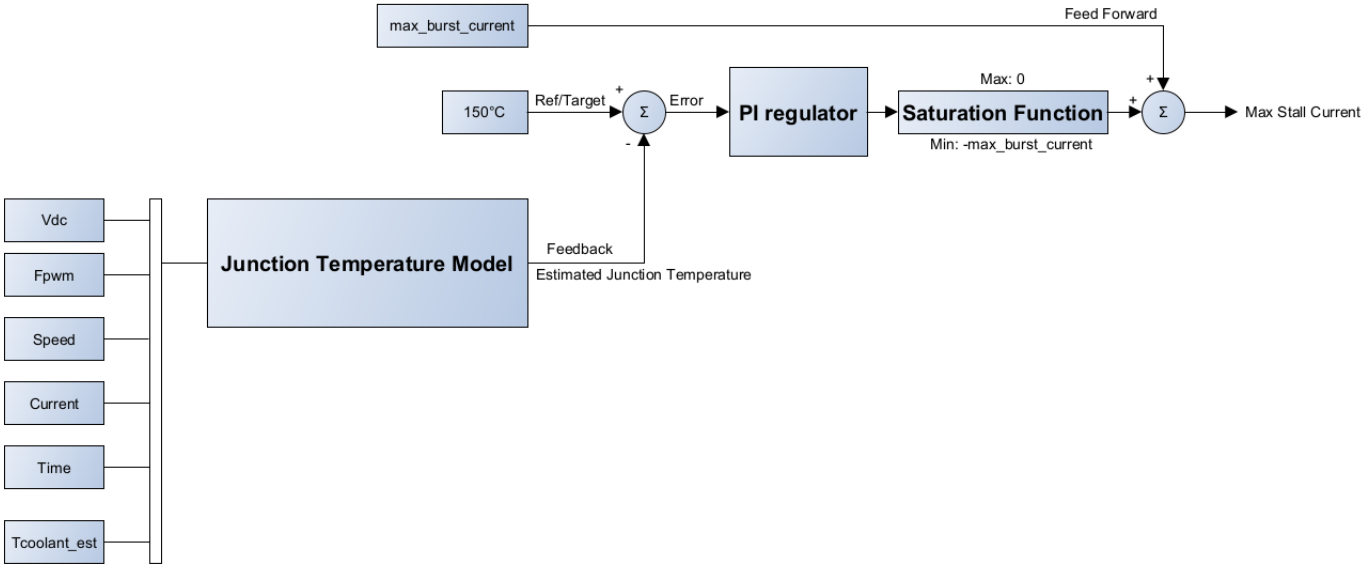


Figure 9: General architecture of Stall Burst Thermal Model.

Through this methodology, compared to a steady state model, more current is allowed at stall for a transient burst, assuming the inverter switch junction temperature is not already at its maximum switch junction temperature.

6.3.1 Stall Burst Thermal Model Settings

There are no required user inputs to activate the Stall Burst Thermal Model outside of using the appropriate inverter.

However, it is highly recommended to limit the `Torque_Rate_Limit_EEPROM_(Nm)_x_10` to no more than 70 Nm/3ms or a setting of 700 for iM225 and iM375 integrated modules. This corresponds to a torque ramp rate of $\tilde{2}$3,000 Nm/s. Rates faster than this may cause stability issues.

There are three new signals relayed over CAN which report: the highest junction temperature by the model in °C, whether Burst Model Mode is in stall or not, and whether the Stall Burst Thermal Model is limiting or not. These messages can be used to monitor the operation of this control mode. These messages, `INV_Stall_Burst_Model_Temp`, `INV_Burst_Model_Mode` (0 = stall and 1 = high speed), and `INV_Limit_Stall_Burst_Model` (1 = On/Limiting and 0 = Off) respectively, are in the latest CAN iM-225 DBC file.

6.3.2 Motor Over-temperature Torque Reduction

During stall operation, excessive heating of the motor end windings may occur. This is accounted for via the Motor Hot Spot model described in section 6.6.

Firmware's 652E through 6531 implemented a less effective strategy to protect against this which may limit the current to zero upon entering stall from a higher motor temperature. It is highly recommended to update the firmware to 6532 or later. If you are unable to update the firmware contact BorgWarner Portland (formerly Cascadia Motion) for information on how to use this derate strategy or avoid it.

6.4 Continuously Variable PWM

The Gen 5 and CMxxx inverters support variable Pulse Width Modulation (PWM) frequency. PWM frequency is the rate at which the inverter switches the DC bus voltage to produce AC voltage waveforms for the motor. It can be advantageous to adjust the PWM frequency depending on motor characteristics and/or inverter operating characteristics. Table 8 lists the supported inverters and respective PWM limitations:

Table 8: Available PWM Settings

Inverter	Allowed Continuous Variable PWM Range [kHz]*	Allowed Stall PWM Range [kHz]*
CM200DX	6 – 24	2 – 24
CM200DZ	6 – 24	2 – 24
CM350DZ	6 – 24	2 – 24
CM350SiC	6 – 16	6 – 16

* The above PWM frequency ranges are the maximum allowed. Not all inverter/motor combinations may work well at all frequencies. Most motor/inverter combinations work well in the 8 – 12 kHz range. If it is desired to operate outside this range consulting with BorgWarner Portland (formerly Cascadia Motion) technical support would be advised.

The Continuous Variable PWM functionality takes in user inputs of a minimum, maximum, and nominal PWM frequency and runs an internal routine to apply the optimal PWM for the given operating point within those constraints.

6.4.1 Continuous Variable PWM Method

The current capability of the inverter depends on several factors:

- Current rating of the power module
- PWM Frequency, higher frequency will reduce maximum current rating
- DC Bus Voltage, higher DC bus voltage will reduce maximum current rating
- Motoring vs Regeneration may affect current rating
- Inverter Coolant Temperature, higher temperature will reduce maximum current rating

The Gen 5 software will adjust the current limit based on these factors. The Continuous Variable PWM functionality takes all these factors into account to apply the optimal PWM that provides the highest current rating while maintaining a PMW frequency adequate for control of the motor.

With Continuous Variable PWM Method the PWM frequency only changes in increments of 1 [kHz]. The only instance that may result in a transition greater than 1 [kHz] is in or out of stall. The minimum time the Continuously Variable PWM Method will stay at a given PWM frequency before transitioning is controlled by the dwell time parameter `PWM_Var_Dwell_Rate_EEPROM_(x3ms)`.

The following sections present a conceptual view of what the Continuous Variable PWM Method would look like if mapped over the full operating space for an iM-375DZ-D. This is purely for reference. The actual map is calculated real time. Stall is not present in these plots, just for visual clarity.

6.4.1.1 Nominal PWM

Continuous Variable PWM Method will prioritize running at the nominal PWM defined by `PWM_Nominal_Freq_EEPROM_(kHz)` unless some other condition is commanded that would require a different PWM to best attain that commanded condition. This can be simply summarized as a starting point of a constant PWM across all operating conditions, as seen in Figure 1.

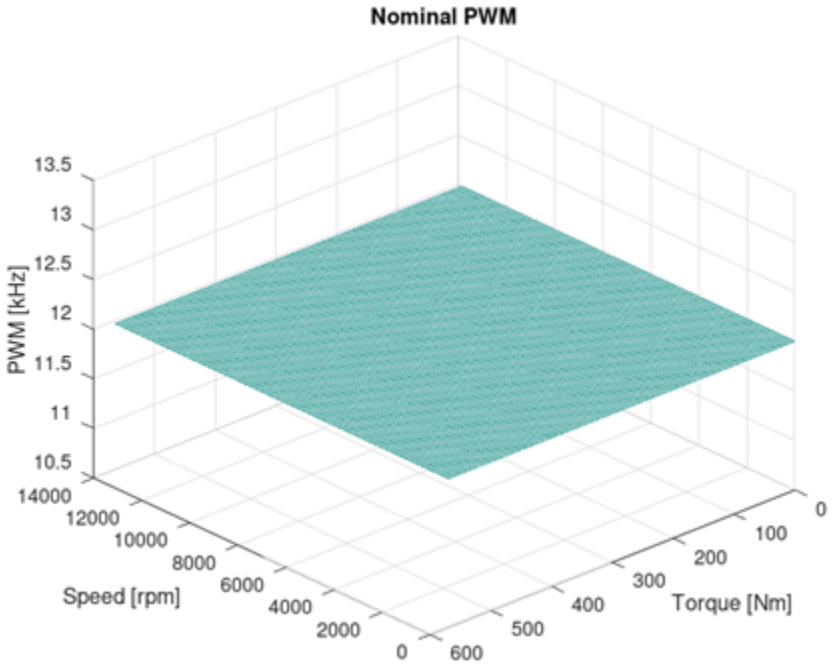


Figure 10: Example Nominal PWM Frequency Application

The nominal PWM frequency acts as the reference point for the desired PWM frequency over a majority of the operating space. This value is generally chosen on criteria of sound, drivability, and noise reduction in the system.

6.4.1.2 Speed Limitations

Electric motors have an inherent minimum PWM frequency to maintain control at a certain speed. The general metric is that at least 15 switching instances should be present per electrical rotation. There are other factors that go into this when determining drivability and control bandwidth, but this simple metric effectively states that at higher speed a higher PWM frequency is required. The Continuous Variable PWM Method will chose the minimum PWM frequency that can run at a given speed without going below the (Minimum PWM Frequency Parameter) or above the (Maximum PWM Frequency

Parameter). Looking purely at speed for the metric of minimum PWM required the Continuous Variable PWM map would look like the example in Figure 2 (for 10 pole motor used in iM-375DZ-D).

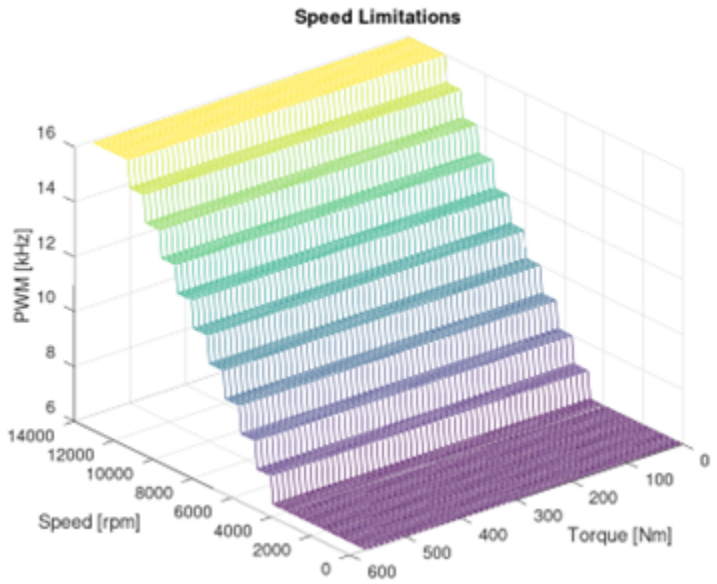


Figure 11: Example of Speed Limitation for Continuous Variable PWM Method

There is some inherent safety built into the selection of the maximum PWM frequency. Selecting a PWM frequency that violates an electric fundamental frequency of less than 10x the EEPROM parameter `Motor_Overspeed_EEPROM_(RPM)` will reset the parameter to the minimum PWM frequency as calculated by:

PWM frequency = `Motor_Overspeed_EEPROM_(RPM)`*Poles/120*15 rounded up.

6.4.1.3 Current Limitations

The most complex relationship with determining the appropriate PWM revolves around the current limitations of the inverter power module. All the factors listed on the beginning of Section 2 change what the capability of the inverter is. Continuous Variable PWM Method considers all these factors to provide a maximum current rating for the inverter. It then optimizes which PWM frequency is the lowest possible frequency that will achieve the intended operating point, but will not chose a PWM above the nominal `PWM_Nominal_Freq_EEPROM_(kHz)` or below the minimum PWM `PWM_Minimum_Variable_Freq_EEPROM_(kHz)`. For example, looking solely at attaining the current request, the Continuous Variable PWM frequency map would look like Figure 3 (for iM-375DZ-D).

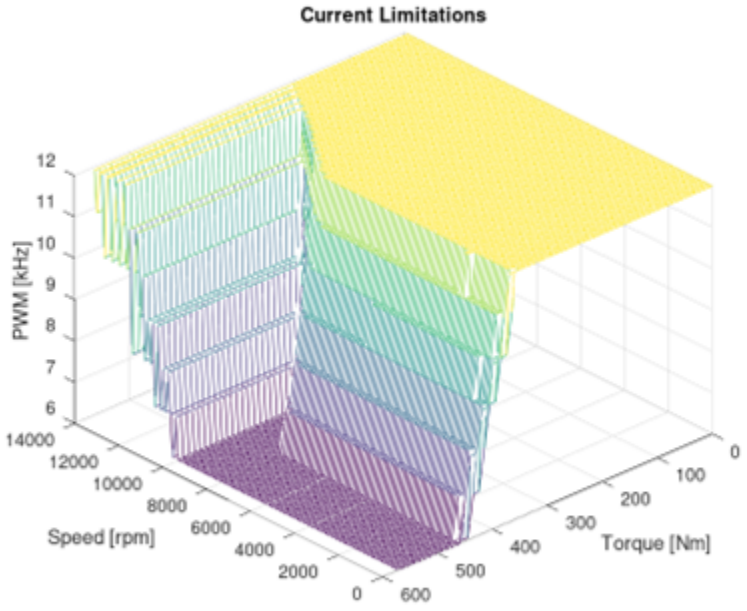


Figure 12: Example of Current Limitation for Continuous Variable PWM Method

There is hysteresis built into the PWM frequency transitions that is not shown in figure 12. The inverter will transition to a lower PWM frequency, to gain more current capability, when it reaches 90% of its current limit at the current PWM frequency. When the current in the inverter drops to 80% of the capability at the higher PWM frequency it will increase its PWM frequency towards the nominal setting.

The current that constitutes the 80% and 90% of capability at a given PWM is largely affected by coolant temperature. Most inverters do not have a coolant temperature sensor, thus the coolant temperature is estimated based on the power module temperature measurement.

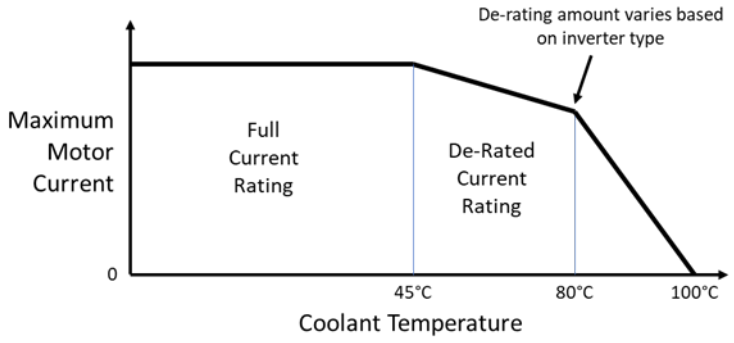


Figure 13: Coolant Temperature De-rating

The above graph shows the current limit of the inverter for a given PWM frequency as the temperature of the coolant is increased. The Coolant Temperature Estimate is included in the CAN broadcast messages. See CAN signal `INV_Coolant_Temp` in CAN message 0x0A2 in the section C, 'Appendix: CAN Broadcast Messages'.

6.4.1.4 Continuous Variable PWM Map

The Continuous Variable PWM Method then considers the minimum PWM frequency from different criteria such as the speed and current presented prior and selects the highest of the PWM frequencies required by each criteria collectively. It also then applies the saturation from the Minimum PWM Frequency EEPROM Parameter and Maximum PWM Frequency EEPROM Parameter. This results in a map that would look like Figure 4 (for iM-375DZ-D).

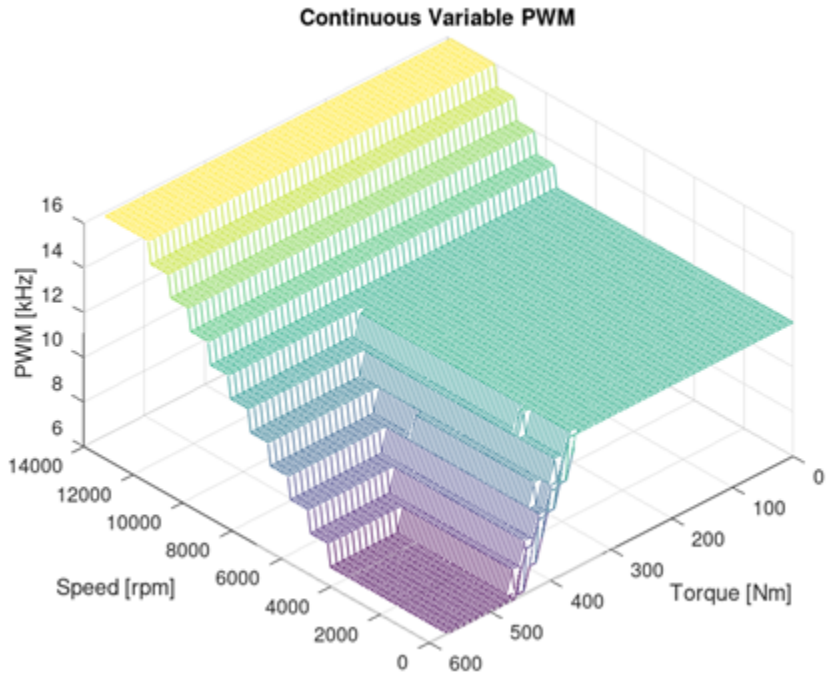


Figure 14: Example of PWM frequency map for Continuous Variable PWM Method

The above map is calculated continuously by the inverter for the current operating point and the figure above is just for a representation and idea of what to expect. High current commands will drop the PWM as necessary, until the PWM must increase to maintain control with higher speed.

6.4.1.5 Low Speed/Stall Condition

Features related to Low Speed / Stall operating condition are still present and the feature applied with the same criteria as it has been prior. The inverter will enter stall if:

- Motor Electrical Speed $<$ 30 Hz.
  - The CM200DX now enters stall $<$ 60 Hz.
- Motor Current $>$ 20% of stall PWM current limit (calculated internally).

When entering the Stall condition, the inverter can switch to a different and lower PWM frequency to maximize the amount of available current. The EEPROM Parameter controlling the frequency used in Stall cannot be set greater than the minimum PWM Frequency.

It is extremely important to note that some motors can have extremely high losses or become uncontrollable if operated at very low PWM frequencies. When the PWM frequency is lowered the PWM induced ripple current in the motor will increase. A low Stall PWM frequency should not be chosen unless the operation at that PWM frequency has been validated.

6.4.1.6 PWM Modes

The Continuous Variable PWM Method can be used in three different modes.

- Nominal PWM Frequency Only (With Stall)
- Continuous Variable PWM Method Enabled (With Stall)
- Continuous Variable PWM Method Enabled (Without Stall)

Nominal PWM Frequency Only (With Stall) is chosen by setting:

- `PWM_Var_Freq_Mode_EEPROM` = **0**

This will run the nominal PWM set by `PWM_Nominal_Freq_EEPROM_(kHz)` for all operating conditions other than stall, and will run the stall PWM `PWM_Stall_Freq_EEPROM_(kHz)` when the conditions are met for stall.

Continuous Variable PWM Method Enabled (With Stall) is chosen by setting:

- `PWM_Var_Freq_Mode_EEPROM` = **1**

This mode is typically what is used with BorgWarner Portland motors. This mode will allow entering the Stall mode when the conditions are met.

Continuous Variable PWM Method Enabled (Without Stall) is chosen by setting:

- `PWM_Var_Freq_Mode_EEPROM` = **2**

This setting has the same functionality as Mode "1" but prevents entering the Stall Condition region. This mode is useful for motors that either don't work well at lower PWM frequency or the inverter doesn't benefit from it.

Alongside the EEPROM settings listed above, it is highly recommended to limit the torque ramp rate of the system (via EEPROM `Torque_Rate_Limit_EEPROM_(Nm)_x_10`) to no more than 70 Nm/3ms or a setting of 700 for HVH250-115 motor core motors (e.g. iM-225/iM-375). This rate corresponds to a torque ramp rate of 23,000 Nm/s, any rate faster than this may cause stability issues.

6.4.2 Recommended EEPROM Settings

The BorgWarner Portland motors and inverters (e.g. iM-225DX-D, im-375DZ-D, iM425DZ-D, etc.) are typically configured to have the recommended settings as shown in the table below.

Table 9: Recommended EEPROM Settings

EEPROM Parameter	Recommended Setting
PWM_Nominal_Freq_EEPROM_(kHz)	10
PWM_Minimum_Variable_Freq_EEPROM_(kHz)	6
PWM_Maximum_Variable_Freq_EEPROM_(kHz)	12
PWM_Stall_Freq_EEPROM_(kHz)	2
PWM_Var_Freq_Mode_EEPROM	1
PWM_Var_Dwell_Rate_EEPROM_(x3ms)	10

6.4.3 EEPROM Setting Safety Checks

To help avoid improper setting of PWM frequency values, the Continuously Variable PWM Method has some built in safety checks for the EEPROM parameters associated with it. These checks only apply when the Continuously Variable PWM Method is enabled via the PWM_Var_Freq_Mode_EEPROM parameter.

In the order these checks are applied, they are listed below:

1. If any of the PWM frequency settings are set outside the acceptable range of the inverter they default to 12 [kHz].
   (a) Example: For an iM-375DZ-D using a CM350DZ if the PWM_Nominal_Freq_EEPROM_(kHz) is set to 30 [kHz] it will revert to 12 [kHz].
2. If the Maximum PWM setting (PWM_Maximum_Variable_Freq_EEPROM_(kHz)) is less than the minimum required PWM for 10 switching events per electrical revolution using the speed from the Max Speed EEPROM Parameter, then the Maximum PWM Frequency setting is raised to the PWM frequency required for 10 switches per electrical cycle.
   (a) Example: For the iM-375DX-D if PWM_Maximum_Variable_Freq_EEPROM_(kHz) is set to 8 [kHz] and Max_Speed_EEPROM_(RPM) is set to 11000 rpm. The PWM_Maximum_Variable_Freq_EEPROM_(kHz) will change to 10 [kHz]. Based on, 11000 [rpm] * 10 [Poles]/120 * 10 = 9166.7 [Hz] which rounds up to 10 [kHz].
3. If the Minimum PWM frequency (PWM_Minimum_Variable_Freq_EEPROM_(kHz)) is greater than the Nominal PWM frequency (PWM_Nominal_Freq_EEPROM_(kHz)) the Minimum PWM frequency EEPROM parameter will be set to the Nominal PWM frequency.
   (a) Example: If PWM_Minimum_Variable_Freq_EEPROM_(kHz) is set to 10 [kHz] and PWM_Nominal_Freq_EEPROM_(kHz) is set to 8 [kHz], PWM_Minimum_Variable_Freq_EEPROM_(kHz) will revert to 10 [kHz].
4. The Maximum PWM frequency PWM_Maximum_Variable_Freq_EEPROM_(kHz) cannot be less than the Nominal PWM frequency (PWM_Nominal_Freq_EEPROM_(kHz)). If so the Maximum PWM frequency will be set to the Nominal PWM frequency.
5. The Stall PWM frequency (PWM_Stall_Freq_EEPROM_(kHz)) cannot be greater than the Minimum PWM frequency (PWM_Minimum_Variable_Freq_EEPROM_(kHz)). If so the Stall PWM frequency

will be set to the Minimum PWM frequency.

(a) When the Continuously Variable PWM Method is not enabled this check happens with the Nominal PWM frequency.

6.4.4 Upgrading from 6525 or Earlier

New BorgWarner Portland CMxxx inverters may come with the appropriate firmware already flashed for Continuously Variable PWM Method, but it is also possible to upgrade an existing CMxxx inverters with older firmware to use this functionality. All this requires is reprogramming the inverter with the latest CM firmware (version equal to our higher than 6526). Continuously Variable PWM Method was first introduced in firmware 6526. Reprogramming the inverter is a topic covered in the general Software Manual and will not be discussed here. Instead, a simple guide to check the new Continuously Variable PWM Method settings are correct is presented below.

Step 1: Save all existing EEPROM parameters to a file for reference.

- Reprogramming can reset resolver calibrations and user settings, so it's bets to save them

Step 2: Reprogram the inverter with software 6526 or later.

- After the reprogram the GUI may present an error saying the EEPROMs have changed. Click OK as this is expected.

Step 3: Check and change the EEPROM settings.

- The new list of EEPROMs should now include those listed in this manual but they may have odd or erroneous values.
- Change these to the desired settings. The recommended settings are listed with the EEPROM descriptions in this manual.
- Set the `PWM_Var_Freq_Mode_EEPROM` = 1 to turn on the CVPWM

Step 4: Power cycle the inverter and check that the EEPROM values are as expected.

6.5 Inverter Hot Spot

6.5.1 Overview

BorgWarner Portland (formerly Cascadia Motion) inverters have a peak current rating and continuous current rating based on hardware temperature limitations. To protect inverter components from overheating, a Inverter Hot Spot Model has been implemented to limit the duration of peak current. This model allows peak current for a limited duration, dependent on coolant temperature, while continuing to protect the inverter against currents that exceed the continuous rating.

This methodology is implemented in the form of a simple thermal model fed into a PI regulator. The target max hot spot temperature is typically 125 °C (Tmax/Ref/Target). This regulator allows the inverter to have the full peak current for typically 30 seconds, then the hot spot temperature model and PI regulator de-rates the maximum motor current as necessary to maintain a max hot spot temperature at the Tmax value. The hot spot temperature is not a physical temperature at any particular location. The hot spot temperature is just a model value used to protect the inverter based on testing at BorgWarner Portland.

The Inverter Hot Spot Thermal Model takes in: actual motor current in Apk, coolant temperature in °C, and available maximum current limit (peak_current) to predict a hot spot temperature in °C and actual maximum inverter current (Max current). With time the temperature in the model will change as it reacts to the actual operating conditions.

The general architecture of control is presented below in 15.

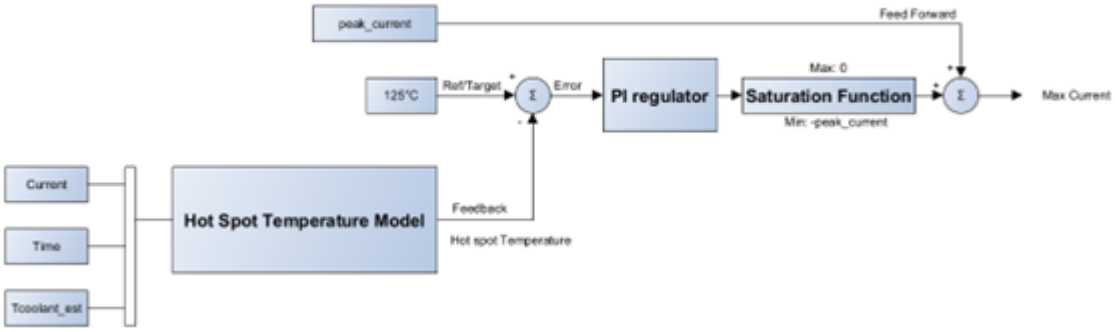


Figure 15: General Architecture of Hot Spot Thermal Model

Through this methodology, compared to a steady state model, more current is allowed for a limited duration, assuming the hot spot temperature is not already at its maximum target of Tmax. The maximum current limit (peak_current) depends on DC Bus Voltage, Coolant temperature, and PWM frequency can vary significantly from the data sheet rating.

The Inverter Hot Spot Model assumes a motor current squared heating effect (resistive heating). Elements of the inverter that are most impacted by long duration currents experience resistive heating (e.g.bus bars, DC link capacitor, etc.).

6.5.2 Implementation

The Inverter Hot Spot Thermal Model is implemented in the code base only for GEN5 inverters. These inverters are listed in the table below. No user input is required to utilize the model and its inverter protection.

Table 10: Hot Spot Models

Hot Spot Thermal Model Inverters
CM200DX
CM200DZ
CM350DZ
CM350SiC

6.5.2.1 Continuous Current Rating

The Inverter Hot Spot Thermal Model results in a continuous current rating that is dependent on the coolant temperature. Higher coolant temperatures result in lower continuous current ratings. The data sheet continuous current rating for CMxxx inverters is indicated at a coolant temperature of 45°C. The hot spot temperature limit (Tmax) is 125°C for most CMxxx inverters (see table below). So, the deltaT for these inverters is 80°C.

The continuous current rating that results from different coolant temperatures can be calculated using the following formula:

Inverter Continuous Current Rating at $Tc = sqrt((T_{max} - T_c) * (I_{cont}@deltaT)^2/deltaT)$

Table 11: Inverter Hot Spot Parameters

Parameter	CM200DX	CM200DZ	CM350DZ	CM350SiC
T_{Max} (°C)	125	125	125	125
$deltaT$ (°C)	80	80	80	80
$Icont@deltaT$ (Arms)	300	200	500	500

The above table gives the deltaT and continuous current rating (in Arms) for various inverters at the time of the manual being written.

For example, the continuous current rating for the CM350DZ at 65°C would be
$sqrt((125 - 65) * 500^2/80) = 433Arms$.

If the coolant temperature is less than 45°C, then the continuous current rating will increase above the rating at 45°C. However, practically speaking it is quite difficult for an installation to guarantee coolant temperatures less than 45°C under all ambient conditions.

6.5.3 Inverter Hot Spot Settings/Reporting

There are no required user inputs to activate the Inverter Hot Spot Thermal Model outside of using the appropriate inverter. The model is always active.

There are two new signals relayed over CAN which report the state of the hot spot limiter. CAN details are provided in the latest DBC file.

Table 12: Inverter Hot Spot CAN Signals

CAN Signal	Description
`INV_Hot_Spot_Temp_Inverter`	The output of the Inverter Hot Spot Temperature Model. Reported in °C.
`INV_Limit_Hot_Spot_Inverter`	A status flag that indicates when the output of the PI regulator is limiting the maximum motor current. It does not indicate if actual motor current limiting is occurring, just that it could be occurring.

6.6 Motor Hot Spot

6.6.1 Overview

BorgWarner Portland (formerly Cascadia Motion) inverters also have a motor hot spot model which is utilized in stall conditions. This model operates in a similar fashion to the Inverter Hot Spot, only for the estimated motor end winding temperature during stall operation. This model is implemented to address uneven heating during prolonged DC current in stall, and the instance of the installed motor temperature sensor not accounting for the excessive end winding temperature given said uneven heating. This model has hard coded and is applied to all BorgWarner Portland catalog motors as discussed below.

This methodology is implemented in the form of a simple thermal model fed into a PI regulator. The target max motor hot spot temperature is 220 °C ($Tmax/Ref/Target$). This regulator allows the inverter to have the full motor current in stall for between 1 - 45 s depending ont he starting motor temperature, then the hot spot temperature model and PI regulator de-rates the maximum motor current as necessary to maintain a max hot spot temperature at the $Tmax$ value.

The Motor Hot Spot Thermal Model takes in: actual motor current in Apk, measured motor temperature in °C, and available maximum current limit (peak_current) to predict a hot spot temperature in °C and actual maximum motor current (Max current). With time the temperature in the model will change as it reacts to the actual operating conditions.

Through this methodology, compared to a steady state model, more current is allowed for a limited duration, assuming the hot spot temperature is not already at its maximum target of $Tmax$.

6.6.1.1 Continuous Current Rating

The Motor Hot Spot Thermal Model results in a continuous current rating that is dependent on the motor temperature temperature. Higher motor temperatures result in lower continuous current ratings. The model is centered around allowing 5s of peak motor current at 110°C motor temperature and then derating to 50% of the max motor current.

The calculation is then the same as the Inverter Hot Spot model except now with a $Tmax = 220$°C, $deltaT = 110$°C, and the $Icont@deltaT$ is $0.5 * I_{peak}$ of the motor. Thus the formula to calculate continuous stall current can be found as such:

Motor Continuous Stall Current Rating at $Tm = sqrt((T_{max} - T_m) * (I_{cont}@deltaT)^2/deltaT)$

These values are summarized below for the catalog BorgWarner Portland motors

Table 13: Motor Hot Spot Parameters

Parameter	SS250-115-DOM	SS250-115-SOM	SS410-150-DOM	SS146-130-DOM
T_{Max} (°C)	220	220	220	220
$deltaT$ (°C)	110	110	110	110
$I_{cont}@deltaT$ (Arms)	450	200	450	250
I_{peak} (Arms)	900	400	900	500

Note this model is only active during stall, and will likely not be felt in general operation other than

during instances of prolonged stall. the table below provides a reference for how long peak motor current will be allowed when entering stall at different motor temperatures.

Table 14: Motor Hot Spot Peak Current Times

Motor Temperature	Time @ Peak Current	Continuous Current
45 (°C)	10s	65%
110 (°C)	5s	50%
180 (°C)	1s	30%

For instances of applied motor current less than the motor peak current in 13 these times will be extended.

6.6.2 Motor Hot Spot Settings/Reporting

There are no required user inputs to activate the Motor Hot Spot Thermal Model. The model is always active.

There are two new signals relayed over CAN which report the state of the hot spot limiter. CAN details are provided in the latest DBC file.

Table 15: Motor Hot Spot CAN Signals

CAN Signal	Description
INV_Hot_Spot_Temp_Motor	The output of the Motor Hot Spot Temperature Model. Reported in °C.
INV_Limit_Hot_Spot_Motor	A status flag that indicates when the output of the PI regulator is limiting the maximum motor current. It does not indicate if actual motor current limiting is occurring, just that it could be occurring.

6.7 Maximum Output Current

Inverter output current is automatically limited in software to match capability with operating conditions. Starting with firmware version 6505 output current is limited based on PWM switching frequency and DC bus voltage. Then in firmware version 6511 and newer, output current is also limited based on coolant temperature estimate. Where coolant temperature is estimated from module temperatures and operating output current. When coolant temperature is 45°C or below there is no reduction in output current due to coolant temperature. There are two separate linear de-rates when coolant is above 45°C. The first linear de-rate is when coolant temperature is between 45°C and 80°C. The second linear de-rate is when coolant is between 80°C and 100°C. Current limits at 45°C and 80°C are provided below. At 100°C the current limit is zero amps. Therefore with 62.5°C coolant the maximum output current will be 50% between the 45°C and 80°C limits. With 90°C coolant the maximum output current will be 50% between 80°C limit and zero.

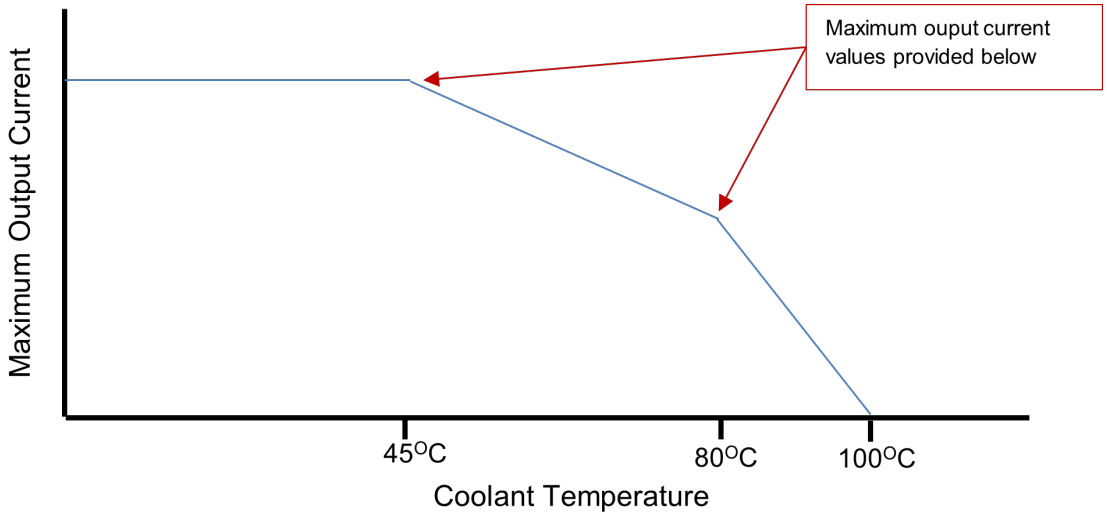


Figure 16: Maximum Output Current

The current limits provided in this manual only covers limits due to coolant temperature, switching frequency, and DC bus voltage.

Table 16: Max Currents

Inverter	Hard Limit on Output Current (A_{RMS})
CM200DX	700
CM200DZ	450
CM350DZ	900
CM350SiC	900

All inverters have a maximum hard limit that will not be exceeded regardless of temperature and switching frequency. Starting with firmware version 651E the hard limit is applied after all other limits.

Therefore in low frequency operation the ratio of current will apply to the maximum output current before the hard limit is applied. Firmware prior to 651E will apply the hard limit before all other limits. Therefore ratio of current for low frequency operation will apply to the hard limit for the lower frequency switching frequencies. Both versions will prevent the final maximum current from exceeding the hard limit. However this difference in when the hard limit is applied significantly effects the maximum output current in low frequency operation. Firmware version 651E and newer allows significantly higher output current in low frequency operation, when using low switching frequencies where the limit before the hard limit is higher than the hard limit. See section 6.3, 'Stall Burst' for more details.

6.8 Download Diagnostic Data

6.8.1 Overview

This section outlines how to download diagnostic data stored in the inverter's volatile memory after a fault occurs.

IMPORTANT NOTE: The Diag Data is lost if the fault is cleared or low voltage power is cycled.

The inverter software provides certain data via the GUI and also via CAN messages. However both CAN and the GUI do not provide data at a very high rate. For a PWM frequency of 12kHz and a CAN message rate of 10ms is still once every 120 PWM cycles. Sometimes it is necessary to be able to examine internal variables with a much finer resolution.

BorgWarner Portland has implemented a feature into our software that keeps an ongoing record of certain internal variables (using a circular buffer). The data is referred to as the Diag Data. The variable set is recorded at every PWM cycle. The buffer length is subject to change but is currently 160 records long and 18 variables are stored. Each variable is internally stored in a 32-bit internal format. However, when the Diag Data is transmitted the data is sent in a 16-bit format converted into real world units like what is used with the RMS GUI and CAN broadcast messages.

The Diag Data is used to diagnose issues that happen in a very narrow window of time. Keep in mind that 160 records at 12kHz is only 13.3ms. Additionally, due to the limited amount of data that can be recorded items such as temperature are not recorded.

6.8.2 Theory of Operation

The internal circular buffer is updated at every PWM interrupt. The data is stored in the buffer at the end of the PWM interrupt. Thus, it contains the feedback variables that were captured as well as the results of the internal control loop calculations. It also records the fault information, if any.

If a fault has occurred the buffer will continue to record 5 more records after the fault has occurred. It will then stop recording until the fault has been cleared.

The assignment of variables to the circular buffer is subject to change. It is currently set as follows:

Index	Variable	Multiplier	Description
1	Gamma_Resolver	10	Instantaneous angle calculated from the resolver feedback. (0 = 0 degrees, 1.0 = 360 degrees).
2	Gamma_Observer	10	Resolver feedback after filtering of observer.
3	Sin_corr	100	Resolver Sin input
4	Cos_corr	100	Resolver Cosine input
5	Ia_corr	10	Phase A adjusted current value
6	Ib_corr	10	Phase B adjusted current value
7	Ic_corr	10	Phase C adjusted current value
8	Vdc	10	DC Bus voltage
9	Iq_cmd	10	Q-axis current command
10	Id_cmd	10	D-axis current command
11	Modulation	10000	Modulation index
12	Flux_weakening_out	10	Amount of field weakening current
13	Vq_cmd	10	Q-axis voltage command
14	Vd_cmd	10	D-axis voltage command
15	Vqs_cmd	10	Q-axis voltage command in stationary reference frame.
16	PWM Frequency	1	Starting with version 651E, PWM Frequency is logged instead of 12V Voltage.
17	Run_faults (lo word)	NA	Run faults low word. See section D.2, 'Run Faults'.
18	Run_faults (hi word)	NA	Run faults high word. See section D.2, 'Run Faults'.

6.8.3 Downloading Data using GUI

To capture the contents of the buffer press the "Download Diag Data" button on the GUI after a fault has occurred. If your version of the GUI software does not have this button please visit the Cascadia Motion website for the latest release of the GUI software.

You will be prompted for a filename to save the data to.

After selecting the file you will be prompted to press "OK" to start the download. After pressing "OK" it may take several minutes to download the data. When the download is complete the GUI software will prompt with a message.

With some versions of the RMS GUI and certain computers it is possible that the download might look frozen. Please continue to allow the download to continue till it is finished.

It is possible to download the Diag Data without a fault having occurred. The Diag Data will then represent a small snapshot of time at the time of download.

6.8.4 Downloading Data using CAN

Diagnostic data is broadcast over the message ID 0x0AF automatically after a fault has occurred and when `CAN_Diag_Data_Tx_Active_EEPROM` is set to 1. Diagnostic data can also be triggered and

broadcast by writing a non-zero value to address 31 using the parameter message (message ID 0x0C1). The default `CAN_ID_Offset_EEPROM` parameter is assumed to be 0x0A0. In other words, add 0x00F to `CAN_ID_Offset_EEPROM` to get the message ID that broadcasts diagnostic data and add 0x021 to `CAN_ID_Offset_EEPROM` to get the parameter message ID.

The format for CAN broadcast is as follows: **CAN MSG ID = Offset + 0x00F**

Byte							
0	**1**	**2**	**3**	**4**	**5**	**6**	**7**
Record #	0	gamma_resolver		gamma_observer		sin_corr	
Record #	1	cos_corr		Ia_corr		Ib_corr	
Record #	2	Ic_corr		Vdc		iq_cmd	
Record #	3	id_cmd		modulation		flux_weak_out	
Record #	4	vq_cmd		vd_cmd		vqs_cmd	
Record #	4	PWM Freq		run_faults (lo)		run_faults (hi)	

Each CAN message has 8 bytes. For CAN, each message includes an index byte, a sub-index byte, and 6 bytes for diagnostic parameters. Total number of CAN messages needed to send one complete record of 18 diagnostic parameters is 6.

Each message is sent at 10 milliseconds rate. It takes 60 milliseconds to send one complete data record. For 160 records, it will take 160 x 60 milliseconds = 9.6 seconds

6.8.5 Output File Format

Internally the software calculations are all done in per unit, that is, 32-bit fixed point format. However, the output format for each variable is 16-bits, that is, each variable is stored as a 16-bit decimal value in the output text file. For GUI, the data is stored in a comma delimited format with one line per record. The CAN output file format primarily depends upon the CAN data logger used. In general, CAN data is stored with one CAN message per line.

In order to convert the data to engineering unit, divide each parameter with its relevant multiplier factor.

BorgWarner Portland has developed a data viewer for the Diag Data file format produced by the RMS GUI. The data viewer can be downloaded from the Cascadia Motion website. See figure 17 for an example:

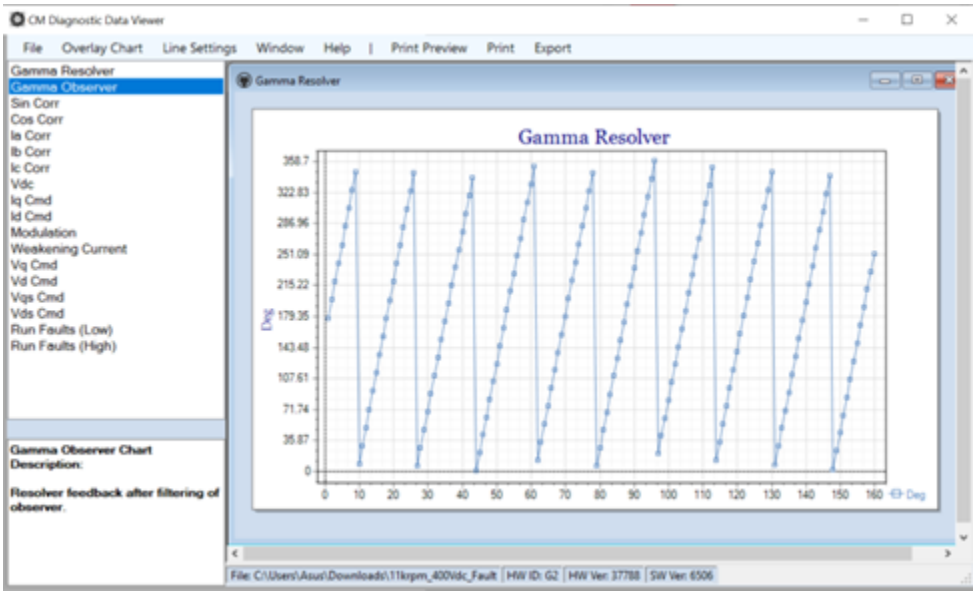


Figure 17: Diag Data Viewer

7 Features

7.1 Using Speed Mode

7.1.1 Overview

BorgWarner Portland (formerly Cascadia Motion) inverters can either operate in Torque mode or Speed mode. In torque mode, which is most commonly used, the user provides a torque command and the controller attempts to achieve that operating torque. Torque mode is most analogous to what is experienced in driving a normal car. The accelerator pedal position provides a command that is roughly proportional to torque.

Speed mode can be useful mostly for certain testing scenarios or for specific vehicle type applications (e.g. marine). In some instances speed mode can be useful in multi-speed transmissions.

Certain application might require switching between torque and speed mode while the inverter is actively controlling the motor. The mode switching can be accomplished, but only when using CAN mode.

7.1.2 Speed Mode

Speed Mode allows the user to control the speed of the motor. The controller has a method to measure the speed of the motor. The actual speed of the motor is compared to the commanded speed of the motor and the resulting error drives a regulator to increase or decrease the torque as needed to achieve the commanded speed target.

The output of the speed regulator is a torque command. The output torque command is not processed by the Torque Rate Limit function.

Because the speed regulator can drive a motor to the maximum torque it must be used with caution. If for example the resolver is not calibrated a motor may not make any torque even with a large amount of current flowing. Thus the motor could quickly to maximum current without any movement of the motor.

For the purposes of this manual, it is important to define what Forward and Reverse mean. The Forward direction is when a motor spins and the motor feedback speed is positive. The Reverse direction is when a motor spins and the motor feedback speed is negative. The controller does not have a concept of clockwise and counterclockwise. That being said many of the motors that the controller is configured for are setup for counterclockwise as being Forward, but not all.

If the controller is to operate in Speed mode at all times, then the EEPROM variable should be used to set the mode to be Speed Mode. The `Run_Mode_EEPROM` parameter should be set to 1 to achieve speed mode.

When operating in speed mode the speed controller will output a torque that is limited to the motor torque limits. The motoring torque (positive torque) is limited to `Motor_Torque_Limit_EEPROM_(Nm)_x_10` and the regeneration torque (negative torque) is limited to `Regen_Torque_Limit_EEPROM_(Nm)_x_10`.

Starting in Gen 3 firmware version 2044 these torque limits may be reduced depending on the configuration of certain de-rating features. These features are the Break Speed setting, the BMS CAN message limits, or Motor Temperature de-rating (Zero/Full Torque Temperatures).

Note: For the Speed Mode to operate properly the two torque limits should be set to values that allow the regulator to work properly. For example, if the regen torque limit is set to zero and the motor is operated with light load the controller will not be able to regulate the motor speed properly.

Figure 18 is a diagram of the speed regulator.

The Commanded Speed as indicated in the diagram is after the Speed Rate Limiter has been applied to the command (whether it be from CAN or VSM mode).

As shown below the Commanded Torque acts as a feed-forward to the output of the speed regulator. For many applications the Commanded Torque should be kept at 0.

If it is desirable to have faster response it may be necessary to provide a Commanded Torque feed-forward. This type of operation is only recommended for CAN control mode. The CAN Commanded Torque becomes the feed-forward value and is NOT constrained by the Torque Rate Limit function. Care must be exercised when applying this feed-forward.

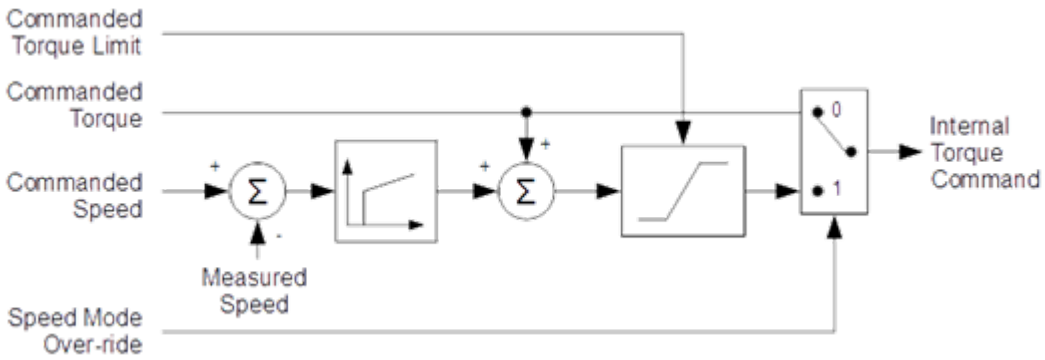


Figure 18: Block Diagram of the Speed Regulator

The PID block of the speed regulator has 4 adjustable gains. The speed regulator PID loop is operated at the PWM frequency.

The PID inputs and outputs are done using per-unit calculations. If needed contact technical support for the base units for the particular motor being operated.

The most common set of per unit bases are as follows:

- Voltage Base = 200V
- Current Base = 300A
- Frequency Base = 175 Hz

From these three bases there are several derived bases:
$Torque_{base} = V_{base} * I_{base}/(2 * PI * F_{base}) = 54.57Nm$
$Speed_{base} = 120 * F_{base}/Poles$ (e.g. for 10 pole motor Fbase = 2100 rpm)

So when attempting simulate or model the PID regulator the inputs and outputs must be first scaled by the per unit bases. For example, if the Commanded Speed was 4100 rpm then the per unit commanded speed would be 4100 / 2100 = 1.952. The output torque from the PID regulator would have to be

multiplied by the torque base to get the actual torque command. For example if the torque command was 1.0 in per unit the actual torque commanded would be 1.0 * 54.57 = 54.57 Nm.

For IPM motors the torque command generated from the speed regulator is used by the id/iq look up table every 3ms. IPM motors do not use a torque regulator. The torque command is used directly every 3ms to find a new id/iq current command.

7.1.3 Speed Mode when in CAN Mode

CAN Mode allows the control of the motor from the CAN Command message. The CAN Command message contains the following information:

Table 17: Speed Mode CAN Command Message

Byte (s)	Name	Description
0,1	Commanded Torque	If the controller is in Torque Mode then the Commanded Torque is the torque command for the motor. Positive numbers represent motoring torque, negative numbers represent regen torque. If the controller is in Speed Mode then the Commanded Torque is used as a feed-forward into the speed regulator.
2,3	Commanded Speed	If the controller is in Speed Mode then the Commanded Speed is the input to motor speed regulator.
4.0	Commanded Direction	If byte 4.0 is set to 1 then the commanded direction is "Forward" where positive torque results in positive speed. If byte 4.0 is set to 0 then the commanded direction is Reverse.
5.0	Inverter Enable	If byte 5.0 is set to 1 then the inverter is enabled, if set to 0 then disabled.
5.1	Active Discharge	See section 7.3, Inverter Discharge
5.2	Speed Mode Over-ride	If this bit is 1 and the controller EEPROM is set for Torque Mode then it will transition the controller to Speed Mode as long as this bit remains 1. If the bit goes back to zero then the controller will go back to Torque Mode. The bit has no effect if the controller EEPROM is set for Speed Mode.
6,7	Commanded Torque Limit	If the Commanded Torque Limit is set to 0 then the torque limits default to the parameters set in the EEPROM.For non-zero torque limits the motoring and regen torque limit is set to the Commanded Torque Limit.

The Commanded Speed can be a positive or negative number. If the Commanded Direction is Forward then a positive Commanded Speed results in a positive speed, and a negative Commanded Speed results in a negative speed. If the Commanded Direction is Reverse then a positive Commanded Speed results in a negative speed, and a negative Commanded speed results in a positive speed.

When using CAN mode for speed control it is important to send periodic command messages to the controller. The torque limits that are affected by speed (regen fade speed for example) are only updated when a Command Message is received (this is no longer true starting with Gen 3 version 2044).

The controller will allow a negative torque to reverse the direction of the motor ONLY if the `Regen_Fade_Speed_EE` parameter is set to 0. If this parameter is set to a non-zero value then any negative torque commands (regen) will be set to zero when the speed of the motor reverses from it commanded direction.

The controller can be enabled directly into speed mode without first entering torque mode by using the Speed Mode Over-ride bit. The controller can also transition from torque mode to speed mode and back to torque mode without disabling the controller by using the same bit.

The Commanded Torque is used as a feed-forward into the speed regulator. Use of the feed-forward may allow a smoother transition or allow for a quicker response when using speed mode.

7.2 Shudder Compensation

7.2.1 Overview

Using an electric motor in a vehicle can expose driveline resonances (shudder) that might not normally be noticed in an ICE vehicle. Typically, these resonances occur at very low speeds and moderate torque levels.

The shudder compensation system implemented within the BorgWarner Portland inverters provides a mechanism for the user to try and counteract the resonance.

The basic idea is to provide a compensating torque that tries to negate any oscillating component to the speed. That is, if the speed is found to be varying (oscillating), an additional torque is added to the command that attempts to cancel the oscillation.

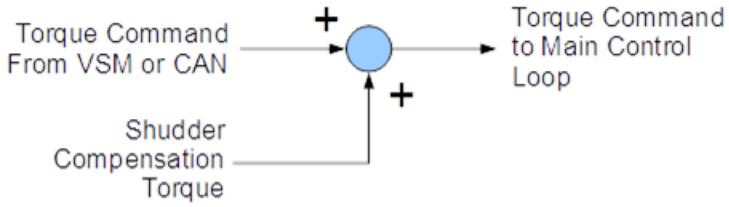


Figure 19: Shudder Torque Implementation

Figure 19 shows the mechanism for including the shudder compensation torque into the torque command. If shudder compensation is enabled the shudder torque value will be added to the normal torque command that comes from the VSM (vehicle state machine) or from a CAN command.

In Gen5 firmware 652E and beyond, for standard BorgWarner Portland catalog products, the inverter will modify the torque command which has already been limited to the max estimated torque capability. In firmware before 652E if the command is higher than what the system can achieve, the shudder compensation is only applied to the command and thus may not take effect in the real resultant torque.

For example, before 652E, if the command was 100 Nm and the system current limit maxed out such that only 50 Nm was possible, even if shudder compensation is allowed to modify the command 30 Nm, that modification would never lower the command below what the real resultant torque is and thus shudder compensation would have no effect. In firmware 652E and after, with standard catalog CM products, the inverter automatically detects that the system is limited and will adjust the torque command such that the 30Nm allowed shudder compensation will act around the 50 Nm capability instead of the 100 Nm user command. This will allow shudder compensation to at least lower the limited torque command to reduce shudder.

The mechanism for calculating the amount of shudder torque compensation is shown in Figure 20. The compensation algorithm compares the electrical speed of the motor to a filtered version of the speed. The difference between the filtered version and the actual speed is what generates the correcting torque to be applied (with the sign negated). The output of the comparison is then clamped to a value between +TCLAMP and -TCLAMP. The compensation torque is phased out below Shudder Fade Speed and also at higher speeds by two speed parameters, Shudder Speed Lo and Shudder Speed Hi.

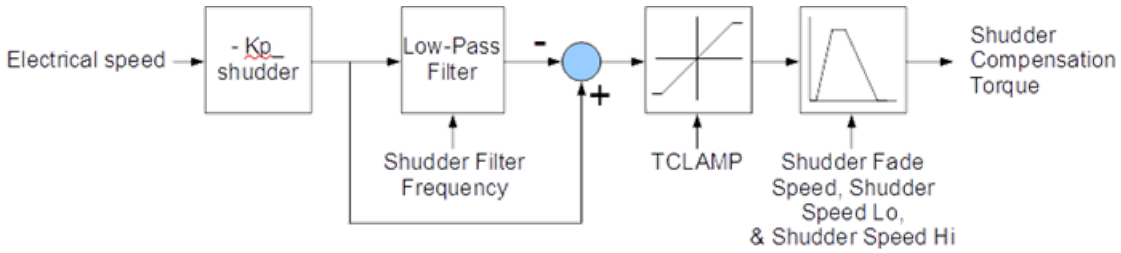


Figure 20: Shudder Torque Algorithm

The shudder compensation calculation is executed every 3ms. The algorithm is intended to correct driveline oscillation that typically occurs at frequencies less than 10Hz and at low vehicle speeds.

The amount of compensation torque applied is controlled by the Kp_Shudder gain. To understand how it is applied it is necessary to understand the calculation units.

The electrical speed used in this calculation is a per unit value of the electrical frequency. For most motors the per unit base value is 175Hz. An electrical speed value of 1.0 thus represents 175Hz (for a 10 pole motor this would be 2100 rpm.)

The resulting Shudder Compensation Torque is also in per unit. For most motors this base quantity for torque is 54.57Nm.

So for example if the delta between measured and actual speed is 100 rpm, this would be a per unit value of 0.0476 (for a 10 pole motor). If this is multiplied by a Kp_Shudder gain of 20.00 the resulting torque in per unit is 0.952 per unit (51.97Nm with the standard per unit base).

7.2.2 Modes

The shudder compensation mode can be selected by setting the `Shudder_Compensation_Enable_EEPROM` parameter to 0, 1 or 2.

Mode 0 will disable shudder compensation.

Mode 1 will implement shudder compensation as outlined in this section.

Mode 2 will implement shudder compensation as outlined in this section but will restrict shudder torque application for torque commands less than 1% of the `Motor_Torque_Limit_EEPROM_(Nm)_x_10` parameter. The CAN signal `INV_Torque_Shudder` will also be added to the `INV_Commanded_Torque` signal.

If shudder compensation is required, it is recommended to use Mode 2.

7.2.3 EEPROM Settings

There are several parameters used to control the Shudder Compensation system. These parameters are EEPROM parameters and can be accessed either via CAN or through the GUI software. See section A.11, 'Shudder Compensation Parameters' for more details.

The value of `Kp_Shudder_EEPROM_x_100` can be modified via CAN while the inverter is operating. See section 8, 'CAN Protocol for more details on how the CAN Parameter Message (Command 23) can be used to modify Kp Shudder.

7.2.4 Tuning Process

It is difficult to provide a definitive process for tuning the shudder compensation parameters.

The first step would be to look at the frequency of oscillation that is occurring. This can be examined by looking at the speed value broadcast over CAN. Taking a log of CAN data when the shudder is occurring should allow the frequency of oscillation to be determined as well as the speed range it occurs in.

Once the frequency and speed range are determined the EEPROM parameters can be adjusted to make sure the compensation algorithm is effective in the desired range.

Driveline resonance can be also seen at high speeds when sudden changes in torque occur. It is possible that shudder compensation could be applied at higher speeds however, special care should be taken as the torque command is being modified.

The maximum value of shudder torque compensation should be kept as small as possible. It is not desirable to overly modify the Torque command coming the driver. It will be necessary to experimentally determine how much torque correction is necessary to stabilize the speed. The shudder compensation torque value is broadcast over CAN. It may be useful to monitor this value. It may provide insight into how much torque compensation is being utilized.

The Kp_Shudder value should be modified to achieve the desired response to shudder. If the value is too high it may cause its own rapid speed changes or oscillations. If the value is too low the compensation will have little impact.

7.3 Inverter Discharge

7.3.1 Overview

BorgWarner Portland inverters contain a substantial amount of capacitance on the DC bus. The capacitors are internally discharged by a resistor at a very slow rate. To avoid excess power dissipation the resistors are sized such that they will discharge the capacitors from maximum voltage to a level below 60V in less than 5 minutes.

For some applications it is necessary to discharge the internal capacitors faster than 5 minutes. Cascadia Motion inverters employ one of two capacitor discharge methods. The first method, avialable on CM200 and CM350DZ inverters, drives D-axis current through the motor. The second method, available on CM350SiC inverters, activates a dedicated discharge circuit.

7.3.2 Motor Method

CM200 and CM350DZ inverters can discharge the DC bus capacitors by injecting a small amount of current into the motor that is connected. The internal capacitors can be discharged even if the motor is not connected, but at a slower rate.

When this feature is activated the inverter will inject a current into the motor that is equal to 5% of the value set for the `IQ_Limit_EEPROM_(A)_x_10`. The current injected is set in the D-axis and should not normally cause any rotation of the motor.

The time it takes to discharge the capacitors will vary depending on the motor attached, the particular Iq limit, and the DC voltage.

7.3.3 Circuit Method

CM350SiC inverters can discharge the DC bus capacitors by activating a dedicated discharge circuit. The dedicated discharge circuitry provides a monitoring function that determines DC bus drop to verify a power source is not connected. The discharge monitoring functionality also has means of circuit error detection, notifying the inverter if a discharge override is necessary. If a discharge override is necessary, the discharge capability may appear to operate slightly differently.

The typical discharge circuit uses a two-step approach, first using the DC bus monitoring functionality to verify that the Precharge and Main contactors are opened. Once this is verified, an 800 Ohm resistor between DC positive and negative is activated. See figure 21 for an example of a CM350SiC active discharge event, including contactor open timing.

Some errors in the active discharge circuitry allow for a discharge override which results in the inverter applying the 800 Ohm resistor between DC positive and negative at 1kHz, 50% duty cycle.

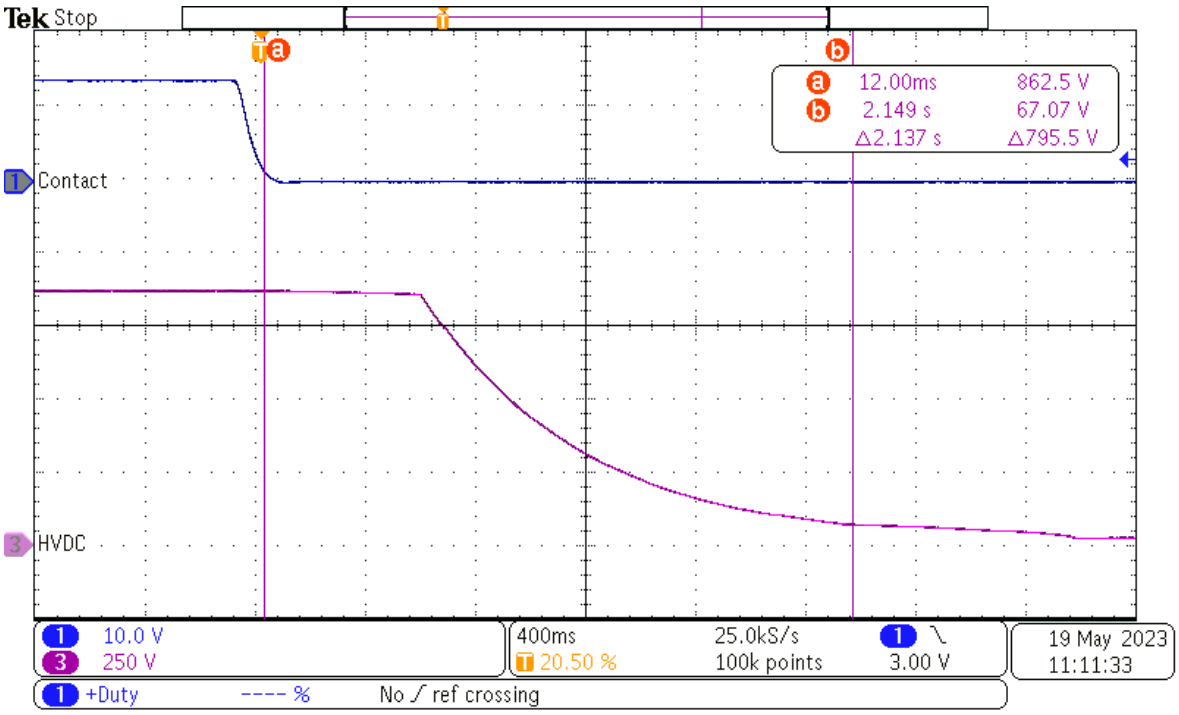


Figure 21: CM350SiC Active Discharge: 850V @ 25°C

7.3.4 Discharging the Inverter

The active discharge process described here requires that the inverter be functioning correctly and that it still have 12V power. It will not function if the inverter loses 12V power. The active discharge can be activated in both VSM mode and in CAN mode.

This section describes different scenarios under which the controller will have the capacitors on the DC voltage side actively discharged. These scenarios include:

1. Ignition is turned off in key switch mode 1
2. A discharge command is sent through the CAN heartbeat command message in CAN mode
3. A fault occurs

Table 18 defines the states used in the discharge process:

Table 18: Inverter Discharge Status

State	Description
0	Discharge disabled
1	Discharge sequence enabled (not actively discharging yet)
2	Speed check (Waiting for speed to go below 75 RPM)
3	Discharge active
4	Discharge complete
5	Discharge error (only available on CM350SiC)
6	Discharge override (only available on CM350SiC)
7	Discharge timeout (only available on CM350SiC)

The Discharge State is shown in the GUI with the following parameter: `Inverter_Discharge_State`.

The Discharge State can also be found in the CAN broadcast message (default CAN ID 0xAA). It is located in byte 4, bits 5 – 7.

7.3.4.1 Mandatory Conditions

Inverter discharge will take place only when all of the following conditions are true:

1. `Discharge_Enable_EEPROM` is set to a value of 1 or 2:

State	Description
0	Discharge Disabled
1	Discharge is enabled without any faults: If `Inverter_Discharge_Enabled_EEPROM` is set to 1, the discharge will not take place when a fault occurs. Shown as value [0 1] in the table below.
2	Discharge is enabled with faults: If `Inverter_Discharge_Enabled_EEPROM` is set to 2, the discharge will automatically take place when a fault occurs. Shown as value [1 0] in the table below.

2. Motor speed must be below 75 RPM. This condition is checked only after one of the events that trigger the discharge take place.

7.3.4.2 Inverter Discharge Triggers

Following events will trigger the discharge process and a discharge command is set:

1. Ignition is turned off in Key Switch Mode 1
2. Discharge command is issued through CAN Heartbeat Command message in CAN mode

3. A POST or RUN fault occurs and Inverter Discharge Mode EEPROM is set to 2.

Once, the discharge command is issued, motor speed is checked. If it is below the threshold (75 RPM), discharge process will begin.

7.3.4.3 Inverter Discharge Complete

As soon as the Inverter Discharge Mode is set to 'Discharge Active' (3), a timer will start and also the DC voltage is monitored. If 9 seconds elapse or the DC voltage falls below 30-V, the discharge process will stop and the Inverter Discharge Mode is set to 'Discharge Complete' (4).

For CM350SiC inverters, if the 9 second timeout elapses with no discharge, the Inverter Discharge Mode will be set to 'Discharge Timeout' (7), which can indicate to the user they may need to prepare for the discharge by opening the Precharge and/or Main contactors, if under user control.

Once the discharge is complete, it will not be reset until one of the following events take place:

1. After ignition input goes low, the power will eventually be recycled and the discharge process will be reset automatically.
2. A Discharge Disable command is issued through CAN if the discharge process was initiated by Discharge Enable CAN command.
3. Clear the faults if there is a fault. Occasionally, a 'DC Low Voltage Threshold' fault may get set during the discharge process.

7.3.4.4 Inverter Discharge Override

The CM350SiC discharge circuitry includes monitoring functionality and provides a means of overriding the discharge circuitry if necessary. This condition is indicated to the user through the Inverter Discharge Mode, 'Discharge Override' (6). No additional action needs to be taken by the user, and the discharge should complete as normal, but discharge timing may be deviate from the typical circuit discharge method.

NOTE: Inverter discharge override is still executed through a circuit, and does not rely on the motor current method for CM200 and CM350DZ inverters.

7.3.4.5 Inverter Discharge with Precharge Bypassed

In certain cases, inverter discharge is allowed when Precharge is bypassed. However, it transfers the responsibility of controlling all outputs to the user. The discharge process needs to have Precharge and Main contactors shut off before initiating the discharge process properly. User must do that.

Moreover, if the key switch mode is 1 (ignition mode), user must also control the OK output in order to take away the 12-V power after discharge is complete.

For CM350SiC inverters, the 'Discharge Timeout' (7) state can be used to determine if the Precharge amd Main contacters need to be shut off by the user.

NOTE: Missing cases in the table below indicate a value of 3 for the `Discharge_Mode_EEPROM` parameter which is an invalid option.

BORGWARNER
7929 SW Burns Way
Wilsonville, OR 97070
www.CascadiaMotion.com
sales@cascadiamotion.com

Case	Precharge Bypass EEPROM	Key Switch Mode EEPROM	VSM/CAN Mode EEPROM	Discharge Mode EEPROM	Discharge Enable	Ignition is turned off	CAN Discharge Command is Sent	A Fault Occurs	Comments
0	0	0	0	0	No				Discharge EEPROM is set to 0
1	0	0	0	1	Yes		X		Discharge controlled by CAN command
2	0	0	0	2	Yes		X	X	Discharge controlled by CAN command
4	0	0	1	0	No				Discharge EEPROM is set to 0
5	0	0	1	1	No				No trigger present to initiate discharge
6	0	0	1	2	Yes			X	A fault can initiate discharge process
8	0	1	0	0	No				Discharge EEPROM is set to 0
9	0	1	0	1	Yes	X	X		Key switch off or CAN command will initiate discharge process.
10	0	1	0	2	Yes	X	X	X	Key switch off, CAN discharge command, or a fault can initiate the discharge process.
12	0	1	1	0	No				Discharge EEPROM is set to 0
13	0	1	1	1	Yes	X			Turning ignition key to off can initiate the discharge process
14	0	1	1	2	Yes	X		X	Turning key switch to off or a fault may initiate discharge process
16	1	0	0	0	No				Discharge EEPROM is set to 0
17	1	0	0	1	Yes		X		Discharge is allowed, however the user needs to control Main and precharge outputs before starting the discharge process.
18	1	0	0	2	Yes		X	X	
20	1	0	1	0	No				Discharge EEPROM is set to 0
21	1	0	1	1	No				In VSM mode with precharge bypassed, Main and Precharge output are under user configuration.
22	1	0	1	2	No				
24	1	1	0	0	No				Discharge EEPROM is set to 0.
25	1	1	0	1	Yes	X	X		All outputs are under user control. After discharge is complete, user needs to turn off OK output to shut down the inverter.
26	1	1	0	2	Yes	X	X	X	
28	1	1	1	0	No				Discharge EEPROM is set to 0.
29	1	1	1	1	No				In VSM Mode with precharge bypassed, Main and precharge output are under user configuration.
30	1	1	1	2	No				

7.4 Self-Sense Assist

The parameter `Self_Sense_Assist_Enabled_EEPROM` should be set to 0, to disable self-sense, unless otherwise instructed by BorgWarner Portland for a customers specific setup. Please contact BorgWarner Portland support if you are need of information about this feature.

7.5 Active Short Circuit

When activated, the active short circuit feature will connect all three motor output phases directly to HVDC- (effectively shorting the motor) when the inverter state machine enters the ‘‘ASC_ACTIVE’’ state. The active short circuit feature is only allowed for certain motor types and is enabled by user configuration, as outlined in section 7.5.2, Modes.

If the inverter is damaged or in a state where it can’t short circuit the motor, then the lack os ASC could result in damage to the inverter under certain conditions. The ASC function requires that the inverter has proper control logic power (typically coming from the 12/24V system).

7.5.1 Inverter State Machine

The Inverter State Machine controls steps related to enabling and disabling the inverter. Each state of the Inverter State Machine is assigned a number. The current value (state) of the Inverter State Machine is broadcast over CAN and is available via the RMS GUI. Two states, ‘‘ASC_INIT’’ and ‘‘ASC_ACTIVE’’ have been inserted into the inverter state machine. These states have been added between the ‘‘IDLE_RUN’’ state and the ‘‘STOP’’ state.

Table 19: Active Short Circuit ISM States

ISM State	State Number
ASC_INIT	13
ASC_ACTIVE	14

When the inverter exits the ‘‘IDLE_RUN’’ state due to either a fault condition or a normal disable request the inverter will check to see if the back EMF of the motor at the current motor speed will create a DC bus voltage that exceeds a certain target DC bus voltage (see equation 1). If the current motor speed exceeds this threshold speed then the inverter will proceed to apply the active short circuit to the motor by first going to the ASC_INIT state. If the current motor speed is less than threshold speed then the inverter will proceed to a normal shutdown process.

For a DX inverter, this DC bus voltage is 500Vdc, while a DZ inverter is 900Vdc.

The speed threshold equation is:

$$\omega = \frac{V_{dctarget}}{\sqrt{3} \times \text{Flux}} \tag{1}$$

Here ω is defined in terms of the motor electrical frequency in rad/s, and Flux is the permanent flux value (Parameter Veh_Flux_Cmd) in Weber’s. To convert the value of speed in electrical rad/s to rpm it is necessary to know the number of motor poles (P), as shown in equation 2.

$$\text{rpm} = \frac{\omega \times 60}{\pi \times P} \tag{2}$$

During the ASC_INIT state the inverter will prepare for the active short circuit to be applied by first opening all the inverter power switches for a user defined amount of time. It will then proceed to the ASC_ACTIVE state and turn on the switches that connect all motor phases to HVDC-.

During Active Short Circuit the motor current may increase, and the motor torque will have some amount of regenerative torque determined by the motor characterstics. The regen torque will hopefully slow the motor down.

If the inverter has a fault condition that disables control of the power electronics, then the ASC will be prematurely disabled.

The ASC_ACTIVE state will remain until the motor speed is detected to be less than a speed threshold found using 1 with a DC bus voltage target of 100Vdc. Once the motor speed falls below the threshold set by this equation, the inverter will turn off all the inverter power electronic switches and proceed to the Inverter State Machine state STOP.

7.5.2 Modes

The ability to short the three-phase outputs of the inverter under certain conditions has been made available via user configuration. There are four modes of operation for the active short circuit feature, configurable through the defsym parameter `Active_Short_Circuit_Enabled_EEPROM`. ASC will only activate if the back-EMF of the motor is above a set threshold, as determined by the motor flux and speed.

Table 20 defines the active short circuit modes:

Table 20: Active_Short_Circuit_Enabled_EEPROM Modes

State	Description
0	ASC Disabled
1	ASC Activates on Fault or Disable
2	ASC Activates on Fault Only
3	ASC Activates on Disable Only

7.5.3 Monitoring ASC State

In addition to the two states added to the inverter state machine, a separate `INV_ASC_State` signal has been added to the Internal States broadcast CAN message. These states allow the user to monitor the active short circuit status.

Table 21 defines the active short circuit modes:

Table 21: INV_ASC_State

State	Name	Description
0	ASC_Disabled	ASC is not supported for this motor, or disabled in EEPROM.
1	ASC_Enabled	ASC is supported and enabled.
2	ASC_Pending	ASC is above voltage threshold to trigger, waiting for fault or disable depending on mode.
3	ASC_Delay	In user-defined EEPROM delay.
4	ASC_Active	Active short circuit is activated.
5	ASC_Complete	Active short circuit event is completed.
6	ASC_Blocked	Active short circuit event blocked by a fault state.
7	ASC_Suppressed	Active short circuit suppressed by user.

7.5.4 Fault Conditions

The following fault conditiosn will block the active short circuit feature from operating:

- run_fault_over_temp_inv_f
- run_fault_hw_gate_f
- run_fault_mod_a_over_temp_f

- run_fault_mod_b_over_temp_f
- run_fault_mod_c_over_temp_f
- run_fault_pcb_over_temp_f
- run_fault_gdb_over_temp_f
- run_fault_gdb_2_over_temp_f
- run_fault_gdb_3_over_temp_f

7.5.5 User Defined Time Delay

The `Active_Short_Circuit_Delay_EEPROM_(us)` parameter can be used to set the delay before the active short circuit feature activates. The delay counter is updated at the control loop rate, meaning if the user sets a delay for 50us, but is running at 12kHz (83us control loop) then the shortest allowable delay is 83us. Similarly, if the user sets a timeout for 150us, the delay will be serviced in 83us increments, meaning the delay will be 166us.

7.5.6 User Speed Off Setting

The active short circuit feature is setup to turn off at a back-EMF value of 100V. If desired, the turn off threshold can be increased via the `Active_Short_Circuit_User_Off_EEPROM` parameter.

7.5.7 Resolver Fault

If the fault that triggers an active short circuit event also results in the loss of speed measurement, a back-up speed measurement, derived through current signal monitoring, will be used to stop the active short circuit event. Under this condition, the reported speed over CAN or the DiagData will be reported as 0 RPM.

7.5.8 ASC Suppression

It may be desired to suppress an upcoming active short circuit event, in which case a CAN Parameter has been made available to suppress the ASC event. The ASC suppression state will persist until cleared by the user. The CAN Parameter address for ASC suppression is 0x48, writing any non-zero value to this address will suppress the ASC feature until a zero is written, clearing the suppression. Note that clearing a suppressed active short circuit event may not re-enable the active short circuit for the pending event.

8 CAN Protocol

This section defines the CAN protocol used by BorgWarner Portland (formerly Cascadia Motion) inverters.

BorgWarner Portland inverters have two CAN interfaces (CAN A and CAN B). The controller is configured to communicate only over CAN A, currently CAN B is reserved for future use.

The CAN interface has multiple purposes:

- Direct control of the motor
- Adjust EEPROM parameters
- Diagnostics and monitoring

8.1 CAN EEPROM Parameters Overview

The CAN Communication baud rate must be set to compatible with the CAN bus being used. All devices on the CAN bus must be at the same baud rate. The available baud rates include 125K, 250K, 500K, and 1M. Higher baud rates tend to be more sensitive to noise than lower baud rates. However, lower baud rates are not able to handles as many CAN messages and can become overloaded. Typical baud rates that are chosen are 250K and 500K.

For PM inverter there is a built-in user configurable CAN termination resistor (120 ohms). The RM inverter do not have a termination resistor built in to the unit. The CMxxx inverter has the ability to connect a termination resistor via making a connection in the I/O connector.

Many CAN related parameters are configurable through parameters that are available through the RMS GUI application as well as CAN. GUI parameters have the same name as mentioned in this document with the exception that they end with the keyword “EEPROM”. Following parameters are used to configure CAN operation:

Please see section A.4, ‘CAN Configuration Parameters’ for a complete table of CAN EEPROM Parameters.

Inverter Command Mode:
This parameter gives the option to operate controller in either VSM mode or CAN mode. In VSM mode, the inverter is operated from the various inputs and outputs of the inverter such brake, accelerator pedal, etc. In VSM mode, broadcast messages are still sent out over the CAN lines. In CAN mode, both GUI and CAN interfaces are active and can be used to monitor and modify parameters. In CAN mode the torque and speed commands come from the Command message. The various inputs of VSM mode have no effect.

0 = CAN Mode
1 = VSM Mode (Default)

CAN ID Offset:
This parameter allows the user to choose their own set of contiguous CAN message identifiers starting with the value in CAN ID Offset. For 11 bit CAN the offset covers a range of 0x00 – 0x7C0. The default offset is 0x0A0. The default range of CAN message IDs is 0x0A0 – 0x0CF. For the J1939 the CAN ID

offset must be in the range of 0x00 – 0xC0. For 29 bit CAN messages the ID can be in the range of 0x0000 – 0xFFC0.

This feature is especially useful when there is more than one controller on the same CAN network.

While setting base address for a controller, it must be made sure that the address range for various controllers do not contain overlapping addresses.

CAN Extended Message Identifier:
This parameter allows switching between CAN standard and extended message identifiers.

0 = Standard CAN Messages (11-bit identifiers)
1 = Extended CAN Messages (29-bit identifiers)

CAN J1939 Option Active:
This parameter allows switching to J1939 format in extended mode. This parameter works in conjunction with 'CAN Extended Message Identifier' parameter above which must be set to 1.

0 = CAN ID will be defined as either 11 bit or 29 bit addressing per the above description.
1 = Extended CAN Messages in SAE J1939 Format

The CAN protocol provides a limited functionality for J1939 protocol with the following fixed parameters:

Priority = 3 (0b001)

Data Page = 0

PDU Format = 0xFF

PDU Specific = CAN ID Offset

Source Address= 0x01 (for both transmitted and received messages)

For example, if CAN ID Offset is set to the default 0xA0, the J1939 CAN message will be broadcast as 0xCFFA001 and so on. The heartbeat command message should be sent out as 0xCFFC001, where the source address of the sending node is still set to be 0x01.

CAN Term. Resistor Present:
In order to use CAN communication, the CAN bus needs to be terminated with a 120 Ohm resistor. PM inverters are equipped with a user configurable termination resistor which is activated through this parameter.

0 = Term. Resistor not active
1 = Term. Resistor active (Default)

If CAN Terminator Resistor is deactivated, it may be necessary to use the GUI interface only since CAN communication may fail without a terminator resistor.

The CM controllers have the ability to activate a CAN terminator, but only by making an external wiring change. Refer to the Hardware manual for more information.

CAN Command Message Active:
To help with the safety of a CAN controlled system it is recommend to activate the CAN Timeout feature. The CAN Timeout feature requires a "heartbeat" CAN Command message to be sent at some

regular interval. The CAN Command message controls the inverter, motor direction, and torque or speed. If the inverter does not receive a CAN command message within the CAN Timeout time (described below) then the inverter will declare a Run Fault of CAN Command Timeout. The inverter will disable the motor. It is important for the user to decide if this Timeout feature is important to their application or not. If CAN communications is lost the inverter continues to hold the last received CAN Command.

0 = The CAN command message Timeout feature is disabled. The controller will hold last received CAN command.
1 = The CAN command message Timeout feature is enabled. A CAN Command message should be sent at some regular interval.

CAN Bit Rate:
250Kbps is the default bit rate. Bus speed can be changed using CAN parameter command message. However, changing this parameter requires a power reset on controller since bus speed is setup only at the initialization of CAN modules in the micro-controller. Also, this input is restricted to valid baud rates. The 4 options for valid baud rate are:

125 = 125 Kbps

250 = 250 Kbps (Default)

500 = 500 Kbps

1000 = 1 Mbps

CAN Active Messages Word:
This parameter is used to enable/disable CAN Broadcast Messages. This parameter is represented as two parameters, CAN Active Messages (Low Word) and CAN Active Messages (High Word) in RMS GUI. Each bit represents a CAN Message broadcast status as follows:

0 = CAN Messages broadcast disabled
1 = CAN Message broadcast enabled (Default)

Please refer to table 26 for details on how to enable/disable each message.

Note, starting with version 2042 the lowest bit of the Hi Word has an opposite effect, if the bit is 0 then it enables a high-speed message at CAN ID Offset plus 16. If the bit is 1 the message is disabled.

CAN Diagnostic Data Transmit Active:
This parameter is used to enable/disable the broadcast of the diagnostic data.

0 = CAN Diagnostic Data broadcast disabled
1 = CAN Diagnostic Data broadcast enabled (Default)

Please refer to section 6.8.4, 'Downloading Data using CAN', for more details on this feature.

CAN Inverter Enable Switch Active:
This parameter is used in CAN mode only.

1 = DIN1 digital input is taken into consideration and the inverter will only be enabled if both DIN1 and inverter command are active. If either one is inactive, the inverter will be disabled.

0 = DIN1 will have no effect on enabling or disabling the inverter (Default)

CAN Timeout:
This parameter sets the CAN Timeout time. The CAN Timeout time is the maximum time between CAN Command messages that will not generate a fault (if the CAN Command Message Active parameter is set to 1). This parameter is set as a multiple of 3 msec. For example, the default value is set to 333 which is equivalent to the actual timeout value of 999ms (333 x 3msec). This parameter delays setting the CAN Timeout fault for the amount of time it represents.

CAN Slave Cmd ID:
Command CAN address of the slaved controller. Must be in the range of 0x022 to 0x7FD (or set to 0 to disable). The address is set to a value that is 0x20 less than the ID offset set on the slaved controller. For example, if the slave controller has an ID offset of 0x1A0 then the CAN Slave Cmd ID is set to 0x1C0. See section on CAN Slave Mode in this manual.

CAN Slave Direction:
The slave controller can be commanded to the same direction command or opposite direction command based on this parameter. A value of 0 is the same direction, a value of 1 is the opposite direction.

CAN Fast Msg Rate / CAN Slow Msg Rate:
In certain installations it is necessary to change the rate at which the CAN Broadcast messages are sent by the controller. Previously messages had a fixed transmission rate of either 10ms (the fast messages) or 100ms (the slow messages). Two new EEPROM parameters have been added that allow the message rates to be changed.

The message rate is set by these two EEPROM parameters in terms of milliseconds (ms). The software loop that sends the CAN messages runs with a 3 ms period. The CAN messages will be sent out at the next increment of 3ms that matches the programmed broadcast rate. For example, if the rate is set to 10ms then the actual broadcast rate will be closer to 12ms. It is important to note that many CAN messages are being triggered to be sent at the same time. The CAN controller will send them out according to the availability of the CAN bus itself and the priority of the messages. The actual time between messages will vary in actual use.

If it is desired that the original CAN message transmission rates be used then the
`CAN_Fast_Msg_Rate_EEPROM_(ms)` parameter can be set to 10 and the
`CAN_Slow_Msg_Rate_EEPROM_(ms)` can be set to 100.

The two parameters should not be set to a time between messages lower than 3ms.

These two parameters can be used to disable the entire set of Fast or Slow CAN messages. If either message rate EEPROM parameter is set to 0 it will disable sending of that group of messages.

8.2 CAN Diagnostic Parameters Overview

Several CAN diagnostics are available through the RMS GUI.

RMS GUI Parameter	Description
CAN_Status_Bus_Off	0: Bus is active 1: CAN Bus has been turned off due to excessive errors. Note: The auto CAN bus restart feature has been enabled. If the CAN bus controller gets turned off it will automatically restart.
CAN_Status_Error_Passive	0: Bus is active 1: CAN Bus controller has gone to the passive state.
CAN_Status_Error_Warning	0: Bus is active 1: The number of CAN errors has reached a warning limit of 96.
CAN_Status_Last_Error_Code	Indicates the last reported error on the CAN bus controller: 0: No Error 1: Stuff Error 2: Form Error 3: Ack Error 4: Bit 1 Error 5: Bit 0 Error 6: CRC Error
CAN_Tx_Error_Counter	Count of errors in sending of CAN messages. The maximum count is 255 before the counter loops back to 0.
CAN_Rx_Error_Counter	Count of errors in receiving CAN messages, maximum count is 127. For each correctly received CAN message the error counter will count down towards 0.

8.3 CAN Format

The CAN protocol conforms to CAN 2.0A (11 bit identifiers) as well as CAN 2.0B (29 bit identifiers). CAN Messages are transmitted with a baud rate determined by the CAN Bit Rate EEPROM parameter. All messages have a data length code (DLC) of 8 bytes and follow little-endian format which implies that the least significant byte is stored at the lowest address. For example, if the command message is setup to turn the inverter on in CAN Speed mode with a speed command of 500 RPM the data bytes should look like this:

Data Byte 0	Data Byte 1	Data Byte 2	Data Byte 3	Data Byte 4	Data Byte 5	Data Byte 6	Data Byte 7
44	1	244	1	0	1	0	0

- Torque Command: Sent as a value in N.m. times 10.
  For example, 30 Nm should be entered as 300 = (1 x 256) + 44
  - Data Byte 0 = 44 (Low Byte)
  - Data Byte 1 = 1 (High Byte)
- Speed Command: Sent as value in RPM.
  For example, 500 RPM is entered as 500 = (1 x 256) + 244

  - Data Byte 2 = 244 (Low Byte)
  - Data Byte 3 = 1 (High Byte)
- Direction Command
  - Data Byte 4 = 0 (Clockwise = Reverse)
  - Data Byte 4 = 1 (Anticlockwise = Forward)
- Inverter Run Command
  - Data Byte 5 = 0 (Disable Inverter)
  - Data Byte 5 = 1 (Enable Inverter)

Each data frame is 89 bits long thus at 250 Kbps the bus can handle a maximum of 2808 messages per second.

8.4 CAN Data Formats

Each message contains one or more items. Each item is formatted and scaled per the definitions below:

Format	Description	Range
Temperature	Signed integer, actual temperature (in °C) times 10	−3276.8 to +3276.7 °C
Low Voltage	Signed integer, actual voltage (in Volts) times 100	−327.68 to +327.67 V
Torque	Signed integer, actual torque (in N.m) times 10	−3276.8 to +3276.7 N-m
High Voltage	Signed integer, actual volts (in Volts) times 10	−3276.8 to +3276.7 V
Current	Signed integer, actual current (in Amps) times 10	−3276.8 to +3276.7 A
Angle	Signed integer, actual angle (in degrees) times 10	0.0 to ±35.9 degrees
Angular Velocity	Signed integer, actual velocity (in RPM)	−32768 to +32767 rpm
Boolean	Unsigned byte, 1 = true/on, 0 = false/off	0 or 1
Frequency	Signed integer, actual frequency (in Hz) times 10	−3276.8 to +3276.7 Hz
Power	Signed integer, actual power (in kW) times 10	−3276.8 to +3276.7 kW
Time	Unsigned long integer or Unsigned integer. These are scaled values in counts that can be calculated by using their respective Scale Factors. For each Scale Factor, see the description column for that parameter.	NA
Flux	Signed integer, actual flux (in Webers) times 1000	−32.768 to +32.767 Webers
Proportional Gain	Unsigned integer, actual gain (unit-less) times 100 OR actual gain (unit-less) times 10000	0 to 655.35 0 to 6.5535
Integral Gain	Unsigned integer, actual gain (unit-less) times 10000	0 to 6.5535
Derivative Gain	Unsigned integer, actual gain (unit-less) times 100	0 to 655.35
Low-pass Filter Gain	Unsigned integer, actual gain (unit-less) times 10000	0 to 6.5535

Format	Description	Range
Per-unit value	These are scaled values that can be calculated by using their respective Scale Factors. For each Scale Factor, see the description column for that parameter.	NA
ADC Count	The value for ADC counts as read directly by the registers of a micro-controller.	0 to 4095
Pressure	Signed integer, actual pressure (in psi) times 10	−3276.8 to +3276.7 psi

8.5 CAN Database File

A CAN database file stores information for a given CAN network. For example, it includes information about CAN nodes, messages, and data bytes for each message. A CAN database file has a.dbc extension and can be used with several CAN data loggers such CANTrace, CANalyzer, CANoe, etc to log CAN data.

BorgWarner Portland provides a CAN database file. The file can be downloaded from the Support section of the Cascadia Motion website.

This file can also be edited by the user with his/her choice of a CAN database editor. For example, Kvaser's Database Editor can be used to create and edit the CAN database file. Kvaser's Database Editor can be found on the Kvaser web site, www.kvaser.com

8.6 CAN Messages

8.6.1 Broadcast Messages

Broadcast messages are sent by the controller continuously irrespective of VSM or CAN command mode. The table below shows the messages that are broadcast and the frequency at which they are sent. The addresses shown below are default addresses based on the default CAN ID Offset of 0x0A0. The CAN ID offset can be changed by using the EEPROM parameter for this variable. Using a different CAN ID offset would be specifically useful in the care where more than one controller is on the same CAN bus network. While setting CAN ID offset address for a controller, make sure that the address range for controllers does not contain overlapping addresses.

A parameter `CAN_ACTIVE_MSGS_EEPROM_(Lo_Word)` with parameter address 148 is defined to enable/disable individual CAN Broadcast Messages. Additionally, there is a parameter, `CAN_ACTIVE_MSGS_EEPROM_(Hi_Word)`, that controls various CAN mailboxes related to the Command message, Slave message, BMS, OBD2, and the U2C. This message should be kept as 0xFFFF (or 0xFFFE if implementing the High Speed Message).

Each bit in `CAN_ACTIVE_MSGS_EEPROM_(Lo_Word)` parameter represents a CAN Message broadcast status as follows:

0 = CAN Messages broadcast disabled
1 = CAN Message broadcast enabled

Slow = controlled by `CAN_Slow_Msg_Rate_EEPROM_(ms)`, typical 10Hz
Fast = controlled by `CAN_Fast_Msg_Rate_EEPROM_(ms)`, typical 100Hz

The controller broadcasts the following messages:

Address	Frequency	Content	CAN Active Messages (Low Word) Bit
0x0A0	Slow/10 Hz	Temperatures #1	0x0001
0x0A1	Slow/10 Hz	Temperatures #2	0x0002
0x0A2	Slow/10 Hz	Temperatures #3	0x0004
0x0A3	Fast/100 Hz	Analog Inputs Voltages	0x0008
0x0A4	Fast/100 Hz	Digital Input Status	0x0010
0x0A5	Fast/100 Hz	Motor Position Information	0x0020
0x0A6	Fast/100 Hz	Current Information	0x0040
0x0A7	Fast/100 Hz	Voltage Information	0x0080
0x0A8	Fast/100 Hz	Flux Information	0x0100
0x0A9	Slow/10 Hz	Internal Voltages	0x0200
0x0AA	Fast/100 Hz	Internal States	0x0400
0x0AB	Fast/100 Hz	Fault Codes	0x0800
0x0AC	Fast/100 Hz	Torque & Timer Information	0x1000
0x0AD	Fast/100 Hz	Modulation Index & Flux Weakening Output Information	0x2000
0x0AE	Slow/10 Hz	Firmware Information	0x4000
0x0AF	100 Hz (fixed)	Diagnostic Data	0x0800
			CAN Active Messages (High Word) Bit
0x0B0	333Hz (fixed)	High Speed Message, note bit of CAN Active Messages High Word must be set to 0 to activate. To activate this message the normal setting would be 0xFFFE.	0x0001
0x0B1	Fast/100 Hz	Torque Capability	0x0002

All of the above message addresses are in standard 11-bit format. The addresses in extended 29-bit and J1939 format are listed below (for the default CAN ID Offset of 0xA0):

Standard 11-bit Format	Extended 29-bit Format	J1939 Format
0x0A0	0x0A0 X	0x0CFFA001
0x0A1	0x0A1 X	0x0CFFA101
0x0A2	0x0A2 X	0x0CFFA201

Standard 11-bit Format	Extended 29-bit Format	J1939 Format
0x0A3	0x0A3 X	0x0CFFA301
0x0A4	0x0A4 X	0x0CFFA401
0x0A5	0x0A5 X	0x0CFFA501
0x0A6	0x0A6 X	0x0CFFA601
0x0A7	0x0A7 X	0x0CFFA701
0x0A8	0x0A8 X	0x0CFFA801
0x0A9	0x0A9 X	0x0CFFA901
0x0AA	0x0AA X	0x0CFFAA01
0x0AB	0x0AB X	0x0CFFAB01
0x0AC	0x0AC X	0x0CFFAC01
0x0AD	0x0AD X	0x0CFFAD01
0x0AE	0x0AE X	0x0CFFAE01
0x0AF	0x0AF X	0x0CFFAF01
0x0B0	0x0B0 X	0x0CFFB001
0x0B1	0x0B1 X	0x0CFFB101
0x0C0 Command Message	0x0C0 X	0x0CFFC001
0x0C1 Parameter Message	0x0C1 X	0x0CFFC101
0x0C2 Parameter Response	0x0C2 X	0x0CFFC201

Table 26: CAN Broadcast Messages

Extended format is often denoted by the letter 'X'. However, based on the CAN logger, this format may be represented differently. Please refer to the manual for the CAN logger that is used.

J1939 messages have the following fixed configuration:

- Priority = 3
- PDU Format = 0xFF
- PDU Specific = CAN ID Offset EEPROM
- Source Address = 1 (for both received and transmitted messages)

Enabling/Disabling Broadcast of CAN Messages:
As an example, in order to disable Temperature #1, #2 and #3 messages in the above table, the parameter command message should be configured as follows:

Data Byte 7 (MSB)	**Data Byte 6 (LSB)**	**Data Byte 5 (MSB)**	**Data Byte 4 (LSB)**	**Data Byte 3**	**Data Byte 2**	**Data Byte 1**	**Data Byte 0**
CAN Active Messages High Word		CAN Active Messages Low Word		Reserved	R/W Command	Parameter Address	
0xFF	0xFF	0xFF	0xF8	0x00	0x01	0x00	0x94

Data Byte 4 controls the following messages:
Bit 0: Temperature #1
Bit 1: Temperature #2
Bit 2: Temperature #3
Bit 3: Analog Input Voltages
Bit 4: Digital Input Status
Bit 5: Motor Position Information
Bit 6: Current Information
Bit 7: Voltage Information

In little-endian format, byte 4 can be looked at as: Bit 7-Bit 6...Bit 0. To enable all messages above, Byte 4 should be set to 0xFF (all bits set to 1). To disable temperature messages, Byte 4 should be set to 0xF8 (Bit 0, 1, and 2 are set to 0). To disable Motor position information, Byte 4 should be set to 0xDF (Bit 5 set to 0)

Data Byte 5 controls the following messages:
Bit 0: Flux Information
Bit 1: Internal Voltages
Bit 2: Internal States
Bit 3: Fault Codes
Bit 4: Torque & Timer Information
Bit 5: Modulation Index & Flux Weakening Output Information
Bit 6: Firmware Information
Bit 7: Diag Data

Data Byte 6 controls the following messages but the user should not disable any of these messages:
Bit 0: High Speed Message (1=off, 0=on)
Bit 1: Torque Capability
Bit 2: Not used
Bit 3: Not used
Bit 4: Not used
Bit 5: Not used
Bit 6: Slave Mode Command Message

Bit 7: BMS Command Message

Data Byte 7 controls the following messages but the user should not disable any of these messages:
Bit 0: OBD2 General Query
Bit 1: OBD2 Specific Query
Bit 2: OBD2 Response
Bit 3: U2C TX Message
Bit 4: U2C RX Message
Bit 5: Parameter Response Message
Bit 6: Parameter Command Message
Bit 7: CAN Command Message

Broadcast Message Definitions For a complete list of CAN broadcast messages please see section C, 'Appendix: CAN Broadcast Messages'.

8.7 Command Message

The Command Message is used to transmit data to the controller. This message is sent from a user-supplied external controller to the motor controller. The Control Message (0x0C0) is used to operate the controller via the CAN interface.

Table 28: 0x0C0: Command Message

Byte(s)	Name	Format	Description
0,1	Torque Command	Torque	Torque command used when in torque mode. When in Speed Mode the Torque Command values become a feed-forward to the Speed Regulator. (See section 7.1, 'Using Speed Mode' for more detail).
2,3	Speed Command	Angular Velocity	Speed command used when in speed mode.
4	Direction Command	Boolean	0 = Reverse 1 = Forward
5-Bit 0	Inverter Enable	Boolean	0 = Inverter Off 1 = Inverter On
5-Bit 1	Inverter Discharge	Boolean	0 = Disable Discharge 1 = Enable Discharge See section 7.3, 'Inverter Discharge' for more information.
5-Bit 2	Speed Mode Enable	Boolean	0 = Do not over-ride mode 1 = If controller is in torque mode then controller will change to speed mode. This is a mode over-ride bit that will change the mode from torque to speed only. It does not change the mode from speed to torque. See section 7.1 for more details.
6,7	Commanded Torque Limit	Torque	If set to 0, the default torque limits sets in the EEPROM parameters are used. If set to a positive number then the Motor and Regen torque limits are set to the torque value sent.

This message should be continuously broadcast at 500 milliseconds rate or faster. Of course, for most vehicle situations a message rate of 10 – 50 milliseconds provides better control of the vehicle. The Command Message is processed by the controller every 3ms. Sending the Command message at a rate faster than 3ms will not improve response. The message must be sent as an 8 byte message (DLC = 8).

If the Command Message is not received faster than the CAN TimeOut time and the Command Message Active Parameter is set to 1 then a CAN Command Message Lost fault will be generated.

When in CAN mode the Command messages should be sent to the controller before the inverter is powered on (or before the CAN Timeout time expires). If they are not then the Command message Lost fault will have to be cleared up on power up.

Note: Commanded Torque Limit feature was added in software version 1953. For previous versions of software these two bytes should be set to 0 and do not have any function.

Note: Commanded Torque will follow torque capability (motoring or regenerating) as of software version 652E for BorgWarner Portland catalog products (iM-225, iM 375, iM-425).

8.7.1 Inverter Enable Safety Options

Inverter Enable Lockout:
This feature is added so that the inverter cannot be accidentally enabled when first powered up. This feature requires that before sending out an Inverter Enable command, the user must send out an Inverter Disable command. Once the inverter sees a Disable command, the lockout is removed and controller can receive the Inverter Enable command.

Inverter Enable Safety Switch:
A new EEPROM parameter, CAN Inverter Enable Switch Active EEPROM, is added as a safety option.

Setting this parameter to 1 will take DIN1 digital input into consideration and the inverter will only be enabled if both DIN1 and inverter command are active. If DIN1 or Inverter Enable Command is inactive, the inverter will be disabled.

If CAN Inverter Enable Switch Active EEPROM is set to 0, DIN1 will have no effect on enabling or disabling the inverter.

Sudden Reversal of the Direction Command:
This safety feature keeps the user from changing the direction command while the inverter is enabled. If the direction command is changed suddenly when the inverter is still enabled, inverter is disabled without triggering any faults. Also, the lockout condition is set again which will force the user to send an Inverter Disable command before re-enabling it.

8.7.2 CAN Message Sequence Example

Here is an example of sending out torque commands to the inverter in 'CAN' mode with run mode required to be 'Torque'. These two EEPROM parameters can be set via GUI after powering up the inverter.

For the message sequence example described below, following assumptions hold true:

GUI EEPROM Parameter #	Default Value	Description
`Inv_Cmd_Mode_EEPROM`	0	CAN Mode
`Run_Mode_EEPROM`	0	Torque Mode
`CAN_ID_Offset_EEPROM`	0	Default CAN ID offset
`CAN_Timeout_(/3ms)_EEPROM`	333	1 second timeout period

8.7.3 CAN Message Sequence Example

When using the controller in CAN mode, it is important to make sure that commands are entered properly to move the motor in the intended direction. Following description provides details on speed

command, speed feedback, torque command, torque feedback and direction command and the possible outcome in each scenario.

CAN Speed Command:
The Speed Command is a signed number. If the speed command is positive then the direction will be the direction of the direction command bit. If the Speed Command is negative then the direction will be opposite of the direction of the direction command bit.

When using speed mode it is important that the torque limits are appropriate for regulating the speed. An unloaded motor will often require some amount of regen torque to keep the speed regulated. When operating with low or negative speeds the Regen Fade Speed EEPROM parameter can prevent regen torque. The Regen Fade Speed can be disabled by setting the EEPROM parameter to 0.

CAN Torque Command:
For a forward direction command:

- Positive torque command will give a positive torque feedback and is motoring for positive speed.
- Negative torque command will give a negative torque feedback and is regen for positive speed.
- Positive torque command will give a positive torque feedback and is regen for negative speed.
- *A negative torque command should not be allowed if already going negative speed.*

For a reverse direction command:

- Positive torque command will give a negative torque feedback & is motoring for negative speed.
- Negative torque command will give a positive torque feedback and is regen for negative speed.
- Positive torque command will give a negative torque feedback and is regen for positive speed.
- *A negative torque command should not be allowed if already going positive speed.*

8.8 Parameter Messages

The Parameter Messages (0x0C1 and 0x0C2) are used to read and write parameters in the controller. These parameters have many different functions. Some parameters are used to set non-volatile information (EEPROM data). Some are used to change functionality. Some are used to monitor various operating parameters that are not part of the broadcast messages.

To write a parameter use message 0x0C1 with byte #2 set to 1 (write). The controller will then respond with message 0x0C2 and if successful byte #2 will be set to 1.

To read a parameter use message 0x0C1 with byte #2 to set 0 (read). The controller will then respond with message 0x0C2 containing the requested data.

Both parameter messages contain 4 bytes for the data that is read or written. Some parameters will only occupy a single byte. If the data occupies less than 4 bytes it will be loaded into byte #4 first, followed by #5, and so on.

If the parameter address is not recognized then the controller parameter response message (0x0C2) will contain 0 in both bytes 0 and 1 of the return data.

8.8.1 Parameter Message Format

Table 29: 0x0C1: Read/Write Parameter Command-Sent to Motor Controller

Byte #	Name	Format	Description
0,1	Parameter Address	Unsigned Int	Each command is identified by a unique address. Refer to sections B.2, 'GUI Command Parameters' and A, 'Appendix: EEPROM Configuration Parameters' for each parameters address.
2	R/W Command	Boolean	0 = read, 1 = write
3	Reserved	NA	NA
4,5	Data	See Data Formats	Data should be entered as dictated in Data Formats
6,7	Reserved	NA	NA

Table 30: 0x0C2: Read/Write Parameter Response-Response to Motor Controller

Byte #	Name	Format	Description
0,1	Parameter Address	Unsigned Int	Each command is identified by a unique address. Refer to sections B.2, 'GUI Command Parameters' and A, 'Appendix: EEPROM Configuration Parameters' for each parameters address. Will return 0 if parameter address is not recognized.
2	Write Success	Boolean	0 = not written, 1 = success
3	Reserved	NA	NA
4,5	Data	See Data Formats	Response data is in the format dictated in Data Formats.
6,7	Reserved	NA	NA

8.8.2 Parameter Address Ranges

The parameters are categorized in several general categories. Some parameters are read-only, and some can be written and read.

Table 31: 0x0C2: Read/Write Parameter Response-Response to Motor Controller

Address Range	Category	Description
0 – 99	General	This address range contains general parameters for control and monitoring.
100 – 499	User EEPROM	This address range is for EEPROM variables. These can only be written when the controller is not operating the motor.

8.8.3 Command Parameters

Address	Name	Format	Description
1	Relay Command	Unsigned integer (0 – 65535)	0xAA00: Normal Run Mode 0x55**nn**: External Relay Control Mode. See Relay Output Options
10	Flux Command	Flux	Modify the flux command.
11	Resolver PWM Delay Command	Unsigned integer (0 – 6250)	This command is used in calibration of the timing of t It is used in determining the peak of the sine wave default value is 1100.
12	Gamma Adjust GUI Command	Degrees	This is a calibration parameter used in the alignme motor with the resolver. This command parameter equivalent GUI Command parameter.
20	Fault Clear	Boolean	Writing a 0 to this parameter clears any active faul through CAN in CAN as well as VSM mode.
21	Set PWM Frequency	Unsigned integer (6 – 24)	Setting the PWM frequency via this Parameter comm settings. The PWM frequency will revert back to the is cycled.
22	AIN Pull-Up Control	Unsigned integer (0 – 7)	On CM inverters that support control of the analog this Parameter will allow commanding the pull up Parameter data control each pull-up control (bit 0 = AIN3).
23	Shudder Compensation Gain Control	Unsigned integer (0 – 65535)	This parameter will allow the `Kp_Shudder` gain to be Compensation to be disabled and reenabled. If the P the Shudder Compensation will be disabled. If the P 0 then the Shudder Compensation will be enabled enabled via EEPROM) and the `Kp_Shudder` gain wil value divided by 100.
30	OBD2 Enable Command	Unsigned integer (0 – 8)	Enables OBD2 CAN mailbox with the ID offset sp bytes.
31	Diag Data Trigger	Unsigned integer (0 – 65535)	A non-zero value will trigger a Diag. Data download

Relay Command

This command is used to control relay outputs.

In order to control a relay, the inverter needs to be put into "External Relay Control" mode. This is achieved by setting byte 5 to 0x55. For byte 4, each bit corresponds to a relay. Bit 0 corresponds to relay 1 and bit 1 corresponds to relay 2. Similarly, bit 7 corresponds to relay 8 as shown in the table below. Note, the hardware does not currently support 8 relays.

Table 33: Relay Control Parameter Message Data

Byte 5	Byte 4
0x55	R4 R3 R2 R1

For example, if the user wants to turn on relay 3, he/she needs to set the data field of the parameter command messages to 0x5504. Similarly, if relay 1 and 2 need to be turned on, a command of 0x5503 must be sent.

Setting the data byte 5 to a value other than 0x55 will kick the inverter out of the "External Relay Control" mode and in to the "Normal Run" mode. The default value for normal run mode is 0xAA.

8.8.4 EEPROM Parameters

These Parameters are sent using the command message format described in 0x0C1 to modify the EEPROM parameters used by the controller. These parameters can only be written when the motor is not enabled. Each parameter will be stored in non-volatile memory at the time of programming. When the power is recycled the parameter will become effective and used by the inverter. Parameters highlighted in yellow will take immediate effect. All EEPROM parameters can also be modified through the GUI interface. Please see section A for a list of EEPROM parameters available.

8.9 OBD2 Messages

BorgWarner Portland (formerly Cascadia Motion) has added some OBD2 functionality to the CAN system.

The goal of this functionality is to allow the use of an OBD2 diagnostic tool. For example a CAN to Bluetooth adapter can be used with an Android application (e.g. Torque app).

BorgWarner Portland has successfully used the OBDLink LX Bluetooth adapter with the Torque app www.torque-bhp.com

The inverter is utilizing the vehicle specific portion of the OBD2 protocol. To access inverter information requires the use of Custom PID codes. All of the items shown in the Memory View of the RMS GUI can be accessed via the Custom PID codes.

To enable OBD2 set the following EEPROM parameter, `CAN_OBD2_Enable_EEPROM` > 0 and less than 8. The value used provides an offset. The default would be to set it to 1. The support for the Torque app (list of Custom PID codes for the app) provided by BorgWarner Portland assumes that the value is set to 1. If multiple inverters exist on the CAN bus (or other devices) the parameter can bet set to different values for each inverter to avoid a conflict between two inverters.

The inverter CAN setup must be setup to be compatible both with other devices on the CAN network (if any) as well as the CAN OBD2 device. Most OBD2 devices will support either 250k or 500k baud.

Each RMS GUI Memory View item is assigned a Custom PID. The PID used is the Memory View address plus 0x1000. So for example Feedback Speed is at memory view address 0x0097. The Custom PID to access the Feedback Speed is at 0x1097.

The OBD2 implementation uses the Mode 22 for the Custom PID.

The OBD2 protocol and the implementation of it have certain CAN ID addresses. The inverter will respond at the following CAN addresses:

Query at 0x7DF (general OBD2 receive address) Query at device specific (what is normally used) at 0x7DF + value of `CAN_OBD2_Enable_EEPROM` Respond at 0x7E7 + value of `CAN_OBD2_Enable_EEPROM`

The controller does not respond to DTC type enquiries. It only responds to the enhanced data queries (mode 22h).

BorgWarner Portland has created a spreadsheet of the available parameters for use with the Torque app, please contact support for more information.

8.10 Orion BMS Support

BorgWarner Portland has added support for the Orion BMS (or other BMS that transmits a compatible BMS message). The BMS CAN message will be used to limit the maximum torque commands to a level that approximates the amount of DC (battery) current that will be flowing.

The BMS CAN Message must be configured as follows:

CAN Message ID: 0x202 (514 decimal)-Can not be changed CAN Message Format: Byte 0 and Byte 1 contain the maximum discharge current in Amps Byte 2 and Byte 3 contain the maximum charge current in Amps

Data format is Little Endian (Intel)

Currents can be transmitted as positive or negative numbers the code will correctly interpret them as charging and discharging currents. In other words it ignores the sign.

For example, 0x02 0x01 0x04 0x02 0x00 0x00 0x00 0x00 will yield a discharge current limit of 258 A and charge current limit of 516 A.

It is up to the user to make sure that both the inverter and the BMS are at the same CAN baud rate.

The Orion BMS Support can be enabled by using the EEPROM Parameter, `CAN_BMS_Limit_Enable_EEPROM`. By setting this parameter to 1 the = firmware will begin accepting messages from the BMS.

To check to see if the inverter is receiving the BMS CAN message check the parameter `BMS_Limit_Msg_Status` in the GUI. If the result is 1 then the message is being received. If the controller stops receiving the CAN message from the BMS then the result will be 0 after 1 second has elapsed.

The `INV_BMS_Limiting_Motor_Torque` flag is also available in the Internal States Broadcast CAN message. Additionally in the Internal States message is a flag indicating when the maximum torque level is being limited by the BMS.

The BMS torque limiting function does not use the inverter measured or estimated DC current. It uses a simple equation to estimate the amount of torque:

Maximum torque = DC bus voltage * DC Current Limit / speed

In this equation the speed is mechanical speed in rad/s and torque is in Nm. The simple equation does not consider motor and inverter efficiency and thus will have accuracy issues that vary with operating conditions. To prevent the maximum torque from being excessively large the speed used in the above equation has a lower limit (525rpm for 10 pole motor)

If the inverter is receiving a charge/discharge current limit that would result in a reduced amount of available torque then it will begin to limit the maximum torque available from the accelerator or brake pedal (VSM mode).

If the vehicle is operated at lower speeds and the battery current limits are normal then the operator would not notice any effect. The maximum battery current draw would always be low even at maximum torque command because the motor speed is low.

If the motor speed is high and if the full accelerator application would exceed the battery current limit then the maximum point of the accelerator would be reduced. So even if the operator did not apply full throttle the operator would note that for a given amount of throttle the amount of torque produced is reduced.

In CAN mode it will also limit the torque command coming from the CAN command message. The BMS CAN Message will also limit the torque in speed mode.

8.11 CAN Slave Operation

In some cases multiple controllers and motors are used in an electric vehicle powertrain. In the case where two motors are used and both motors are connected in such a way that they should operate at the same commanded torque then it is possible to operate one of the controllers in a slave mode.

To enable Slave Operation the `CAN_Slave_Cmd_ID_EEPROM` parameter must be set to CAN Command Address of the Slave Controller. This address is the Slave Control CAN ID Offset plus 0x20 (32 decimal). The Slave CAN message will be sent at a fixed rate of approximately 10ms regardless of the broadcast rates set by the Fast and Slow CAN rate EEPROM parameters.

To disable the Slave CAN message from being sent set the `CAN_Slave_Cmd_ID_EEPROM` value to be 0.

In this mode one controller, the Master, that decides what the torque command is. The second controller is the Slave and will have the same torque command as the Master.

The Slave controller can only be set to Torque mode and should be an identical type of motor with the same motor setup (current, speed, torque limits). The Slave controller must be in CAN mode.

If the Master controller is in Torque mode then the Slave controller will be sent the same commanded torque. The Slave controller is sent the torque command that the master is processing. However, the

torque command does not contain any shudder compensation. If the Master controller is in speed mode then the Slave controller is sent the torque command that is being used in the internal speed regulator.

The Slave controller will still respond to speed and temperature limits for the motor that it is controlling. If the Orion BMS feature is enabled then the torque limit of the master will take into account that the slave controller is consuming the same amount of DC current.

In some installations that multiple motors the physical placement of the motors dictates that the motors although turning in the same physical direction are turning in opposite electrical directions. Thus from the point of view of the controller it is necessary to have a Slave command that is commanding the opposite direction. This can be handled using the `CAN_Slave_Dir_EEPROM` parameter. With a setting of 0 the Slave controller will receive the same commanded direction that the Master is operating in. With a setting of 1 the Slave controller will receive a direction command that is opposite of the Master controller.

The Master and Slave controller must be set to the same CAN baud rate and the same Extended message setting.

It is recommended that the `CAN_Command_Message_Active` feature be used to prevent an unsafe condition if the Slave controller were to stop receiving CAN messages.

To make sure that the Slave controller command message is being sent make sure that the `CAN_ACTIVE_MSGS_EEPROM_(Hi_Word)` has bit 22 set.

8.12 Rolling Counter

There could be failure modes where a CAN message can be retransmitted or lost for a time. It is possible that a simple timeout on reception is not enough to detect this error. To mitigate this issue a Rolling Counter is to be added to the CAN Command Message.

The Rolling Counter is a U4 number that will be incremented from 0 to 15 and repeat. The Rolling Counter value will be transmitted by the system controller to the inverter.

At power on the inverter will not know what the correct rolling counter value is. The first received CAN Command message will set the Rolling Counter current value to the received value plus one.

When a Command Message is received the Rolling Counter current value will be set to the received value plus one regardless of whether the received Rolling Counter value is correct or not.

The inverter will check the Rolling Counter for validity. If an error has occurred, then it will flag that an incorrect rolling counter value has been received.

Each time an error has been received then it will increment a Debounce Counter. The Debounce Counter is incremented by the Up Count Value. Each time that a correct Rolling Counter value is received the Debounce Counter will be decremented by a Down Count value. The Debounce Counter minimum value is limited to 0.

If the Debounce Counter reaches a Debounce Counter Max value then a fault will be triggered and the inverter will be disabled. The Debounce Counter will be held at the Debounce Counter Max value even if more faults occur.

The Fault triggered by this Debounce Counter will be the already existing CAN Command Message Lost fault, bit 11 of Run Faults.

When faults are cleared (through the GUI or CAN) the Rolling Counter and the Debounce Counter will be reset to their power on values.

The table below shows the EEPROM parameters associated with this feature.

EEPROM Parameter	GUI Addr	CAN Addr	Description
CAN_Debounce_Counter_Max_EEPROM	0x01B8	0x00EE	The value of the Debounce Counter at which the controller will declare a fault. If this EEPROM parameter is set to 0 it will disable the rolling counter function. (Default value is 20.)
CAN_Debounce_Up_Count_EEPROM	0x01B9	0x00EF	The number of counts that the Debounce Counter will increment when a Rolling Counter error has been flagged. (Default value is 5.)
CAN_Debounce_Down_Count_EEPROM	0x019A	0x00F0	The number of counts that the Debounce Counter will decrement when a correct Rolling Count message has been received.

CAN Command Message Modification The existing CAN Command message will be modified with the addition of a Rolling Counter value as shown in bold below.

Table 35: 0x0C0: Command Message

Byte #	Name
0,1	Torque Command
2,3	Speed Command
4	Direction Command
5-Bit 0	Inverter Enable
5-Bit 1	Inverter Discharge
5-Bit 2	Speed Mode Enable
5-Bit 4 – 7	Rolling Counter
6,7	Commanded Torque Limit

CAN Status Message Modification The Internal States CAN message provides information about the currently expected Rolling Counter value. The CAN message details are provided in table 64.

GUI Monitoring It is possible to monitor the current state of the Rolling Counter using the RMS GUI.

GUI Monitoring Parameter	Address	Description
CAN_Rolling_Count_Flag	0x09D	Will flag when a Rolling Counter error has occurred. Will clear when a correct message has been received.
CAN_Debounce_Count	0x09E	Current value of the Debounce Counter.

Examples

Below are several examples of what would happen with various received Rolling Counter values. The example assumes that the default EEPROM values are used.

Rolling Counter						
Msg1	Msg2	Msg3	Msg4	Msg5	Msg6	**Results**
0	1	2	3	4	5	Rolling Counter is good, Debounce Counter = 0, no fault.
0	2	3	4	5	6	One missed message, Debounce Counter reached 5, but then decremented back to 0, no fault.
0	2	3	4	5	7	Two missed (+10), 3 correct (-9) Debounce Counter = 1, no fault.
0	1	3	5	7	9	Four missed (+20), fault is declared.

A Appendix: EEPROM Configuration Parameters

A.1 Motor Configuration Parameters

EEPROM Parameter	GUI Addr	CAN Addr	Value Range	Description
Motor_Type_EEPROM	0x0119	0x0096	0 – 255	This parameter is used to select the motor that will be connected to the inverter. There are several motor specific manuals on the Cascadia Motion website that will provide information on the correct motor type number to be used.
Resolver_PWM_Delay_EEPROM	0x0118	0x0097	0 – 6250	This parameter adjusts a delay that is used to synchronize the resolver feedback to the PWM cycle. It is only used with motors that use resolvers. See section 4.1, 'Resolver Calibration' for more information on resolver calibration.
Gamma_Adjust_EEPROM_(Deg)_x_10	0x011A	0x0098	−3599 – 3599	This is a calibration parameter used in the alignment of the magnetic field of the motor with the resolver / SIN/COS encoder. This parameter is only used with PM type motors. See section 4.1, 'Resolver Calibration' for more information on resolver calibration.
Sin_Offset_EEPROM_(ADC_Counts)	0x0163	0x009C	0 – 4096	This feature is dependent on the hardware version of the PM unit. In some cases, the resolver sine and cosine outputs may require adjustments for improved signals. These offsets are added as ADC counts to calibrate the sin and cosine signals directly.
Cos_Offset_EEPROM_(ADC_Counts)	0x0164	0x009D	0 – 4096	This feature is dependent on the hardware version of the PM unit. In some cases, the resolver sine and cosine outputs may require adjustments for improved signals. These offsets are added as ADC counts to calibrate the sin and cosine signals directly.

BORGWARNER
7929 SW Burns Way
Wilsonville, OR 97070
www.CascadiaMotion.com
sales@cascadiamotion.com

EEPROM Parameter	GUI Addr	CAN Addr	Value Range	Description
DCLink_FW_Reg_Use_EEPROM	0x01C9	0x011E	0, 1	Set this parameter to 1 to enable Internal PM Motor DC Link Field Weakening for Catalog integrated units. Not available for non-catalog integrated units.
Self_Sense_Assist_Enabled_EEPROM	0x01C1	0x00FB	0, 1	This parameter should only be set if instructed by BorgWarner Portland (formerly Cascadia Motion) support.

Table 37: Motor Configuration Parameters

BORGWARNER
7929 SW Burns Way
Wilsonville, OR 97070
www.CascadiaMotion.com
sales@cascadiamotion.com

A.2 System Configuration Parameters

EEPROM Parameter	GUI Addr	CAN Addr	Value Range	Description
Serial_Number_EEPROM	0x0113	0x00AE	0 – 65535	Used for storage of the unit serial number.
Precharge_Bypassed_EEPROM	0x0115	0x008C	0 – 1	Set to 1: Setting this to a 1 will bypass the pre-charge sequence. When the drive is powered it will go directly to state "Wait State". Set to 0: Setting this to a 0 will enable the pre-charge sequence as described above. Default is 0.
Run_Mode_EEPROM	0x0116	0x008E	0 – 1	Set to 1: Setting this to a 1 will force the drive into speed control mode. This mode is only recommended for demonstration purposes when the motor is not connected to a high inertia load such as a vehicle. The Accelerator input will command a speed. Contact the factory for more information. For speed mode to operate correctly the Regen Torque Limit must be greater than 0. It should be set to at least 10% of the Motor Torque Limit. Set to 0: Setting this to a 0 will place the drive into torque mode. This is the normal operating mode for the drive. Default is 0.
Inv_Cmd_Mode_EEPROM	0x011B	0x008F	0 – 1	This parameter sets the operating mode of the inverter. Set to 0: Operate under control of the CAN bus. The CAN bus is responsible for enabling and disabling the motor. The brake, forward, and reverse switches are not used. Set to 1: Operate under control of accelerator input and switches (VSM Mode).

BORGWARNER
7929 SW Burns Way
Wilsonville, OR 97070
www.CascadiaMotion.com
sales@cascadiamotion.com

BORGWARNER
7929 SW Burns Way
Wilsonville, OR 97070
www.CascadiaMotion.com
sales@cascadiamotion.com

EEPROM Parameter	GUI Addr	CAN Addr	Value Range	Description
Key_Switch_Mode_EEPROM	0x012B	0x0095	$0-1$	This parameter provides alternate key switch modes. This allows different types of ignition for vehicles. 0 = Allows a simple on/off switch for powering up the inverter. 1 = Provides the functionality of a more traditional ignition switch with momentary START signal that powers up the inverter and keeps it powered until the ignition switch is turned off. This configuration must use the IGNITION and START inputs. Key Switch Mode is only effective in VSM Mode. CAN mode remains unaffected However, the parameter can be updated through both GUI and CAN.
Discharge_Enable_EEPROM	0x016D	0x00AD	0,1,2	Controls the Active Discharge process. Can be used to discharge the internal high voltage capacitors quicker than the passive discharge. See the Inverter Discharge Process Manual for more information. 0 = discharge disabled 1 = discharge is enabled without any faults 2 = discharge is enabled with faults

BORGWARNER
7929 SW Burns Way
Wilsonville, OR 97070
www.CascadiaMotion.com
sales@cascadiamotion.com

EEPROM Parameter	GUI Addr	CAN Addr	Value Range	Description
Relay_Output_State_EEPROM	0x012C	0x00AA	0 – 0xFFFF	This parameter controls all relays. To keep the compatibility with previous versions (prior to firmware version 1909), this parameter should be set to 0x000C which will maintain the functionality of OK and fault outputs. Bit 0: Relay 1-Precharge output Bit 1: Relay 2-Main Output Bit 2: Relay 3-OK Output Bit 3: Relay 4-Fault Output Bits 8 – 15 define potential alternative functions for each output. Currently the only alternative function is the Fault Output, if bit 11 is set the Fault output will stay on rather than blink the fault code. Please see the table below for detailed behavior of each relay.

Table 38: System Configuration Parameters

A.3 Continuous Variable PWM Operation

The following EEPROM settings are used to setup Continuous Variable PWM.

Table 39: Continuously Variable PWM Settings

EEPROM Parameter	GUI Addr.	CAN Addr.	Value Range	Description
PWM_Nominal_Freq_EEPROM_(kHz)	0x0150	0x00F1	6 – 24	This sets the nominal PWM frequency in kHz. The typical setting is 10[kHz].
PWM_Minimum_Variable_Freq_EEPROM_(kHz)	0x01BE	0x00F6	6 – 24	This sets the Minimum PWM used by the Continuously Variable PWM in [kHz]. The typical setting is 6[kHz].
PWM_Maximum_Variable_Freq_EEPROM_(kHz)	0x01BF	0x00F7	6 – 24	This sets the Maximum PWM used by the Continuously Variable PWM in [kHz]. The typical setting is 12 [kHz].
PWM_Stall_Freq_EEPROM_(kHz)	0x01BD	0x00F5	2 – 24	This sets the Maximum PWM used by the stall region of the Continuously Variable PWM in [kHz]. The typical setting is 2 [kHz].
PWM_Var_Freq_Mode_EEPROM	0x0179	0x00FA	0, 1, 2	This sets the PWM mode you wish to run. • 0 = Nominal PWM only, with stall region • 1 = Continuous Variable PWM, with stall region • 2 = Continuous Variable PWM, no stall region
PWM_Var_Dwell_Rate_EEPROM_(x3ms)	0x01C0	0x00F8	1 – 65535	This sets the minimum time the Continuous Variable PWM Method will stay at a given PWM frequency before transitioning. A typical setting is 10 for 0.03 [s].

BORGWARNER
7929 SW Burns Way
Wilsonville, OR 97070
www.CascadiaMotion.com
sales@cascadiamotion.com

BORGWARNER
7929 SW Burns Way
Wilsonville, OR 97070
www.CascadiaMotion.com
sales@cascadiamotion.com

Relay	Command Mode 0: CAN 1: VSM	Precharge Bypass 0: No 1: Yes	Output Relay Config 0: CAN 1: Normal	CAN Command 0: Off 1: On	Precharge States Active?	Output State Final 0: Off 1: On	Function	Description
1: Precharge	0	0	0	0	Y	0	CAN	Output under precharge control. Afterwards, goes to CAN control.
	0	0	0	1	Y	1	CAN	Output under precharge control. Afterwards, goes to CAN control.
	0	0	1	0	Y	1	Normal	Output under precharge control. Afterwards, goes to CAN control.
	0	0	1	1	Y	1	Normal	Output under precharge control. Afterwards, goes to CAN control.
	0	1	1	0	N	0	CAN	Output directly goes to CAN control.
	0	1	1	1	N	1	CAN	Output directly goes to CAN control.
	0	1	1	0	N	1	Normal	Output directly goes to output config.
	0	1	1	1	N	1	Normal	Output directly goes to output config.
	1	0	0	X	Y	1	Normal	Output under precharge control. Afterwards, goes to output config.
	1	0	1	X	Y	1	Normal	Output under precharge control. Afterwards, goes to output config.
	1	1	0	X	N	0	Normal	Output directly goes to output config.
	1	1	1	X	N	1	Normal	Output directly goes to output config.

Relay	Command Mode 0: CAN 1: VSM	Precharge Bypass 0: No 1: Yes	Output Relay Config 0: CAN 1: Normal	CAN Command 0: Off 1: On	Precharge States Active?	Output State Final 0: Off 1: On	Function	Description
2: Main	0	0	0	0	Y	1	Normal	Output will toggle during precharge. Afterwards, goes to CAN control.
	0	0	0	1	Y	1	Normal	Output will toggle during precharge. Afterwards, goes to CAN control.
	0	0	1	0	Y	1	Normal	Output will toggle during precharge. Afterwards, goes to CAN control.
	0	0	1	1	Y	1	Normal	Output will toggle during precharge. Afterwards, goes to CAN control.
	0	1	1	0	N	0	CAN	Output directly goes to CAN control.
	0	1	1	1	N	1	CAN	Output directly goes to CAN control.
	0	1	1	X	N	0	Normal	Output is on. No precharge function.
	0	1	1	X	N	1	Normal	Output is on. No precharge function.
	1	0	0	X	Y	1	Normal	Output under precharge control. Output is on after precharge.
	1	0	1	X	Y	1	Normal	Output under precharge control. Output is on after precharge.
	1	1	0	X	N	0	Normal	Output directly goes to output config.
	1	1	1	X	N	1	Normal	Output directly goes to output config.

BORGWARNER
7929 SW Burns Way
Wilsonville, OR 97070
www.CascadiaMotion.com
sales@cascadiamotion.com

Relay	Command Mode 0: CAN 1: VSM	Precharge Bypass 0: No 1: Yes	Output Relay Config 0: CAN 1: Normal	CAN Command 0: Off 1: On	Precharge States Active?	Output State Final 0: Off 1: On	Function	Description
3: Fault	0	X	0	0/1	X	0/1	CAN	CAN Control.
	1	X	1	X	X	1	Normal	Will blink a code based on existing fault.
	1	X	1	X	X	1	Alt	Solid on during existing fault.
4: OK	0	X	0	0/1	X	0/1	CAN	CAN Control.
	1	X	1	X	X	1	Normal	On to indicate 12V on the inverter.

Table 40: Relay Output Options

A.4 CAN Configuration Parameters

RMS GUI Parameter	GUI Addr.	CAN Addr.	Value Range	Description
CAN_ID_Offset_EEPROM	0x011D	0x008D	0x000 – 0x7C0	This parameter allows the user to choose their own set of contiguous CAN message identifiers starting with the value in CAN ID Offset. This offset covers a range of 0 – 0x7C0. The default offset is 0x0A0. The default range is 0x0A0 – 0x0CF. This feature is especially useful when there are more than one controller on the same CAN network. While setting base address for a controller, it must be made sure that the address range for controllers does not contain overlapping addresses.
CAN_Extended_Msg_ID_EEPROM	0x0131	0x0090	0, 1	This parameter allows switching between CAN standard and extended message identifiers. 0 = Standard CAN Messages 1 = Extended CAN Messages
CAN_J1939_Option_Active_EEPROM	0x0132	0x00AB	0, 1	This parameter allows switching between extended message identifiers with or without SAE J1939 format. 0 = J1939 formatting is not active 1 = J1939 formatting is active
CAN_OBD2_Enable_EEPROM	0x0152	0x00B1	0 – 7	0 = OBD2 Support is disabled 1 – 7 = OBD2 Support is enabled with the address offset defined by the value.
CAN_BMS_Limit_Enable_EEPROM	0x017F	0x00B2	0, 1	0 = BMS CAN Message Torque Limiting is disabled. 1 = BMS CAN Message Torque Limiting is enabled.

BORGWARNER
7929 SW Burns Way
Wilsonville, OR 97070
www.CascadiaMotion.com
sales@cascadiamotion.com

RMS GUI Parameter	GUI Addr.	CAN Addr.	Value Range	Description
CAN_Command_Message_Active	0x011F	0x0092	0, 1	The parameter enables CAN Command Timeout feature. The CAN Timeout feature if enabled will generate a fault if the CAN Command Message is not received within the time set by CAN Timeout parameter. It is recommended that this feature be enabled. 0 = The CAN Timeout feature is disabled. 1 = The CAN Timeout feature is enabled.
CAN_Bit_Rate_EEPROM_(kbps)	0x0120	0x0093	125 – 1000	The CAN Bus bit rate can be changed using this parameter. However, changing this parameter requires a power reset on controller since bus speed is setup only at the initialization of CAN modules. Also, The parameter is restricted to valid baud rates. The 4 options for valid baud rate are: 125 = 125 Kbps 250 = 250 Kbps 500 = 500 Kbps 1000 = 1 Mbps
CAN_ACTIVE_MSGS_EEPROM_(Lo_Word)	0x0129	0x0094	0x0000 – 0xFFFF	This parameter is used to enable or disable CAN Broadcast Messages. Each bit represents a CAN Message broadcast status as follows: 0 = CAN Messages broadcast disabled 1 = CAN Message broadcast enabled (Default) Please refer to the table of CAN Broadcast Messages in section C, 'Appendix: CAN Broadcast Messages' for details on how to enable/disable each message.
CAN_ACTIVE_MSGS_EEPROM_(Hi_Word)	0x012A	0x0094	0x0000 – 0xFFFF	This parameter can be used to disable individual CAN mailboxes. This parameter should not normally be used and the default value of 0xFFFF should be used.

BORGWARNER
7929 SW Burns Way
Wilsonville, OR 97070
www.CascadiaMotion.com
sales@cascadiamotion.com

RMS GUI Parameter	GUI Addr.	CAN Addr.	Value Range	Description
CAN_Fast_Msg_Rate_EEPROM_(ms)	0x0180	0x00EB	0 – 65535	Sets the broadcast rate in ms of messages that are in the Fast group. If set to 0 will disable all Fast group messages.
CAN_Slow_Msg_Rate_EEPROM_(ms)	0x0181	0x00EC	0 – 65535	Sets the broadcast rate in ms of messages that are in the Slow group. If set to 0 will disable all Slow group messages.
CAN_Diag_Data_Tx_Active_EEPROM	0x0160	0x009E	0, 1	This parameter is used to enable/disable the broadcast of the diagnostic data. 0 = CAN Diagnostic Data broadcast disabled 1 = CAN Diagnostic Data broadcast enabled (Default) Please refer to section 6.8, 'Download Diagnostic Data', for more details on this feature.
CAN_Inv_Enab_Switch_Active_EEPROM	0x0170	0x009F	0, 1	1 = DIN1 digital input is taken into consideration and the inverter will only be enabled if both DIN1 and inverter command are active. If either one is inactive, the inverter will be disabled. 0 = DIN1 will have no effect on enabling or disabling the inverter (Default)
CAN_Timeout_(/3ms)_EEPROM	0x016F	0x00AC	0– 65535	This parameter sets how long before the CAN timeout error is set. The timeout is only active if the CAN Command Message Active is set to 1. The time is set in counts of 3ms. So for example setting a value of 333 will give a timeout time of 999 ms.
CAN_Slave_Cmd_ID_EEPROM	0x0177	0x00E9	0– 0x7FD	0 = disable slave mode 0x22 thru 0x7FD enables Slave mode CAN message output.
CAN_Slave_Dir_EEPROM	0x0178	0x00EA	0, 1	0 = Direction command of Slave controller is the same as the Master. 1 = Direction command of Slave controller is the opposite of the Master.

BORGWARNER
7929 SW Burns Way
Wilsonville, OR 97070
www.CascadiaMotion.com
sales@cascadiamotion.com

RMS GUI Parameter	GUI Addr.	CAN Addr.	Value Range	Description
CAN_Debounce_Counter_Max_EEPROM	0x01B8	0x00EE	0–65535	The value of the Debounce Counter at which the controller will declare a fault. If this EEPROM parameter is set to 0 it will disable the rolling counter function.
CAN_Debounce_Up_Count_EEPROM	0x01B9	0x00EF	0–65535	The number of counts that the Debounce Counter will increment when a Rolling Counter error has been flagged.
CAN_Debounce_Up_Down_EEPROM	0x01BA	0x00F0	0–65535	The number of counts that the Debounce Counter will decrement when a correct Rolling Count message has been received.

Table 41: CAN Configuration Parameters

BORGWARNER
7929 SW Burns Way
Wilsonville, OR 97070
www.CascadiaMotion.com
sales@cascadiamotion.com

A.5 Current Configuration Parameters

EEPROM Parameter	GUI Addr.	CAN Addr.	Value Range	Description
IQ_Limit_EEPROM_(A)_x_10	0x0101	0x0064	See motor setup manual	This parameter sets the Q-axis current limit. The Q-axis current is an industry term for the torque producing portion of the motor current. The current level is set in terms of peak amps. For example, to set a level of 400 amps peak use a parameter setting of 4000.
ID_Limit_EEPROM_(A)_x_10	0x0102	0x0065	See motor setup manual	This parameter sets the D-axis current limit. The D-axis current is an industry term for the flux producing portion of the motor current. For induction motors it is necessary to provide flux current to the motor. For PM motors the flux is provided by the magnets. However, at high speeds it is necessary to weaken the flux. D-axis current will be used with PM motors to reduce the magnet flux. The current level is set in terms of peak amps. For example, to set a level of 400 amps peak use a parameter setting of 4000.

Table 42: Current Parameters

The total motor current is the vector determined by the Q-axis current and the D-axis current. So the total current is the square root of $IQ^2 + ID^2$. The result in is Apk, if you want the RMS value of the current then divide the total current calculated by sqrt (2).

BORGWARNER
7929 SW Burns Way
Wilsonville, OR 97070
www.CascadiaMotion.com
sales@cascadiamotion.com

A.6 Voltage & Flux Parameters

EEPROM Parameter	GUI Addr.	CAN Addr.	Value Range	Description
DC_Volt_Limit_EEPROM_(V)_x_10	0x0104	0x0066	0 – 10000	This parameter is used to implement a DC Bus voltage limiting feature. The parameter should be set higher than the maximum battery voltage. It is not recommended that the system use this feature to protect the battery from over-voltage. This parameter should be set to a voltage higher than the the maximum voltage of the battery.
DC_Volt_Hyst_EEPROM_(V)_x_10	0x0105	0x0067	300	Used with the above parameter.
DC_UnderVolt_Thresh_EEPROM_(V)_x_10	0x0117	0x0068	0 – 10000	This is the under-voltage fault threshold voltage. If it is desired that the drive does not detect under-voltage faults the value can be set to 0.
Veh_Flux_EEPROM_(Wb)_x_1000	0x0100	0x006A	0 – 30000	This parameter sets the back EMF (flux) constant for the motor. It will automatically default to the correct value when the motor type is changed. Most of the time, the default value is sufficient and this value seldom needs to be changed. The flux value is set in units of Webers. For example to set a value of 0.1 Webers set the parameter to 100.

Table 43: Current Parameters

BORGWARNER
7929 SW Burns Way
Wilsonville, OR 97070
www.CascadiaMotion.com
sales@cascadiamotion.com

A.7 Temperature Parameters

RMS GUI Parameter	GUI Addr.	CAN Addr.	Value Range	Description
Inv_OverTemp_Limit_EEPROM_(C)_x_10	0x0106	0x0070	−40 – 125° C	This parameter sets the Inverter temperature limit. The temperature is measured from three sensors that are mounted inside the power module. Generally the module temperature will be about 0 – 20° C higher than the water temperature. The temperature is set is degrees Celsius times 10 (85° C is set as 850). If the temperature exceeds this value then the inverter will turn off and declare a fault.
Mtr_OverTemp_Limit_EEPROM_(C)_x_10	0x0121	0x0071	−40 – 250° C	This parameter sets the Motor temperature limit (if the motor has a temperature sensor). The temperature is set is degrees Celsius times 10 (150 ° C is set as 1500). If the temperature exceeds this value then the inverter will turn off and declare a fault.
RTD_Selection_EEPROM_(BITS_1_0)	0x014A	0x00CB	0 – 3	For Gen 5 inverters this EEPROM parameter configures the type of RTD that is connected to the two different RTD inputs (valid parameters values are 0 thru 3). 0: RTD1 = PT1000 RTD2 = PT100 1: RTD1 = PT100 RTD2 = PT1000 2: RTD1 = PT1000 RTD2 = PT1000 3: RTD1 = PT100 RTD2 = PT100

Table 44: Temperature Parameters

BORGWARNER
7929 SW Burns Way
Wilsonville, OR 97070
www.CascadiaMotion.com
sales@cascadiamotion.com

BORGWARNER
7929 SW Burns Way
Wilsonville, OR 97070
www.CascadiaMotion.com
sales@cascadiamotion.com

A.8 Accelerator & Torque Parameters

The accelerator pedal input provides a torque command to the motor. The graph below details the relationship between the accelerator input voltage and the torque command:

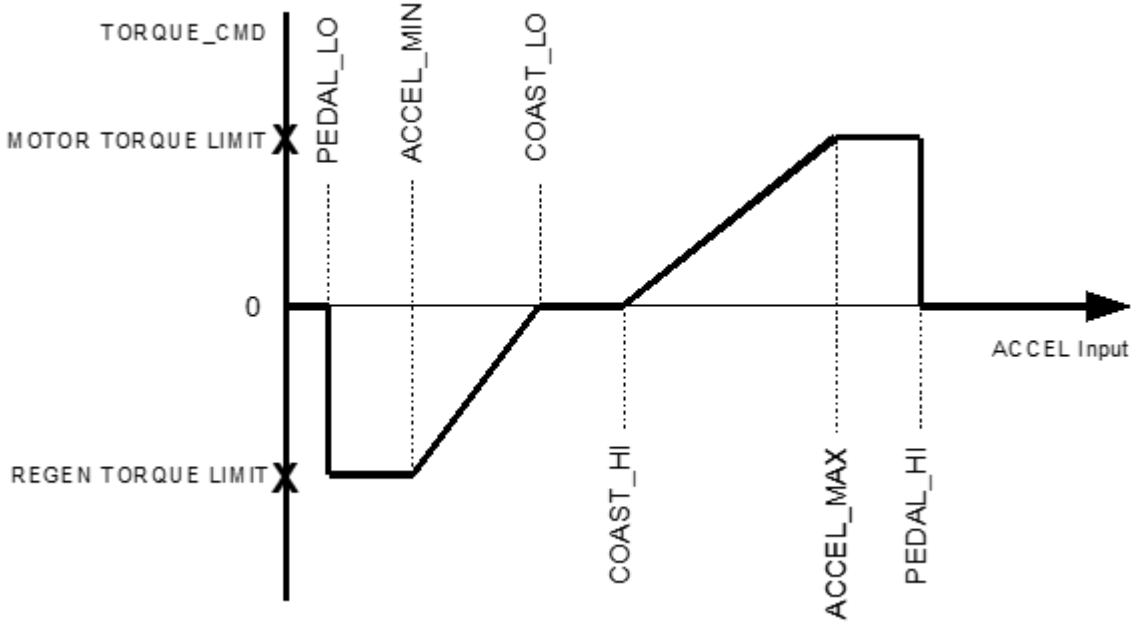


Figure 22: Pedal Functionality

Below is a list of the parameters that effect how the accelerator input works. The accelerator input has a range of 0 to 500. This corresponds to a physical range of 0 to 5.00 volts on the input. The parameters are designed for a pedal that provides a low input voltage when the pedal is released and a higher voltage as the pedal is pressed. If the vehicle has a pedal that operates in the opposite direction use the ACCEL PEDAL FLIPPED parameter as described below.

For initial setup and calibration, the accel pedal voltage can either be monitored by a volt meter, or it can be monitored by the GUI software over the serial port.

BORGWARNER
7929 SW Burns Way
Wilsonville, OR 97070
www.CascadiaMotion.com
sales@cascadiamotion.com

EEPROM Parameter	GUI Addr.	CAN Addr.	Value Range	Description
Accel_Pedal_Flipped_EEPROM	0x0114	0x0084	0 or 1	If the pedal increases in voltage as it is pressed use a value of 0 (not flipped). If the pedal decreases in voltage as it is pressed use a value of 1 (flipped). When this parameter is 1, the pedal voltage will first be processed by the equation: newpedalvoltage = 5.00 − old pedal voltage. Thus will make the pedal act the same as a pedal that normally increases in voltage.
Pedal_Lo_EEPROM_(V)_x_100	0x0107	0x0078	1 – 500	For accelerator inputs less than this value the torque command is zero. This value should be set to a value that is lower than the lowest possible accelerator position, but higher than zero. If the accelerator input were to be shorted to ground the desired torque command is zero.
Accel_Min_EEPROM_(V)_x_100	0x0108	0x0079	1 – 500	For accelerator inputs between PEDAL_LO and ACCEL_MIN the torque command is set to a constant value of REGEN TORQUE LIMIT. Depending on the desired characteristics of the vehicle this range could be very small.
Coast_Lo_EEPROM_(V)_x_100	0x0109	0x007A	1 – 500	For accelerator inputs between ACCEL_MIN and COAST_LO the torque command is linearly from REGEN TORQUE LIMIT to zero. If desired this range allows the operator to control the amount of regen torque.
Coast_Hi_EEPROM_(V)_x_100	0x010A	0x007B	1 – 500	For the range between COAST_LO and COAST_HI the torque command is zero. Normally this range would be fairly small.
Accel_Max_EEPROM_(V)_x_100	0x010B	0x007C	1 – 500	For the range between COAST_HI and ACCEL_MAX the torque is linearly increased from zero to the MOTOR TORQUE LIMIT. This would be the normal driving range.

EEPROM Parameter	GUI Addr.	CAN Addr.	Value Range	Description
Pedal_Hi_EEPROM_(V)_x_100	0x010C	0x007D	1 – 500	For the range between ACCEL_MAX and PEDAL_HI the torque command is held constant at MOTOR TORQUE LIMIT. PEDAL_HI should be set above the normal range of pedal motion, but below 500.
Motor_Torque_Limit_EEPROM_(Nm)_x_10	0x0110	0x0081	See Motor Manual	This parameter sets the maximum torque that can be commanded by the controller in motoring mode. It is active in both VSM mode and CAN mode. However, if the current limit of the drive is reached before the torque command has been achieved the controller will limit on the current first. If this happens the operator will feel an additional amount of unused pedal range at the top end. The motor torque limit should always be set at a torque that would be lower than or equal to the current limit. Torque value is set in Nm times 10. For example to set 300 Nm use a value of 3000.
Regen_Torque_Limit_EEPROM_(Nm)_x_10	0x0111	0x0082	See Motor Manual	This parameter sets the maximum regen torque that can be commanded by the controller. It is active in both VSM mode and CAN mode. In VSM mode this parameter is the maximum regen torque that is commanded when the pedal is fully released. Torque value is set in Nm times 10. For example to set 300 Nm use a value of 3000.
Braking_Torque_Limit_EEPROM_(Nm)_x_10	0x0112	0x0083	See Motor Manual	This parameter sets the amount of the torque applied when the brake input is active in VSM mode. It does not have any effect when in CAN control mode. Torque value is set in Nm times 10. For example to set 300 Nm use a value of 3000.

BORGWARNER
7929 SW Burns Way
Wilsonville, OR 97070
www.CascadiaMotion.com
sales@cascadiamotion.com

BORGWARNER
7929 SW Burns Way
Wilsonville, OR 97070
www.CascadiaMotion.com
sales@cascadiamotion.com

EEPROM Parameter	GUI Addr.	CAN Addr.	Value Range	Description
Torque_Rate_Limit_EEPROM_(Nm)_x_10	0x014B	0x00A8	0.1 – 250.0 Nm	This parameter adjusts how quickly the torque command is allowed to change. The parameter is set in terms of torque increment every 3 milliseconds. Torque value is set in Nm times 10. The maximum setting of this parameter (250Nm) could be higher than the motor could actually achieve.
Full_Torque_Temp_EEPROM_(C)_x_10	0x015D	0x0073	−40 – 250° C	Below this temperature threshold where the full torque is available. As the motor temperature is increased from Full_Torque_Temp_EEPROM_(C)_x_10 to Zero_Torque_Temp_EEPROM_(C)_x_10, the allowed torque capability is linearly decreased. This parameters should be less than Zero_Torque_Temp_EEPROM_(C)_x_10 which should be less than Mtr_OverTemp_Limit_EEPROM_(C)_x_10.
Zero_Torque_Temp_EEPROM_(C)_x_10	0x015E	0x0072	−40 – 250° C	Temperature threshold where the torque is zero. This value should be less than Mtr_OverTemp_Limit_EEPROM_(C)_x_10.

Table 45: VSM Parameters

The Motor_Torque_Limit_EEPROM_(Nm)_x_10 and Regen_Torque_Limit_EEPROM_(Nm)_x_10 parameters set the maximum value of commanded torque. They will be modified internally based on motor speed as the motor cannot put out full torque over the entire speed range.

The accelerator should be designed so that in its normal range of operation it is greater than 0 volts and less than 5 volts. The parameters Pedal_Lo_EEPROM_(V)_x_100 and Pedal_Hi_EEPROM_(V)_x_100 should be set so that if the input goes to 0 or 5 the torque command goes to zero.

These parameters allow the controller to be setup to command a pedal off amount of regen torque. This regen torque would mimic the engine compression feel that vehicles often have.

Example Setup:

BORGWARNER
7929 SW Burns Way
Wilsonville, OR 97070
www.CascadiaMotion.com
sales@cascadiamotion.com

As an example let's assume that assume that the accelerator input comes from a potentiometer. That is, the one end of the pot is connected to AGND. The other end is connected to XDCR_PWR (+5V), and the wiper is connected to AIN1. This setup is shown in the example application schematic.

First we need to determine the range of travel of this potentiometer. With the controller 12V turned on measure the voltage on the wiper of the pot (AIN1). Note how the voltage changes as the pedal is pushed and released. If the voltage increases as the pedal is pressed then the `Accel_Pedal_Flipped_EEPROM` parameter needs to be set to 0. If the voltage decreases then the `Accel_Pedal_Flipped_EEPROM` parameter needs to be set to 1. Whenever the parameter is set to 1 all of the other parameter settings must be calculated as follows $parameter = 500 - actualvoltage * 100$. For example if you desire a parameter to be set to 1.20 volts then the actual parameter setting will be $500 - 1.20 * 100 = 380$.

For this example we will assume that the voltage increases as the pedal is pressed. So `Accel_Pedal_Flipped_EEPROM` will be set to 0.

First measure the wiper voltage (AIN1) when the pedal is in the fully off position. For this example let's assume the measured value is 0.83 volts.

The `Pedal_Lo_EEPROM_(V)_x_100` parameter should be set to a value that is lower than this measured value. In this example let's set it to 0.40 volts (this corresponds to `Pedal_Lo_EEPROM_(V)_x_100` = 40). We want to set the parameter `Accel_Min_EEPROM_(V)_x_100` to be equal to this measured value (`Accel_Min_EEPROM_(V)_x_100` = 83). This will cause the torque to start increasing as soon as the pedal begins to be pressed.

Now measure the value of the wiper voltage (AIN1) when the pedal is fully pressed. For this example let's assume that measured value is 4.75 volts.

When the pedal is fully pressed we want to be commanding full torque so set the `Accel_Max_EEPROM_(V)_x_100` parameter to this measured value (`Accel_Max_EEPROM_(V)_x_100` = 475).

The `Pedal_Hi_EEPROM_(V)_x_100` parameter should be set to a value that is above this measured value but less than 5.00 volts. In this example let's set the value to 4.90 volts (`Pedal_Hi_EEPROM_(V)_x_100` = 490).

The `Coast_Lo_EEPROM_(V)_x_100` and `Coast_Hi_EEPROM_(V)_x_100` parameters define a range of pedal position where the torque command will be zero. For this example we'll define this range to be fairly narrow and with the pedal only slightly depressed. So we will set `Coast_Lo_EEPROM_(V)_x_100` to 1.10 volts (110) and `Coast_Hi_EEPROM_(V)_x_100` to 1.20 volts (120).

BORGWARNER
7929 SW Burns Way
Wilsonville, OR 97070
www.CascadiaMotion.com
sales@cascadiamotion.com

Motor Over-temperature Torque Reduction

This feature allows the Torque Capability to take motor temperature into consideration. Figure 23 shows the relationship between Torque Capability and Motor Speed. Based on the calculation of the slope and offset of the line from `Full_Torque_Temp_EEPROM_(C)_x_10` to `Zero_Torque_Temp_EEPROM_(C)_x_10`, the new torque capability is reduced by a factor of (slope * Motor Temperature + offset). `Zero_Torque_Temp_EEPROM_(C)_x_10` should be less than `Zero_Torque_Temp_EEPROM_(C)_x_10`, which should be less than `Mtr_OverTemp_Limit_EEPROM_(C)_x_10`.

Figure 23: Motor Over-temperature Torque Reduction

BORGWARNER
7929 SW Burns Way
Wilsonville, OR 97070
www.CascadiaMotion.com
sales@cascadiamotion.com

A.9 Speed Configuration Parameters

Torque Capability Curve is a function of Motor Speed, a feedback parameter from the Motor Control. Figure 24 shows the relationship between Torque Capability and Motor Speed:

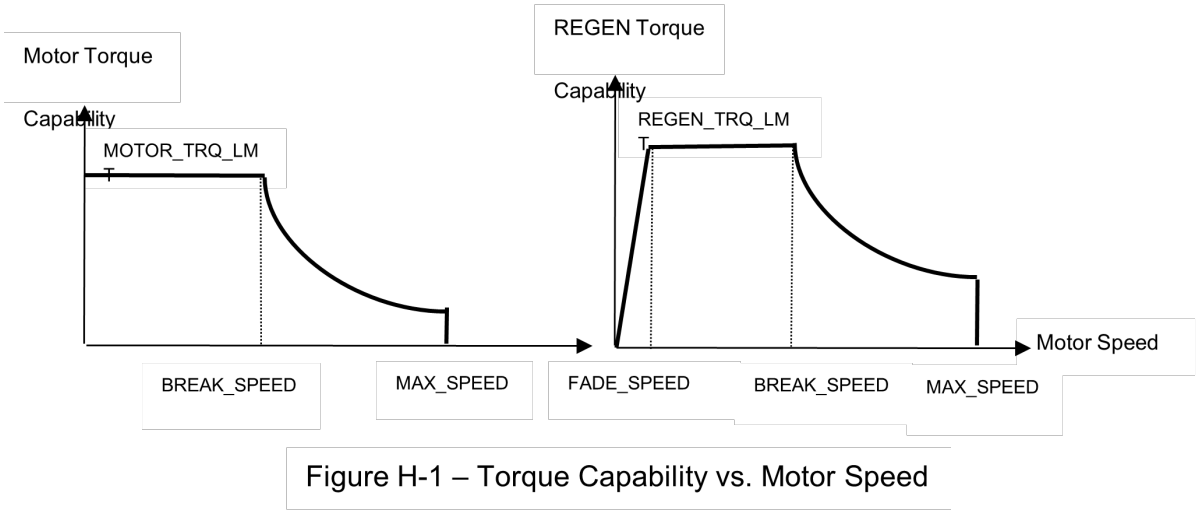


Figure 24: Torque Capability vs. Motor Speed

There are two types of Torque Capability curves, Motor Torque Capability and Regen Torque Capability. The two quantities MOTOR_TRQ_LMT and REGEN_TRQ_LMT (see previous section) define the maximum values for these curves.

When motors exceed a certain speed the amount of torque that they can produce will drop. The BREAK_SPEED parameter defines a curve that represents this drop in torque. The curve is defined BREAK_SPEED divided by actual speed time the torque limit.

The purpose of this curve is to reduce the torque limit so that the accel input does not try and command torque that the motor cannot deliver. If CAN mode is used or other torque limit means the BREAK_SPEED parameter can be set equal to MAX_SPEED to eliminate this effect.

The following table lists the calibration parameters that pertain to the above graphs. The values of these parameters come from the EEPROM and are set via the RMS GUI software.

BORGWARNER
7929 SW Burns Way
Wilsonville, OR 97070
www.CascadiaMotion.com
sales@cascadiamotion.com

EEPROM Parameter	GUI Addr.	CAN Addr.	Value Range	Description
Motor_Overspeed_EEPROM_(RPM)	0x0103	0x006F	1 – 30000 RPM	This parameter dictates when a over-speed fault is thrown. Recommended to be set higher than Max_Speed_EEPROM_(RPM).
Max_Speed_EEPROM_(RPM)	0x010F	0x0080	1 – 30000 RPM	This parameter sets the maximum allowable speed. If the speed is above this value the torque command will be reduced to zero. (Default value: 10,000 RPM)
Regen_Fade_Speed_EEPROM_(RPM)	0x010D	0x007E	1 – 30000 RPM	This parameter sets at which the amount of regen torque available is reduced. (Default value: 200 RPM). This parameter also controls whether regen torque can be applied when the motor is spinning in the opposite direction of the commanded direction. If the parameter is set to a value > 0 then no regen torque commands are allowed when the motor is spinning in the opposite direction of the commanded direction. If the parameter is set to 0 then regen torque commands can be given at all times.
Break_Speed_EEPROM_(RPM)	0x010E	0x007F	1 – 30000 RPM	This parameter sets the speed at which the maximum torque command is reduced to compensate for a reduction of available torque due to field weakening. (Default value: 3000 RPM)
Speed_Rate_Limit_EEPROM_(RPM/sec)	0x014E	0x00A9	100– 5100 RPM/sec	This parameter adjusts how quickly the speed command is allowed to change. The parameter is set in terms of speed increment every second. Default value is set to 100 RPM/sec. This parameter has no effect on torque mode operation.

Table 46: Speed Parameters

BORGWARNER
7929 SW Burns Way
Wilsonville, OR 97070
www.CascadiaMotion.com
sales@cascadiamotion.com

Max Speed Torque Reduction

This feature allows the Torque Capability to take maximum speed into consideration. Figure 25 shows the relationship between Torque Capability and Motor Speed. When the speed goes above the Max Speed, it begins a linear reduction in the torque towards zero. The slope of the reduction is such that at (Max Speed * 1.02) the torque is zero. The torque slope would be calculated based on the available torque at max speed. This reduction of torque is applied to motoring as well as regen.

Figure 25: Max Speed Torque Reduction

BORGWARNER
7929 SW Burns Way
Wilsonville, OR 97070
www.CascadiaMotion.com
sales@cascadiamotion.com

A.10 PID Regulator Parameters

The motor controller is some instances use a torque regulator and a speed regulator. For non IPM type motors the torque regulator is used all of the time. The speed regulator is only used if the controller is in Speed Mode (see Run Mode parameter). The regulators are both based on the classic PID architecture. Each of these regulators has 4 gain values associated with them. They are:

- Kp: Proportional Gain
- Ki: Integral Gain
- Kd: Derivative Gain
- Klp: Low Pass Filter Gain

Generally it is not necessary to adjust these gains. In some instances if the torque regulator seems unstable it may be necessary to adjust the value.

RMS GUI Parameter	GUI Addr.	CAN Addr.	Value Range	Description
Kp_Torque_EEPROM_x_10000	0x012D	0x00A4	0 – 65535	Torque Regulator proportional gain. This is a times 10000 value. Multiply the value within the valid range by 10000 before programming it using RMS GUI application.
Ki_Torque_EEPROM_x_10000	0x012E	0x00A5	0 – 65535	Torque Regulator integral gain. This is a times 10000 value. Multiply the value within the valid range by 10000 before programming it using RMS GUI application. Setting the Ki Torque value to 0 will disable the torque regulator and make the Iq command currently be directly calculated from the torque command.
Kd_Torque_EEPROM_x_100	0x012F	0x00A6	0 – 65535	Torque regulator derivative gain. This is a times 100 value. Multiply the value within the valid range by 100 before programming it using RMS GUI application.
Kp_Torque_EEPROM_x_10000	0x0130	0x00A7	0 – 65535	Torque regulator low pass filter gain. This is a times 10000 value. Multiply the value within the valid range by 10000 before programming it using RMS GUI application.

RMS GUI Parameter	GUI Addr.	CAN Addr.	Value Range	Description
Kp_Speed_EEPROM_x_100	0x0122	0x00A0	0 – 65535	Speed regulator proportional gain. This is a times 100 value. Multiply the value within the valid range by 100 before programming it using RMS GUI application.
Ki_Speed_EEPROM_x_10000	0x0123	0x00A1	0 – 65535	Speed regulator integral gain. This is a times 10000 value. Multiply the value within the valid range by 10000 before programming it using RMS GUI application.
Kd_Speed_EEPROM_x_100	0x0124	0x00A2	0 – 65535	Speed regulator derivative gain. This is a times 100 value. Multiply the value within the valid range by 100 before programming it using RMS GUI application.
Klp_Speed_EEPROM_x_10000	0x0125	0x00A3	0 – 65535	Speed regulator low pass gain. This is a times 10000 value. Multiply the value within the valid range by 10000 before programming it using RMS GUI application.

Table 47: Speed Parameters

BORGWARNER
7929 SW Burns Way
Wilsonville, OR 97070
www.CascadiaMotion.com
sales@cascadiamotion.com

BORGWARNER
7929 SW Burns Way
Wilsonville, OR 97070
www.CascadiaMotion.com
sales@cascadiamotion.com

A.11 Shudder Compensation Parameters

Using an electric motor in a vehicle can expose driveline resonances (shudder) that might not normally be noticed in an ICE vehicle. Typically these resonances occur at very low speeds and moderate torque levels.

The shudder compensation system implemented on the PMxxx family converters provides a mechanism for the user to try and counteract the resonance.

The basic idea is to provide a compensating torque that tries to drive any AC components of the speed to zero. That is if the speed is found to be varying (oscillating) and additional torque is added to the command that attempts to remove the oscillation.

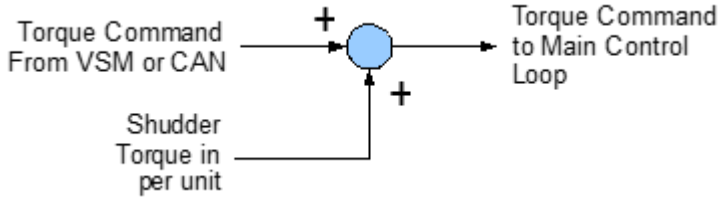


Figure 26: Shudder Torque Implementation

Figure 26 shows the mechanism for including the shudder compensation torque into the torque command. If shudder compensation is enabled the shudder torque value will be added to the normal torque command that comes from the VSM (vehicle state machine) or from a CAN command.

The mechanism for calculating the correct value of shudder torque compensation is shown in Figure 27. The compensation algorithm compares the electrical speed of the motor to a filtered version of the speed. The output of the comparison is then clamped to a value between +TCLAMP and -TCLAMP. This value is then phased out based on two speed parameters, Shudder Speed Lo and Shudder Speed Hi.

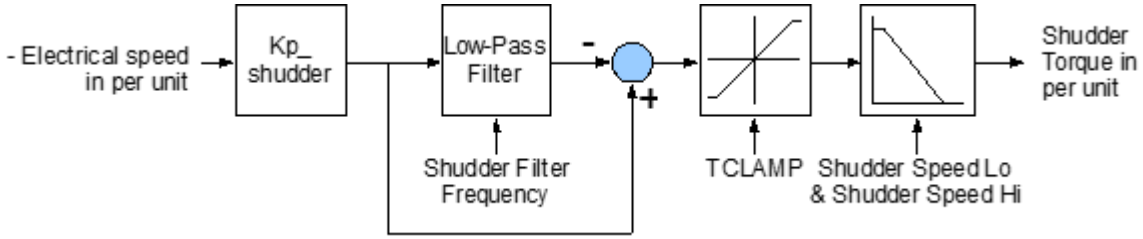


Figure 27: Shudder Compensation Algorithm

EEPROM Parameter	GUI Addr.	CAN Addr.	Value Range	Description
Shudder_Compensation_Enable_EEPROM 1 = Shudder Compensation Mode 1 2 = Shudder Compensation Mode 2	0x0134	0x00BB	0,1, 2	0 = Shudder Compensation Disabled
Kp_Shudder_EEPROM_x_100	0x0135	0x00BC	0.1 – 50	This parameter defines the gain of the shudder compensation controller. This parameter has a scaling factor of 100. Thus a setting of 100 gives a gain of 1.00. The default value of the gain is 20 (or a parameter setting of 2000). Testing of the vehicle system will be necessary to determine the best gain setting.
TCLAMP_Shudder_EEPROM_(Nm)_x_10	0x0136	0x00BD	0 – 100Nm	This parameter defines the maximum amount of compensation torque that will be added to the commanded torque. The parameter has a scaling factor of 10. Thus a setting of 10 gives a torque of 1.0 Nm. The default value is 19.9 Nm.
Shudder_Filter_Freq_EEPROM_(Hz)_x_10	0x0137	0x00BE	0.1 – 20 Hz	This parameter determines the frequency of the low-pass filter used in the shudder compensation algorithm (See Figure 27). The parameter has a scaling factor of 10. Thus a setting of 10 gives a frequency of 1.0 Hz. The default value of the parameter is 3.0 Hz (setting of 30). The filter frequency should be lower than the frequency of resonance of the drive-line. Again it may be necessary to perform testing on the vehicle to determine the correct value.
Shudder_Speed_Lo_EEPROM_(RPM)	0x0138	0x00C0	0 – 32000 RPM	Lower bound for linear region of shudder compensation. For speeds below this value, the full value of shudder torque is used. Defaults to 300.

BORGWARNER
7929 SW Burns Way
Wilsonville, OR 97070
www.CascadiaMotion.com
sales@cascadiamotion.com

EEPROM Parameter	GUI Addr.	CAN Addr.	Value Range	Description
Shudder_Speed_Hi_EEPROM_(RPM)	0x0139	0x00C1	0 – 32000 RPM	Higher bound for linear region of shudder compensation. For speeds above this value, the shudder torque will be 0. Defaults to 400.
Shudder_Speed_Fade_EEPROM_(RPM)	0x0140	0x00BF	0 – 32000 RPM	This parameters is used to define the linear phase in of the shudder torque compensation at lower speeds starting from 0 RPM. Between this value and Shudder_Speed_Lo, full value of shudder torque is used. This value must be lower than Shudder_Speed_Lo value.

Table 48: Speed Parameters

BORGWARNER
7929 SW Burns Way
Wilsonville, OR 97070
www.CascadiaMotion.com
sales@cascadiamotion.com

BORGWARNER
7929 SW Burns Way
Wilsonville, OR 97070
www.CascadiaMotion.com
sales@cascadiamotion.com

A.12 Brake Parameters (applies to VSM Mode)

The Brake input works in two modes. These modes include Switch mode and Brake Pot mode. The switch mode allows for only a single value of braking torque (regen). The Brake Pot mode allows for a variable amount of braking torque. Normally, the Brake Pot would be connected to the brake pedal of a vehicle and would change in voltage relative to the amount of brake pedal applied.

Brake Switch Mode

In this mode, the digital input DIN3 is used. Starting with version 1984 DIN7 can also be used for the brake switch input. DIN7 allows a +12V signal for the brake input instead of a grounded signal when using DIN3. When entering braking mode, the controller ramps the torque according to the regen_ramp_rate parameter. Figure 28 explains the relationship between time and regen torque when the brake input is pressed:

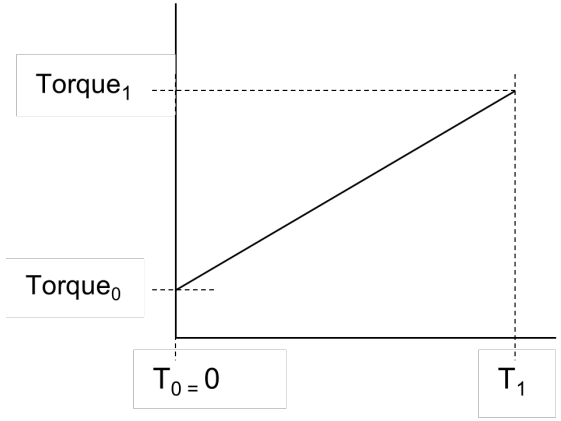


Figure 28: Brake Switch Mode

Where T_0 is the start time (in seconds) which is always 0 in this case, T_1 is the ramp period indicated by the equivalent EEPROM parameter in seconds, $Torque_0$ is value of torque that is currently produced, and $Torque_1$ is the VSM Braking Torque Limit.

Brake Pot Mode

Figure 29 details the relationship between the brake input voltage and the regen torque command:

BORGWARNER
7929 SW Burns Way
Wilsonville, OR 97070
www.CascadiaMotion.com
sales@cascadiamotion.com

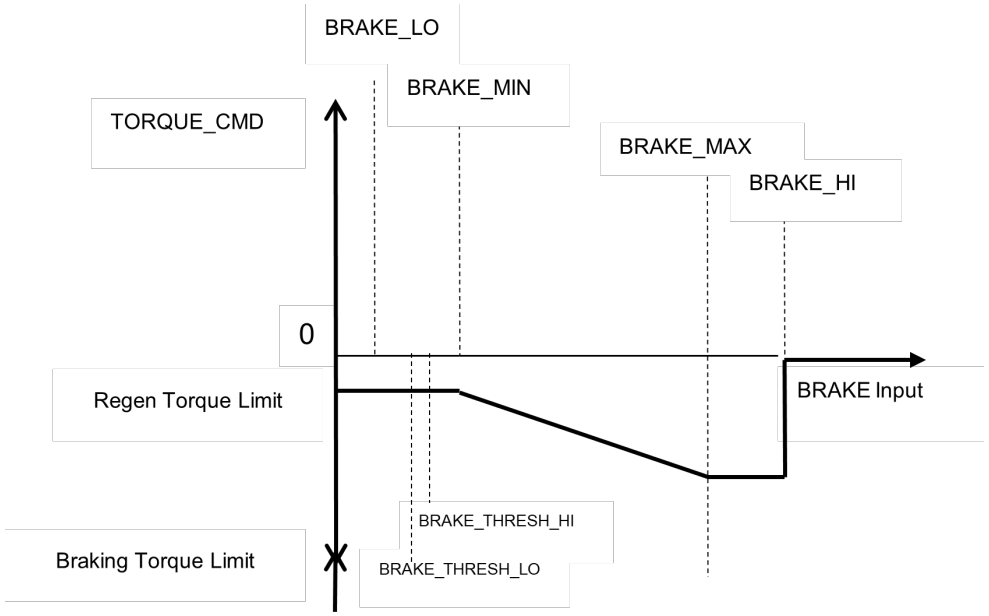


Figure 29: Brake Pot Mode

The brake input has a range of 0 to 500. This corresponds to a physical range of 0 to 5.00 volts on the input. The parameters are designed for a pedal that provides a low input voltage when the pedal is released and a higher voltage as the pedal is pressed. If the vehicle has a pedal that operates in the opposite direction use the BRAKE PEDAL FLIPPED parameter as described below.

For initial setup and calibration, the brake pedal voltage can either be monitored by a volt meter, or it can be monitored by the GUI software over the serial port.

Below is a list of the parameters that effect how the brake input works:

EEPROM Parameter	GUI Addr.	CAN Addr.	Value Range	Description
`Brake_Mode_EEPROM`	0x013A	0x00B4	0, 1 or 2	This parameter selects the mode for the brake input. 0: Brake Switch Mode 1: Brake Pot Mode

BORGWARNER
7929 SW Burns Way
Wilsonville, OR 97070
www.CascadiaMotion.com
sales@cascadiamotion.com

EEPROM Parameter	GUI Addr.	CAN Addr.	Value Range	Description
Brake_Switch_Bypassed_EEPROM	0x015F	0x00C7	0 – 2	This parameter decides if the brake input should be ignored or not in VSM mode: 0: Do not ignore brake input (process as usual) 1: Ignore brake input for starting the vehicle and for regen 2: Ignore brake input only for starting the vehicle
Brake_Pedal_Flipped_EEPROM	0x013F	0x00BA	0 or 1	If the pedal increases in voltage as it is pressed use a value of 0 (not flipped). If the pedal decreases in voltage as it is pressed use a value of 1 (flipped). When this parameter is 1, the pedal voltage will first be processed by the equation: newpedalvoltage = 5.00 − old pedal voltage. Thus will make the pedal act the same as a pedal that normally increases in voltage.
Brake_Lo_EEPROM_(V)_x_100	0x013B	0x00B5	1 – 500	For brake inputs less than this value the torque command is zero. This value should be set to a value that is lower than the lowest possible brake position, but higher than zero. If the brake input were to be shorted to ground the desired torque command is zero. Below this value, Brake Input Short Fault is set.
Brake_Min_EEPROM_(V)_x_100	0x013C	0x00B6	1 – 500	For brake inputs less than this value, the torque command is held at 0.
Brake_Max_EEPROM_(V)_x_100	0x013D	0x00B7	1 – 500	For brake inputs between BRAKE_MIN and BRAKE_MAX, the torque command is linearly decreased from 0 to Braking Torque Limit.
Brake_Hi_EEPROM_(V)_x_100	0x013E	0x00B8	1 – 500	For the range between BRAKE_MAX and BRAKE_HI the torque command is held constant at Braking Torque Limit. BRAKE_HI should be set above the normal range of pedal motion, but below 500. Above this value, Brake Input Open Fault is set.

BORGWARNER
7929 SW Burns Way
Wilsonville, OR 97070
www.CascadiaMotion.com
sales@cascadiamotion.com

EEPROM Parameter	GUI Addr.	CAN Addr.	Value Range	Description
Brake_Thresh_Lo_EEPROM_(V)_x_100	0x0161	0x0100	1 – 500	This value is supposed to be between Brake_Lo_EEPROM_(V)_x_100 and Brake_Min_EEPROM_(V)_x_100. Below this threshold, brake is considered inactive (OFF).
Brake_Thresh_Hi_EEPROM_(V)_x_100	0x0162	0x0101	1 – 500	This value is supposed to be between Brake_Lo_EEPROM_(V)_x_100 and Brake_Min_EEPROM_(V)_x_100. This value should be greater than Brake_Thresh_Lo_EEPROM_(V)_x_100 to provide some hysteresis for turning the brake switch on and off. Above this threshold, brake is considered active (ON).
Regen_Ramp_Rate_EEPROM_(Sec)_x_1000	0x0133	0x00B9	3 – 20000	This value of time is entered in milliseconds. This is the time in which regen torque value ramps down to the braking torque limit. This time can also be represented as $\|T_1 - T_0\|$.

Table 49: Speed Parameters

A.13 Active Short Circuit Parameters

RMS GUI Parameter	GUI Addr.	CAN Addr.	Value Range	Description
Active_Short_Circuit_Enabled_EEPROM	0x01C5	0x011B	0 – 3	The Active Short Circuit mode. 0: Off 1: Enabled with fault or disable 2: Enabled with fault only 3: Enabled with disable only
Active_Short_Circuit_Delay_EEPROM_(us)	0x01C6	0x011C	0 – 65535	The user-defined delay between triggering ASC and activating it.
Active_Short_Circuit_User_Off_EEPROM	0x01C7	0x011D	0 – 65535	The user-defined ASC off speed.

BORGWARNER
7929 SW Burns Way
Wilsonville, OR 97070
www.CascadiaMotion.com
sales@cascadiamotion.com

B Appendix: GUI Parameters

B.1 GUI Watch Parameters

The GUI provides the ability to monitor several operation parameters of the controller. It is also helpful for checking connections to the controller. Items can be added from the Item list to the Watch window to view the parameter.

RMS GUI Parameter	Description
RUN_Command(Trq=0_Spd=1)	Displays the current command mode (Torque control or Speed control).
Commanded_Speed_(RPM)	Shows the Commanded speed if the controller is in Speed mode.
Feedback_Speed_(RPM)	Shows the motor speed as calculated from particular motor position feedback sensor used for the motor type (e.g. encoder/resolver).
Commanded_Torque_(Nm)_x_10	The commanded torque is displayed if the controller is in torque control mode.
Feedback_Torque_(Nm)_x_10	This is the motor torque as calculated by the controller. The torque is calculated based on motor currents and the parameters of the motor. If the motor is running in reverse the Feedback Torque will have the opposite sign to the Commanded Torque.
Voltage_Feedback_Speed_(RPM)	This parameter shows the motor speed as calculated from measuring the back EMF of a PM motor. This parameter will only be valid if there is sufficient back EMF to generate a measurable voltage and the motor is not enabled. It is useful to ensure that motor phasing matches the resolver feedback (same direction/speed).
Torque_Shudder_(Nm)_x_10	Amount of torque compensation that is being applied when using the Shudder compensation feature.
Inverter_Discharge_State	The current inverter discharge state. See table 18 for definitions.
V_DC_Filtered_(V)_x_10	DC Bus Voltage measurement.
V_MAG_Filtered_(V)_x_10	The magnitude of the output voltage being applied to the motor. This is represented in line to neutral peak volts.
SW_Over_Voltage_(V)_x_10	A hard-coded value for over-voltage threshold this is used during pre-charge process and during normal operation for over-voltage detection.
I_DC_Filtered_(A)_x_10	The DC Bus current. The controller can only calculate this value as it does not actually measure the DC bus current. The calculation is based on an estimate of the motor power and the DC Bus voltage.
I_MAG_Filtered_(A)_x_10	The motor phase current magnitude. This is the peak value of the current (not RMS).
SW_Over_Current_(A)_x_10	A hard-coded value for over-current threshold this is used during normal operation for over-current detection.

BORGWARNER
7929 SW Burns Way
Wilsonville, OR 97070
www.CascadiaMotion.com
sales@cascadiamotion.com

RMS GUI Parameter	Description
Motor_Temp_(C)_x_10	Shows the motor temperature if available. The sensor used is selected automatically via the motor type. Some motors do not have a sensor selected and this will display 0 then.
Mod_A_Temp_(C)_x_10	The temperature of the sensor embedded in Phase A of the power module.
Mod_B_Temp_(C)_x_10	The temperature of the sensor embedded in Phase B of the power module.
Mod_C_Temp_(C)_x_10	The temperature of the sensor embedded in Phase C of the power module.
PCB_Temp_(C)_x_10	Temperature of the control board PCB.
GDB_1_Temp_(C)_x_10	Temperature of the gate driver board PCB 1.
GDB_2_Temp_(C)_x_10	Temperature of the gate driver board PCB 2.
GDB_3_Temp_(C)_x_10	Temperature of the gate driver board PCB 3.
RTD1_Temp_(C)_x_10	Temperature of the sensor hooked to the RTD1 input.
RTD2_Temp_(C)_x_10	Temperature of the sensor hooked to the RTD2 input.
ID_Bits	3 = CM200
Inverter_Mode	The Inverter State, see description in section 11.4
VSM_State	The VSM State, see description in section 11
Inverter_Enable	Displays a 1 when the inverter is enable, 0 if disabled.
BMS_Limit_Msg_Status	Indicates if BMS is limiting or not.
Vehicle_Direction	Shows the commanded vehicle direction, 1 = Forward, 0 = Not commanded, -1 = Reverse
Ignition_Input	Shows the state of DIN5, 1 = asserted, 0 = de-asserted.
Start_Input	Shows the state of DIN6, 1 = asserted, 0 = de-asserted.
Brake_Switch	Shows the state of DIN3, 1 = asserted, 0 = de-asserted.
Forward_Switch	Shows the state of DIN1, 1 = asserted, 0 = de-asserted.
Reverse_Switch	Shows the state of DIN2, 1 = asserted, 0 = de-asserted.
Regen_Disable_Switch	Shows the state of DIN4, 1 = asserted, 0 = de-asserted.
OK_Output_Status	Shows the state of RLY3, 1 = asserted, 0 = de-asserted.
Precharge_Output_Status	Shows the state of RLY1, 1 = asserted, 0 = de-asserted.
Main_Output_Status	Shows the state of RLY2, 1 = asserted, 0 = de-asserted.
Fault_Output_Status	Shows the state of RLY4, 1 = asserted, 0 = de-asserted.
Hall_Input_1_Status	Shows the status of Hall Input 1
Hall_Input_2_Status	Shows the status of Hall Input 2
Hall_Input_3_Status	Shows the status of Hall Input 3
Encoder_Input_A_Status	Shows the status of Encoder Input A

BORGWARNER
7929 SW Burns Way
Wilsonville, OR 97070
www.CascadiaMotion.com
sales@cascadiamotion.com

RMS GUI Parameter	Description
Encoder_Input_B_Status	Shows the status of Encoder Input B
Encoder_Input_Z_Status	Shows the status of Encoder Input Z
SAT_Fault_Output_Status	Shows the status of HW Desaturation fault, 0 = asserted, 1 = de-asserted
OC_Fault_Output_Status	Shows the status of HW Over-current fault, 0 = asserted, 1 = de-asserted
VSM_Accel_Filtered	Shows the voltage applied to AIN1, 0 = 0 volts, 500 = 5.0 volts
VSM_Brake_Filtered	Shows the voltage applied to AIN3, 0 = 0 volts, 500 = 5.0 volts
Power_on_Timer_3ms_(Hi_byte)	The controller keeps a count of how many 3ms intervals have occurred since power was applied. It is represented as a 32 bit number.
Power_on_Timer_3ms_(Lo_byte)	See above.
Motor_Angle_(DEG)_x_10	Shows the rotational position of the motor shaft. Can be used to verify encoder or resolver operation.
Delta_Resolver_In_Fil_(Deg)_x_10	This parameter is used for calibration of the resolver offset. It shows the offset between the back EMF angle and the resolver angle. Only valid if the motor is not enabled.
Delta_Resolver_In_Fil_On_Coast	This parameter displays Delta_Resolver_In_Fil_(Deg)_x_10 during motor coast down.
Gamma_Resolver_(Deg)_x_10	This parameter is the actual electrical angle of the motor in degrees times 10. That is 3600 is 360 degrees and 0 is 0 degrees. The electrical angle of the motor should not be confused with the mechanical angle. The electrical angle reflects the number of poles the motor has. For example, if a motor has 12 poles (6 pole pairs), the resolver will go through 6 rotations electrically for every shaft rotation.
Sin_Minimum_Value_(V)(x100)	The minimum Sin voltage value seen during encoder calibration.
Sin_Maximum_Value_(V)(x100)	The maximum Sin voltage value seen during encoder calibration.
Sin_Offset_(V)(x100)	The current Sin offset programmed.
Cos_Minimum_Value_(V)(x100)	The minimum Cos voltage value seen during encoder calibration.
Cos_Maximum_Value_(V)(x100)	The maximum Sin voltage value seen during encoder calibration.
Cos_Offset_(V)(x100)	The current Cos offset programmed.

Table 51: GUI Watch Parameters

BORGWARNER
7929 SW Burns Way
Wilsonville, OR 97070
www.CascadiaMotion.com
sales@cascadiamotion.com

B.2 GUI Command Parameters

Some features require sending data to the inverter which can be accomplished via various GUI Commands. Items can be added from the Item list to the Command window to modify the parameter.

EEPROM Parameter	GUI Addr.	CAN Addr.	Description
`Outputs_Command`	0x0012	0x0001	0xAA00: Normal Run Mode 0x55*nn*: External Relay Control Mode See `Relay_Output_State_EEPROM`.
`Veh_Flux_Command_(Wb)(x1000)`	0x0004	0x000A	Modifies the flux command.
`Gamma_Adjust_(Deg)_x_10`	0x0010	0x000C	This is a command parameter. The value can be adjusted by typing the new data in the GUI. This parameter is used with the resolver calibration procedure. This parameter is an offset angle added to the resolver feedback angle. The parameter will reset to the EEPROM whenever the power is cycled to the controller.
`Fault_Clear`	0x0002	0x0014	Writing a 0 to this parameter clears any active faults.
`PWM_Frequency`	NA	0x0015	Setting the PWM frequency via this parameter command will over-ride the EEPROM settings. The PWM frequency will revert back to the EEPROM settings when power is cycled.
`Analog_Input_Pull_Up_Ctrl_`	0x0022	0x0016	This parameter allows enabling of the AIN1-AIN3 pull-ups. Bit 0 controls AIN1, bit 1 controls AIN2 and bit 2 controls AIN3.
`Shudder_Comp_Gain`	NA	0x0017	If shudder compensation is enabled, this parameter will allow `Kp_Shudder_EEPROM_x_100` to be modified and for the Shudder Compensation to be disabled and re-enabled. If the Parameter data is set to 0, shudder compensation will be disabled.
`OBD2_Enable_Command`	NA	0x1E	Enables OBD2 CAN mailbox with the ID offset specified in the parameter data bytes.
`Encoder_Calibration_Command`	0x0015	0x0013	TBD

BORGWARNER
7929 SW Burns Way
Wilsonville, OR 97070
www.CascadiaMotion.com
sales@cascadiamotion.com

Diag Data Trigger	NA	0x1F	Trigger a diag data download over CAN.

Table 52: GUI Command Parameters

C Appendix: CAN Broadcast Messages

Table 53: 0x0A0: Temperatures #1

Byte(s)	Name	Format	Description
0,1	INV_Module_A_Temp	Temperature	Temperature of Power Module, Phase A
2,3	INV_Module_B_Temp	Temperature	Temperature of Power Module, Phase B
4,5	INV_Module_C_Temp	Temperature	Temperature of Power Module, Phase C
6,7	INV_GDB_Temp	Temperature	Temperature of Gate Driver Board

Table 54: 0x0A1: Temperatures #2

Byte(s)	Name	Format	Description
0,1	INV_Control_Board_Temp	Temperature	Temperature of Control Board
2,3	INV_RTD1_Temperature	Temperature	Temperature read from RTD input #1
4,5	INV_RTD2_Temperature	Temperature	Temperature read from RTD input #2
6,7	INV_Hot_Spot_Temp_Motor	Temperature	The motor hot spot temperature estimate. See section **??**, '**??**' for more details.

Table 55: 0x0A2 Temperatures #3 & Torque Shudder

Byte(s)	Name	Format	Description
0,1	INV_Coolant_Temp	Temperature	Estimated coolant temperature
2,3	INV_Hot_Spot_Temp_Inverter	Temperature	Estimated hot spot temperature internal to inverter
4,5	INV_Motor_Temp	Temperature	Filtered temperature value from the motor temperature sensor
6,7	INV_Torque_Shudder	Temperature	A value of torque used in shudder compensation

BORGWARNER
7929 SW Burns Way
Wilsonville, OR 97070
www.CascadiaMotion.com
sales@cascadiamotion.com

Table 56: 0x0A3: Analog Input Voltage

Bit(s)	Name	Format	Description
0 – 9	INV_Analog_Input_1	Low Voltage	Voltage on Analog Input #1
10 – 19	INV_Analog_Input_2	Low Voltage	Voltage on Analog Input #2
20 – 29	INV_Analog_Input_3	Low Voltage	Voltage on Analog Input #3
32 – 41	INV_Analog_Input_4	Low Voltage	Voltage on Analog Input #4
42 – 51	INV_Analog_Input_5	Low Voltage	Voltage on Analog Input #5
52 – 61	INV_Analog_Input_6	Low Voltage	Voltage on Analog Input #6

Table 57: 0x0A3: Analog Input Voltages (for CM firmware where iM-225 motor type is used)

Bit(s)	Name	Format	Description
0 – 15	INV_Oil_Temperature	Temperature	Oil Temperature of iM-225 motor
16 – 31	INV_Oil_Pressure	Pressure	Oil Pressure of iM-225 motor
32 – 41	INV_Analog_Input_4	Low Voltage	Voltage on Analog Input #4
42 – 51	INV_Analog_Input_5	Low Voltage	Voltage on Analog Input #5
52 – 61	INV_Analog_Input_6	Low Voltage	Voltage on Analog Input #6

Table 58: 0x0A4: Digital Input Status

Bit(s)	Name	Format	Description
0	INV_Digital_Input_1	Boolean	Status of Digital Input #1, Forward switch
1	INV_Digital_Input_2	Boolean	Status of Digital Input #2, Reverse switch
2	INV_Digital_Input_3	Boolean	Status of Digital Input #3, Brake switch
3	INV_Digital_Input_4	Boolean	Status of Digital Input #4, Regen Disable switch
4	INV_Digital_Input_5	Boolean	Status of Digital Input #5, Ignition switch
5	INV_Digital_Input_6	Boolean	Status of Digital Input #6, Start switch
6	INV_Digital_Input_7	Boolean	Status of Digital Input #7, Valet Mode
7	INV_Digital_Input_8	Boolean	Status of Digital Input #8

BORGWARNER
7929 SW Burns Way
Wilsonville, OR 97070
www.CascadiaMotion.com
sales@cascadiamotion.com

BORGWARNER
7929 SW Burns Way
Wilsonville, OR 97070
www.CascadiaMotion.com
sales@cascadiamotion.com

Table 59: 0x0A5: Motor Position Information

Byte(s)	Name	Format	Description
0,1	INV_Motor_Angle_Electrical	Angle	The electrical angle of the motor as read by the encoder or resolver.
2,3	INV_Motor_Speed	Angular Velocity	The measured speed of the motor.
4,5	INV_Electrical_Output_Frequency	Frequency	The actual electrical frequency of the inverter.
6,7	INV_Delta_Resolver_Filtered	Angle	This is used in calibration of resolver angle adjustment. The range of this parameter is ±180°. Values between 180° and 360° are shown as negative angle. For example, 270° is equal to -90°, and 190° is equal to -170°.

Table 60: 0x0A6: Current Information

Byte(s)	Name	Format	Description
0,1	INV_Phase_A_Current	Current	The measured value of Phase A Current.
2,3	INV_Phase_B_Current	Current	The measured value of Phase B Current.
4,5	INV_Phase_C_Current	Current	The measured value of Phase C Current.
6,7	INV_DC_Bus_Current	Current	The calculated DC Bus current.

Table 61: 0x0A7: Voltage Information

Byte(s)	Name	Format	Description
0,1	INV_DC_Bus_Voltage	High Voltage	The actual measured value of the DC bus voltage.
2,3	INV_Output_Voltage	High Voltage	The calculated value of the output voltage, in peak line-neutral volts.
4,5	INV_VAB_Vd_Voltage	High Voltage	Measured value of the voltage between Phase A and Phase B (VAB) when the inverter is disabled. Vd voltage when the inverter is enabled.
6,7	INV_VBC_Vq_Voltage	High Voltage	Measured value of the voltage between Phase B and Phase C (VBC) when the inverter is disabled. Vq voltage when the inverter is enabled.

Table 62: 0x0A8: Flux Information

Byte(s)	Name	Format	Description
0,1	INV_Vd_ff	Flux	D-axis voltage feed-forward.
2,3	INV_Vq_ff	Flux	Q-axis voltage feed-forward
4,5	INV_Id	Current	D-axis current feedback
6,7	INV_Iq	Current	Q-axis current feedback

Table 63: 0x0A9: Internal Voltages

Byte(s)	Name	Format	Description
0,1	INV_Ref_Voltage_1_5	Low Voltage	1.5V Reference voltage.
2,3	INV_Ref_Voltage_2_5	Low Voltage	2.5V Reference voltage.
4,5	INV_Ref_Voltage_5_0	Low Voltage	5.0V Reference voltage.
6,7	INV_Ref_Voltage_12_0	Low Voltage	12V Reference voltage.

Byte(s)	Name	Format	Description
0	INV_VSM_State	Internal	0: VSM Start State 1: Pre-charge Init State 2: Pre-charge Active State 3: Pre-charge Complete State 4: VSM Wait State 5: VSM Ready State 6: Motor Running State 7: Blink Fault Code State 14: Shutdown in Process. In key switch mode 1, user has turned the key switch to off position. 15: Recycle Power State. User must recycle power when the unit is in this state.
1	INV_PWM_Frequency	kHz	The PWM frequency may change depending on operating conditions.

BORGWARNER
7929 SW Burns Way
Wilsonville, OR 97070
www.CascadiaMotion.com
sales@cascadiamotion.com

Byte(s)	Name	Format	Description
2	INV_Inverter_State	Internal	0: Power on State 1: Stop State 2: Open Loop State 3: Closed Loop State 4: Wait State 5– 7: *Internal States* 8: Idle Run State 9: Idle Stop State 10– 12: *Internal States*
3	Relay States	Internal	Bit 0: INV_Relay_1_Status (1=active) Bit 1: Relay 2 Status Bit 2: Relay 3 Status Bit 3: Relay 4 Status Bit 4: Relay 5 Status Bit 5: Relay 6 Status
4-Bit 0:	INV_Inverter_Run_Mode	Internal	0 = Torque Mode 1 = Speed Mode
4-Bit 1:	INV_Self_Sensing_Assist_Enable	Internal	0 = Disabled 1 = Enabled Indicates if Self-Sensing Assist is currently active. Only for select motors.
4-Bits 2 – 4:	INV_ASC_State	Internal	Current Active Short Circuit State: 0: ASC_Disabled 1: ASC_Enabled 2: ASC_Pending 3: ASC_Delay 4: ASC_Active 5: ASC_Complete 6: ASC_Blocked 7: ASC_Suppressed

BORGWARNER
7929 SW Burns Way
Wilsonville, OR 97070
www.CascadiaMotion.com
sales@cascadiamotion.com

Byte(s)	Name	Format	Description
4-Bits 5 – 7:	INV_Inverter_Discharge_State	Internal	Current Inverter Active Discharge State: 0: Disabled 1: Enabled, waiting 2: Speed Check 3: Active 4: Completed All other states are reserved for future use.
5-Bit 0:	INV_Inverter_Command_Mode	Internal	0 = CAN Mode 1 = VSM Mode When in CAN Mode the inverter takes commands from the CAN messages.
5-Bits 4 – 7:	INV_Rolling_Counter	Internal	The value of the currently expected Rolling Counter value.
6-Bit 0:	INV_Inverter_Enable_State	Internal	0 = Inverter is disabled 1 = Inverter is enabled
6-Bit 1:	INV_Burst_Model_Mode	Internal	0 = Stall 1 = High Speed At low speeds Stall Burst Model will apply when mode is 0 (Stall) for select inverters only. See section 6.3.
6-Bit 2:	INV_BMS_Limiting_Regen_Torque	Internal	0 = Not Limited 1 = Limited
6-Bit 4:	INV_Limit_Motor_Temp_Derate	Internal	Motor temp factor derate from EEPROM settings is limiting output current. 0 = Not Limited 1 = Limited
6-Bit 5:	INV_Limit_Hot_Spot_Motor	Internal	Motor hot spot is limiting output current. 0 = Not Limited 1 = Limited
6-Bit 6:	INV_Key_Switch_Start_Status	Internal	0 = Not Active 1 = Active Provides latched indication of when the start signal has been received. Typically used when using VSM mode.

BORGWARNER
7929 SW Burns Way
Wilsonville, OR 97070
www.CascadiaMotion.com
sales@cascadiamotion.com

Byte(s)	Name	Format	Description
6-Bit 7:	INV_Inverter_Enable_Lockout	Internal	0 = Inverter can be enabled 1 = Inverter cannot be enabled This feature is added so that the inverter cannot be accidentally enabled. This feature requires that before sending out an Inverter Enable command, the user must send out an Inverter Disable command. Once the inverter sees a Disable Command, the lockout is removed and the controller can receive the Enable Command.
7-Bit 0:	INV_Direction_Command	Internal	1 = Forward 0 = Reverse, if inverter is enabled. Stopped if inverter is disabled.
7-Bit 1:	INV_BMS_Active	Internal	0 = BMS Message is not being received 1 = BMS Message is being received
7-Bit 2:	INV_BMS_Limiting_Motor_Torque	Internal	Indicates if motoring torque is being limited
7-Bit 3:	INV_Limit_Max_Speed	Internal	Indicates if torque is being limited due to the motor speed exceeding the maximum motor speed.
7-Bit 4:	INV_Limit_Hot_Spot_Inverter	Internal	Indicates if the current is limited due to the inverter hot spot temperature regulator.
7-Bit 5:	INV_Low_Speed_Limiting	Internal	Indicates if the current is limited due to low speed current limiting
7-Bit 6:	INV_Limit_Coolant_Derating	Internal	Indicates if the current is limited due to excessive coolant temperature
7-Bit 7:	INV_Limit_Stall_Burst_Model	Internal	Indicates if the current is being limited due to the stall burst model.

Table 64: 0x0AA: Internal States

BORGWARNER
7929 SW Burns Way
Wilsonville, OR 97070
www.CascadiaMotion.com
sales@cascadiamotion.com

BORGWARNER
7929 SW Burns Way
Wilsonville, OR 97070
www.CascadiaMotion.com
sales@cascadiamotion.com

Table 65: 0x0AB: Fault Codes

Byte(s)	Name	Format	Description
0,1	INV_Post_Fault_Lo	Internal	Each bit represents a fault. See section D.1, 'POST Faults' for more details.
2,3	INV_Post_Fault_Hi	Internal	Each bit represents a fault. See section D.1, 'POST Faults' for more details.
4,5	INV_Run_Fault_Lo	Internal	Each bit represents a fault. See section D.2, 'Run Faults' for more details.
6,7	INV_Run_Fault_Hi	Internal	Each bit represents a fault. See section D.2, 'Run Faults' for more details.

Table 66: 0x0AC: Torque & Timer Information

Byte(s)	Name	Format	Description
0,1	INV_Commanded_Torque	Torque	The commanded torque.
2,3	INV_Torque_Feedback	Torque	The estimated motor torque based on motor parameters and feedbacks.
4,5,6,7	INV_Power_On_Timer	(Counts x 0.003) sec	This timer is updated every 3msec. This timer will roll-over in approximately 5 months. The timer will reset to 0 at power on. Monitoring this can be useful to show when a reset of the processor has occurred.

Table 67: 0x0AD: Modulation Index & Flux Weakening Output Information

Byte(s)	Name	Format	Description
0,1	Modulation Index	Per-unit value	This is the modulation index. The scale factor is x100. To get the actual modulation index divide the value by 100.
2,3	Flux Weakening Output	Current	This is normally the current output of the flux regulator. When the DCLink_FW_Reg_Use_EEPROM is active (1) then the output will be volts even though the label is still in amps.
4,5	Id Command	Current	The commanded D-axis current
6,7	Id Command	Current	The commanded D-axis current

BORGWARNER
7929 SW Burns Way
Wilsonville, OR 97070
www.CascadiaMotion.com
sales@cascadiamotion.com

Table 68: 0x0AE: Firmware Information

Byte(s)	Name	Format	Description
0,1	EEPROM Version/Project Code	NA	This is an EEPROM version that is assigned to each project. For factory use only!
2,3	Software Version	NA	This is the software version with major and minor release values.
4,5	Date Code (mmdd)	NA	This is the portion of date code that displays month and date information in *mmdd* format.
6,7	Date Code (yyyy)	NA	This is the portion of the date code that displays year information in *yyyy* format.

Table 69: 0x0AF: Diagnostic Data

Byte(s)	Name	Format	Description
Please refer to section 6.8 'Download Diagnostic Data' for more details.			

Table 70: 0x0B0: High Speed Message (transmitted at 3ms)

Byte(s)	Name	Format	Description
0,1	Torque Command	Torque	The commanded torque.
2,3	Torque Feedback	NA	The estimated motor torque.
4,5	Motor Speed	Angular Velocity	The measured motor speed.
6,7	DC Bus Voltage	High Voltage	The actual measured DC bus voltage.

Table 71: 0x0B1: Torque Capability

Byte(s)	Name	Format	Description
0,1	Motor Torque Available	Torque	The available torque for motoring.
2,3	Regen Torque Available	Torque	The available torque for regen.
4,5	NA	NA	
6,7	NA	NA	

D Appendix: Faults

D.1 POST Faults

POST Fault	Fault Word	Description
Hardware Gate/Desaturation Fault	0x00000001	A hardware de-saturation fault occurs for any of the following conditions: • Excess current causes a short-circuit in an IGBT. • An IGBT circuit is bad. • An over-voltage condition occurs on DC bus. **Currently, this fault cannot be cleared using the 'Clear Fault Command'. In order to clear this fault, inverter power must be recycled.**
HW Over-current Fault	0x00000002	This fault occurs when any of the current sensors detect an over-current condition which could be positive or negative. All six over-current faults are ORed together to cause the HW over-current fault.
Accelerator Shorted	0x00000004	Accelerator input voltage is less than `Pedal_Lo_EEPROM_(V)_x_100`.
Accelerator Open	0x00000008	Accelerator input voltage is more than `Pedal_Hi_EEPROM_(V)_x_100`.
Current Sensor Low	0x00000010	Current sensor reading is lower than the hard-coded value (-22.5 Amps).
Current Sensor High	0x00000020B	Current sensor reading is higher than the hard-coded value (22.5 Amps).
Module Temperature Low	0x00000040	This fault is currently not active.
Module Temperature High	0x00000080	One or more of the three module temperatures are above 125 ° C.
Control PCB Temperature Low	0x00000100	PCB temperature is below -24 ° C.
Control PCB Temperature High	0x00000200	PCB temperature has exceeded 125 ° C.
Gate Drive PCB Temperature Low	0x00000400	GDB temperature is below -24 ° C.
Gate Drive PCB Temperature High	0x00000800	GDB temperature has exceeded 125 ° C.
5V Sense Voltage Low	0x00001000	V Sense reading is too low
5V Sense Voltage High	0x00002000	5V Sense reading is too high
12V Sense Voltage Low	0x00004000	12V Sense reading is too low
12V Sense Voltage High	0x00008000	12V Sense reading is too high
2.5V Sense Voltage Low	0x00010000	2.5V Sense reading is too low
2.5V Sense Voltage High	0x00020000	2.5V Sense reading is too high
1.5V Sense Voltage Low	0x00040000	1.5V Sense reading is too low
1.5V Sense Voltage High	0x00080000	1.5V Sense reading is too high

BORGWARNER
7929 SW Burns Way
Wilsonville, OR 97070
www.CascadiaMotion.com
sales@cascadiamotion.com

POST Fault	Fault Word	Description
DC Bus Voltage High	0x00100000	During pre-charge, DC voltage is above the hard-coded SW over-voltage limit. SW over-voltage limit can be checked from the monitored parameter list by adding `SW_Over_Voltage_(V)_x_10` to the watch list.
DC Bus Voltage Low	0x00200000	DC bus voltage is below 100-V.
Pre-charge Timeout	0x00400000	DC bus voltage is not charging at the rate of 2.7 V/50 msec and 3 seconds have elapsed.
Pre-charge Voltage Failure	0x00800000	After pre-charge is complete, DC voltage has changed by more than 10-V within 15 msec.
EEPROM Checksum Invalid	0x01000000	EEPROM checksum is not valid.
EEPROM Data Out of Range	0x02000000	This fault is currently not active.
EEPROM Update Required	0x04000000	The number of EEPROM parameters has changed (most of the time increased), check the new parameters and set appropriate values.
Reserved	0x08000000	
Reserved	0x10000000	
Reserved	0x10000000	
Brake Shorted	0x40000000	Brake input voltage is less than the value in EEPROM parameter, `Brake_Lo_EEPROM_(V)_x_100`.
Brake Open	0x80000000	Brake input voltage is more than the value in EEPROM parameter, `Brake_Hi_EEPROM_(V)_x_100`.

Table 72: POST Faults

BORGWARNER
7929 SW Burns Way
Wilsonville, OR 97070
www.CascadiaMotion.com
sales@cascadiamotion.com

D.2 Run Faults

RUN Fault	Fault Word	Description
Motor Over-speed Fault	0x00000001	Motor speed is greater than `Motor_Overspeed_EEPROM_(RPM)`.
Over-current Fault	0x00000002	One or more of the three phase currents is above the hard-coded SW over-current limit. SW over-current limit can be checked from the monitored parameter list by reading `SW_Over_Current_(A)_x_10`.
Over-voltage Fault	0x00000004	Filtered value of DC voltage is above the hard-coded SW over-voltage limit. SW over-voltage limit can be checked from the monitored parameter list by adding `SW_Over_Voltage_(V)_x_10` to the watch list.
Inverter Over-temperature Fault	0x00000008	One or more of the three module temperatures are above the value in EEPROM parameter, `Inv_OverTemp_Limit_EEPROM_(C)_x_10`.
Accelerator Input Shorted Fault	0x00000010	Accelerator input is below `Pedal_Lo_EEPROM_(V)_x_100`.
Accelerator Input Open Fault	0x00000020	Accelerator input is above `Pedal_Hi_EEPROM_(V)_x_100`.
Direction Command Fault	0x00000040	Both directions forward and reverse are active at the same time. **This fault has been de-activated.**
Inverter Response Time-out Fault	0x00000080	Inverter has not been enabled within 2 minutes of receiving the inverter enable command either through VSM or CAN.
Hardware Gate/Desaturation Fault	0x00000100	A hardware de-saturation fault occurs for any of the following conditions: • Excess current causes a short-circuit in an IGBT. • An IGBT circuit is bad. • An over-voltage condition occurs on DC bus. **Currently, this fault cannot be cleared using the 'Clear Fault Command'. In order to clear this fault, inverter power must be recycled.**
Hardware Over-current Fault	0x00000200	This fault occurs when any of the current sensors detect an over-current condition which could be positive or negative. All six over-current faults are ORed together to cause the HW over-current fault.
Under-voltage Fault	0x00000400	DC bus voltage is below the value in EEPROM parameter, `DC_UnderVolt_Thresh_EEPROM_(V)_x_10`

BORGWARNER
7929 SW Burns Way
Wilsonville, OR 97070
www.CascadiaMotion.com
sales@cascadiamotion.com

RUN Fault	Fault Word	Description
CAN Command Message Lost Fault	0x00000800	The inverter is not able to see the heartbeat command message when in CAN mode.
Motor Over-temperature Fault	0x00001000	The motor temperature value exceeds `Mtr_OverTemp_Limit_EEPROM_(C)_x_10`
Reserved	0x00002000	
Reserved	0x00004000	
Reserved	0x00008000	
Brake Input Shorted Fault	0x00010000	Brake input is below the value `Brake_Lo_EEPROM_(V)_x_100`
Brake Input Open Fault	0x0002000	Brake input is above the value `Brake_Hi_EEPROM_(V)_x_100`
Module A Over-temperature Fault	0x00040000	Module A temperature has exceeded `Inv_OverTemp_Limit_EEPROM_(C)_x_10`
Module B Over-temperature Fault	0x00080000	Module B temperature has exceeded `Inv_OverTemp_Limit_EEPROM_(C)_x_10`
Module C Over-temperature Fault	0x00100000	Module C temperature has exceeded `Inv_OverTemp_Limit_EEPROM_(C)_x_10`
PCB Over-temperature	0x00200000	PCB temperature has exceeded `Inv_OverTemp_Limit_EEPROM_(C)_x_10`
GDB1 Over-temperature	0x00400000	GDB 1 temperature has exceeded `Inv_OverTemp_Limit_EEPROM_(C)_x_10`
GDB 2 Over-temperature	0x00800000	GDB2 temperature has exceeded `Inv_OverTemp_Limit_EEPROM_(C)_x_10`
GDB 3 Over-temperature	0x01000000	GDB3 temperature has exceeded `Inv_OverTemp_Limit_EEPROM_(C)_x_10`
Current sensor fault	0x02000000	Current sensor fault
Gate Driver Over-voltage	0x04000000	Gate driver over-voltage
Reserved	0x08000000	
Hardware Over-voltage	0x10000000	Hardware over-voltage exceeded hardware detection, this value is dependent on inverter.
Reserved	0x20000000	
Resolver Fault	0x40000000	Resolver faulted
Reserved	0x80000000	

Table 73: RUN Faults

BORGWARNER
7929 SW Burns Way
Wilsonville, OR 97070
www.CascadiaMotion.com
sales@cascadiamotion.com

E Appendix: Stall Burst Performance Data

E.1 CM200DX Stall Burst Performance

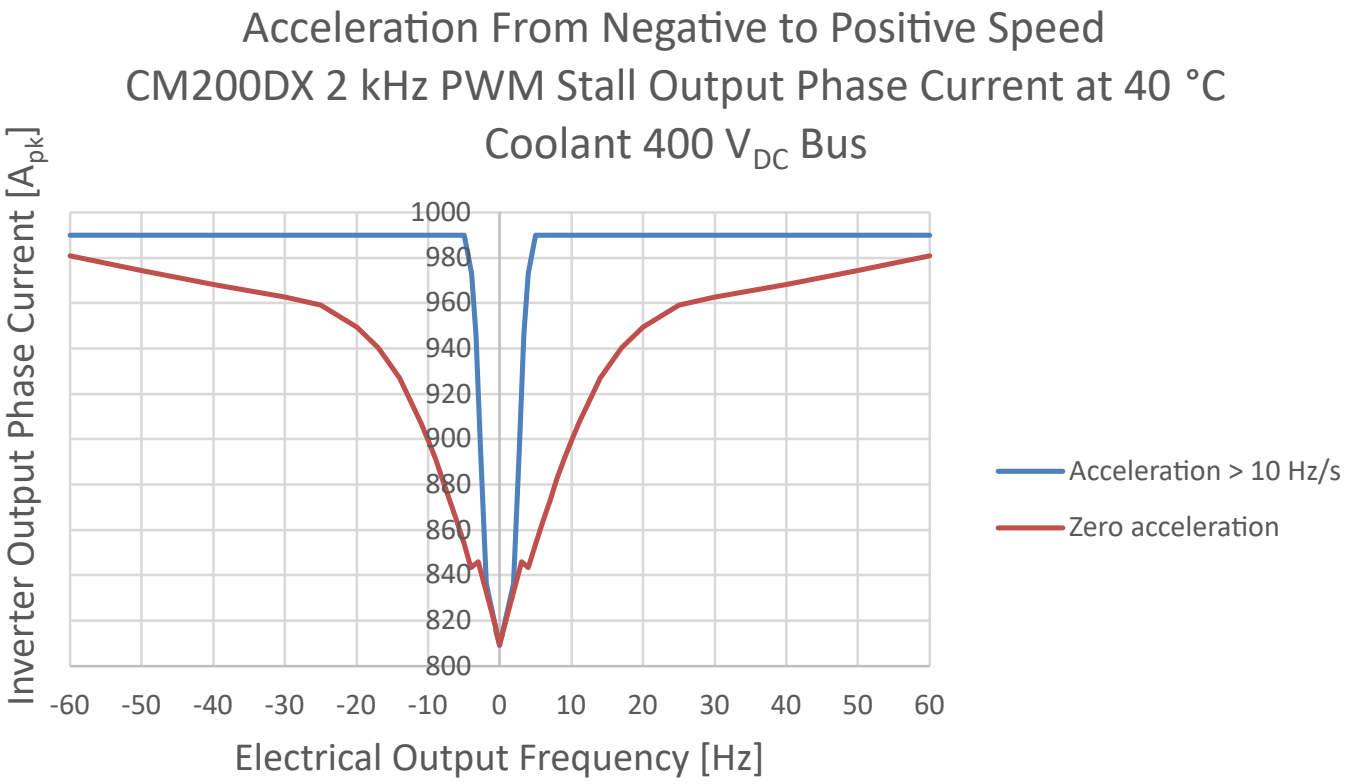


Figure 30: CM200DX Stall Output Phase Current vs Electrical Output Frequency.

Figure 30 reflects what happens when a vehicle rolls back, then accelerates forward faster than 10 Hz/s. Performance at various constant electrical output frequencies is also shown for comparison. Vehicles rarely have true zero acceleration when full torque is commanded due to driveline slop. Therefore, performance with acceleration better represents typical performance.

Results will vary depending on the initial rotor position because different rotor positions have different distributions of current in each phase. The above figure represents worst case initial rotor position. Sometimes rolling back slightly when on an incline can help low speed performance by changing the rotor initial position.

If held at zero speed, the steady state current is 809.2 A_{pk} for 120 seconds. Current will reduce further to as low as 223 A_{pk} due to hot spot limiting after 120 seconds.

Note, at zero speed phase current is DC, and thus equivalent to A_{pk}. Motor control tables may often list current in A_{RMS}, thus divided the A_{pk} by $\sqrt{2}$ to get the appropriate A_{RMS} for use in lookup tables where necessary.

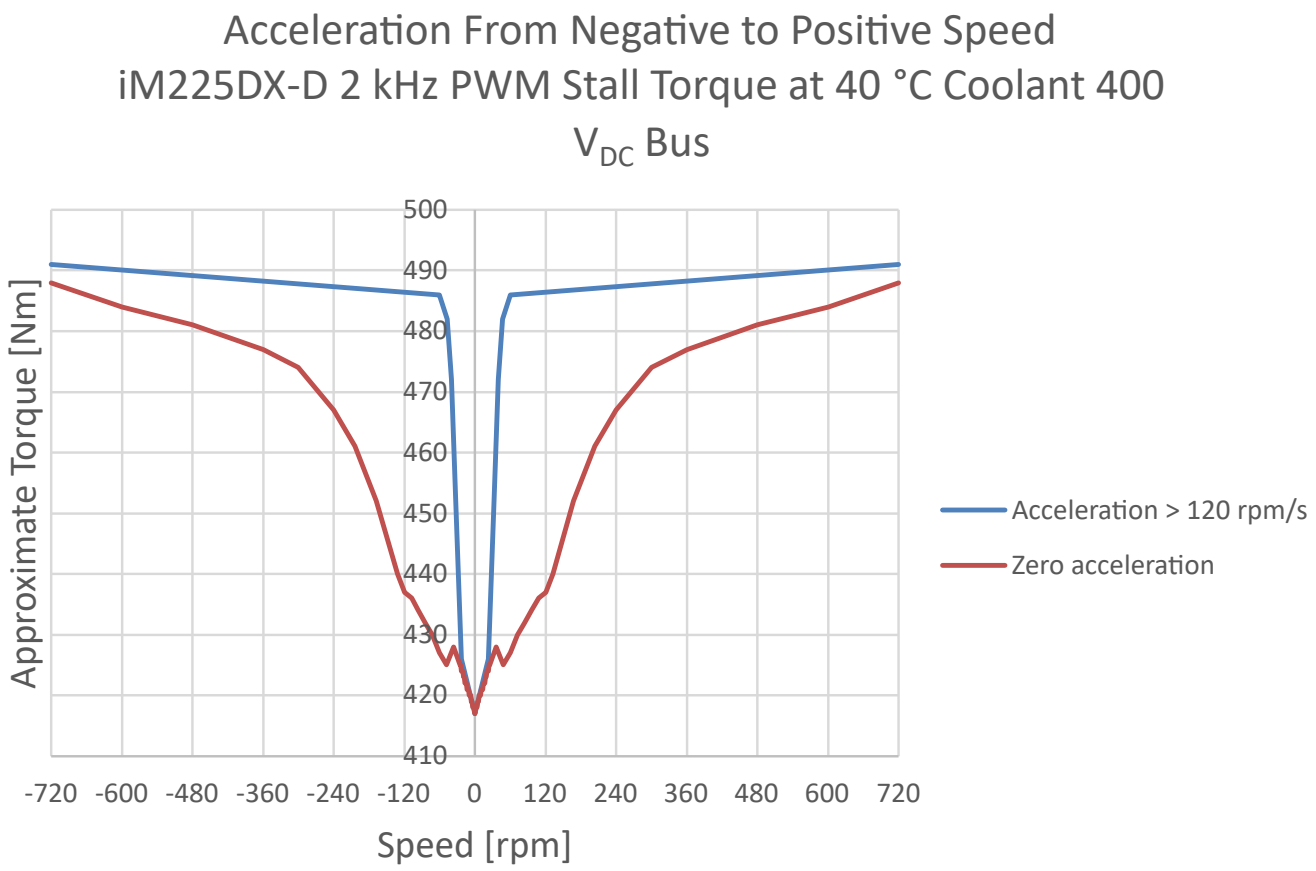


Figure 31: iM225DX-D Stall Torque vs Speed.

E.2 CM200DZ Stall Burst Performance

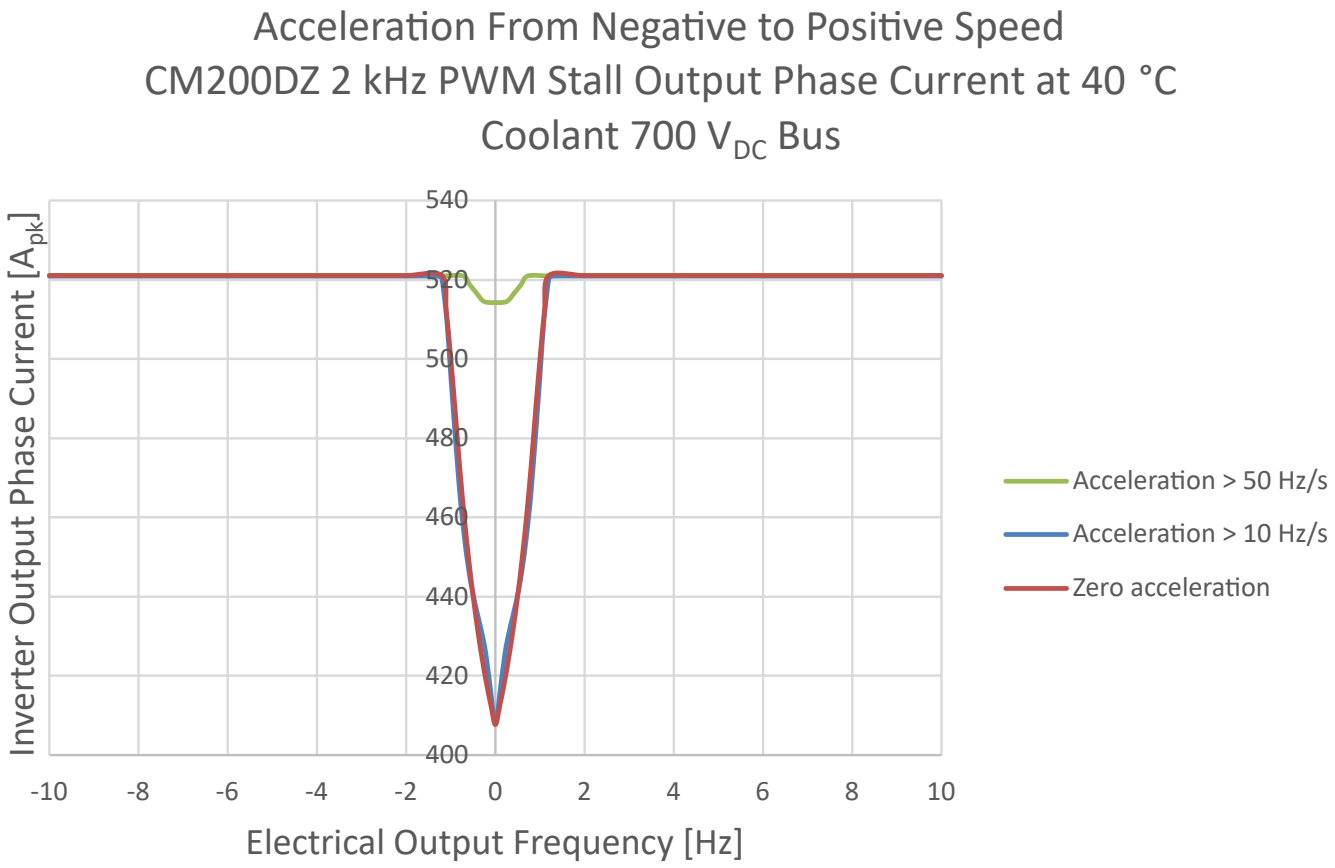


Figure 32: CM200DZ Stall Output Phase Current vs Electrical Output Frequency.

Figure 32 reflects what happens when a vehicle rolls back, then accelerates forward faster than 10 Hz/s and 50 Hz/s. Performance at various constant electrical output frequencies is also shown for comparison. CM200DZ performance when accelerating can be better than constant output frequency performance. Rolling back and accelerating forward minimizes duration at zero speed resulting in better performance while passing through zero speed. Rolling back and ramping torque during the rollback results in better performance because torque is ramping while spinning which results in a lower initial temperature when entering zero speed. Power module junction temperature rises faster and higher when torque is slowly ramped at zero speed instead of when motor is spinning. Vehicles rarely have true zero acceleration when full torque is commanded due to driveline slop. Therefore, performance with acceleration better represents typical performance.

Results will vary depending on the initial rotor position because different rotor positions have different distributions of current in each phase. The above figure represents worst case initial rotor position.

If held at zero speed, the steady state current is 407.7 A_{pk} for 47 seconds. Current will reduce further to as low as 272 A_{pk} due to hot spot limiting after 47 seconds.

Note, at zero speed phase current is DC, and thus equivalent to A_{pk}. Motor control tables may often list current in A_{RMS}, thus divided the A_{pk} by $\sqrt{2}$ to get the appropriate A_{RMS} for use in lookup tables where necessary.

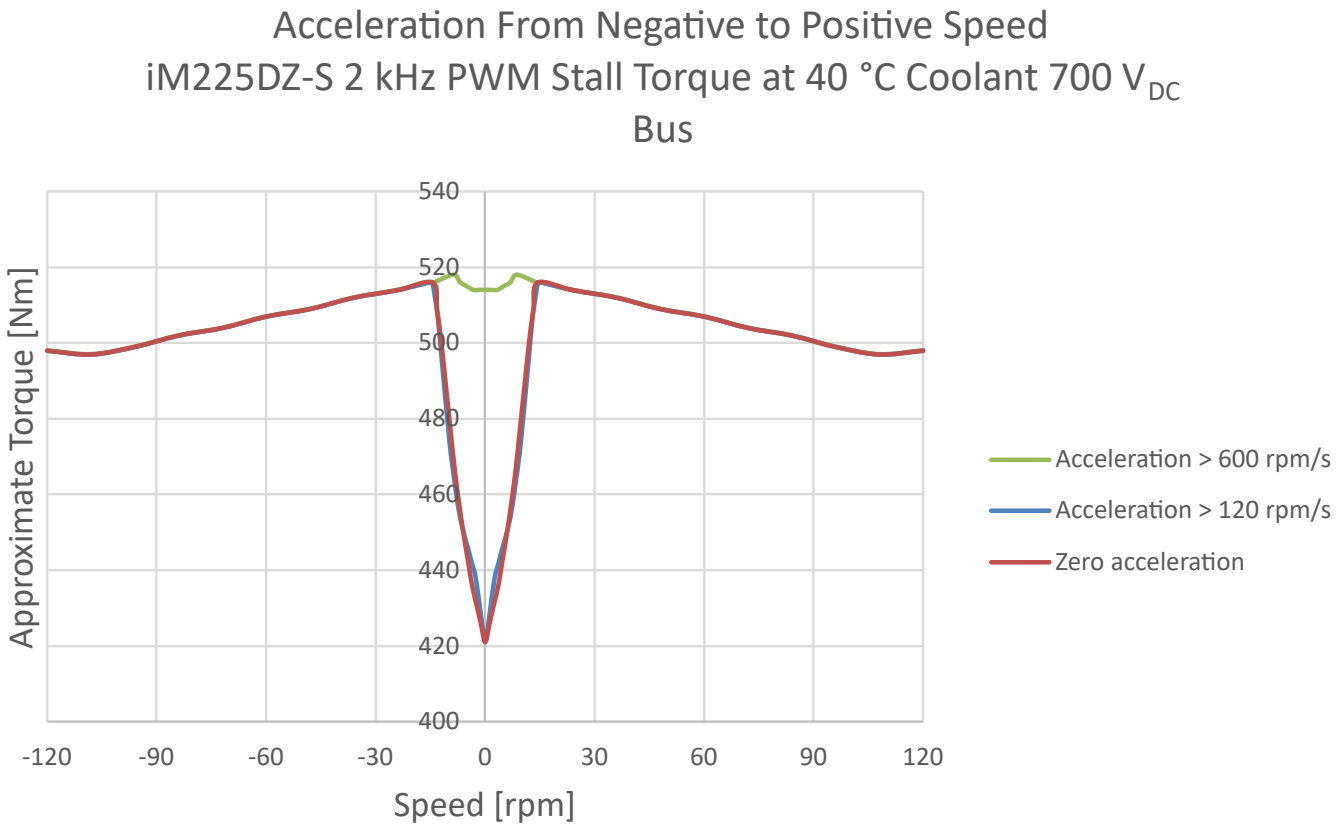


Figure 33: iM225DZ-S Stall Torque vs Speed.

E.3 CM350DZ Stall Burst Performance

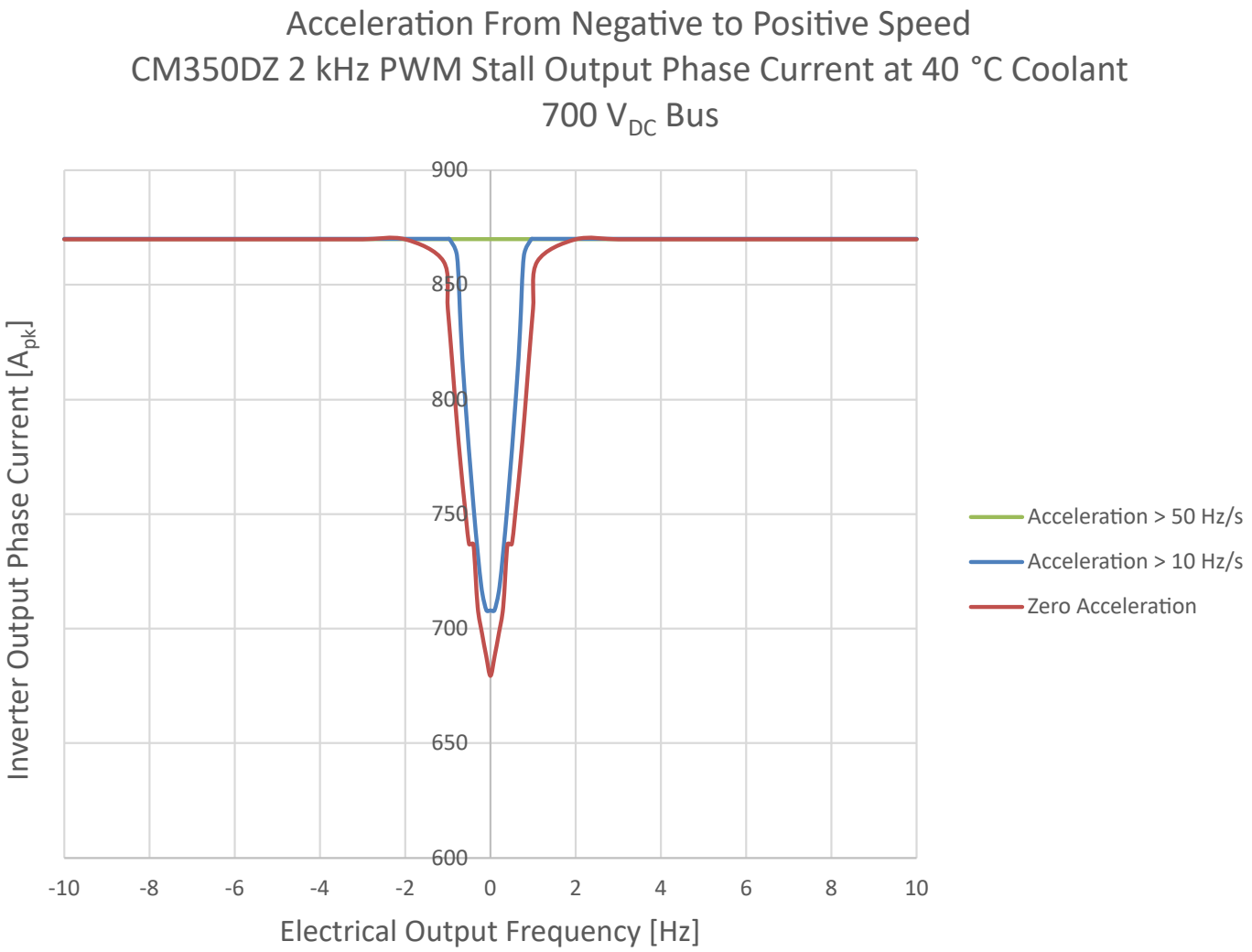


Figure 34: CM350DZ Stall Output Phase Current vs Electrical Output Frequency.

Figure 34 reflects what happens when a vehicle rolls back, then accelerates forward faster than 10 Hz/s and 50 Hz/s. Performance at various constant electrical output frequencies is also shown for comparison. CM350DZ performance when accelerating can be better than constant output frequency performance. Rolling back and accelerating forward minimizes duration at zero speed resulting in better performance while passing through zero speed. Rolling back and ramping torque during the rollback results in better performance because torque is ramping while spinning which results in a lower initial temperature when entering zero speed. Power module junction temperature rises faster and higher when torque is slowly ramped at zero speed instead of when motor is spinning. Vehicles rarely have true zero acceleration when full torque is commanded due to driveline slop. Therefore, performance with acceleration better represents typical performance.

Results will vary depending on the initial rotor position because different rotor positions have different

distributions of current in each phase. The above figure represents worst case initial rotor position.

If held at zero speed, the steady state current is 679.3 A_{pk} for 106 seconds. Current will reduce further to as low as 655 A_{pk} due to hot spot limiting after 106 seconds.

Note, at zero speed phase current is DC, and thus equivalent to A_{pk}. Motor control tables may often list current in A_{RMS}, thus divided the A_{pk} by $\sqrt{2}$ to get the appropriate A_{RMS} for use in lookup tables where necessary.

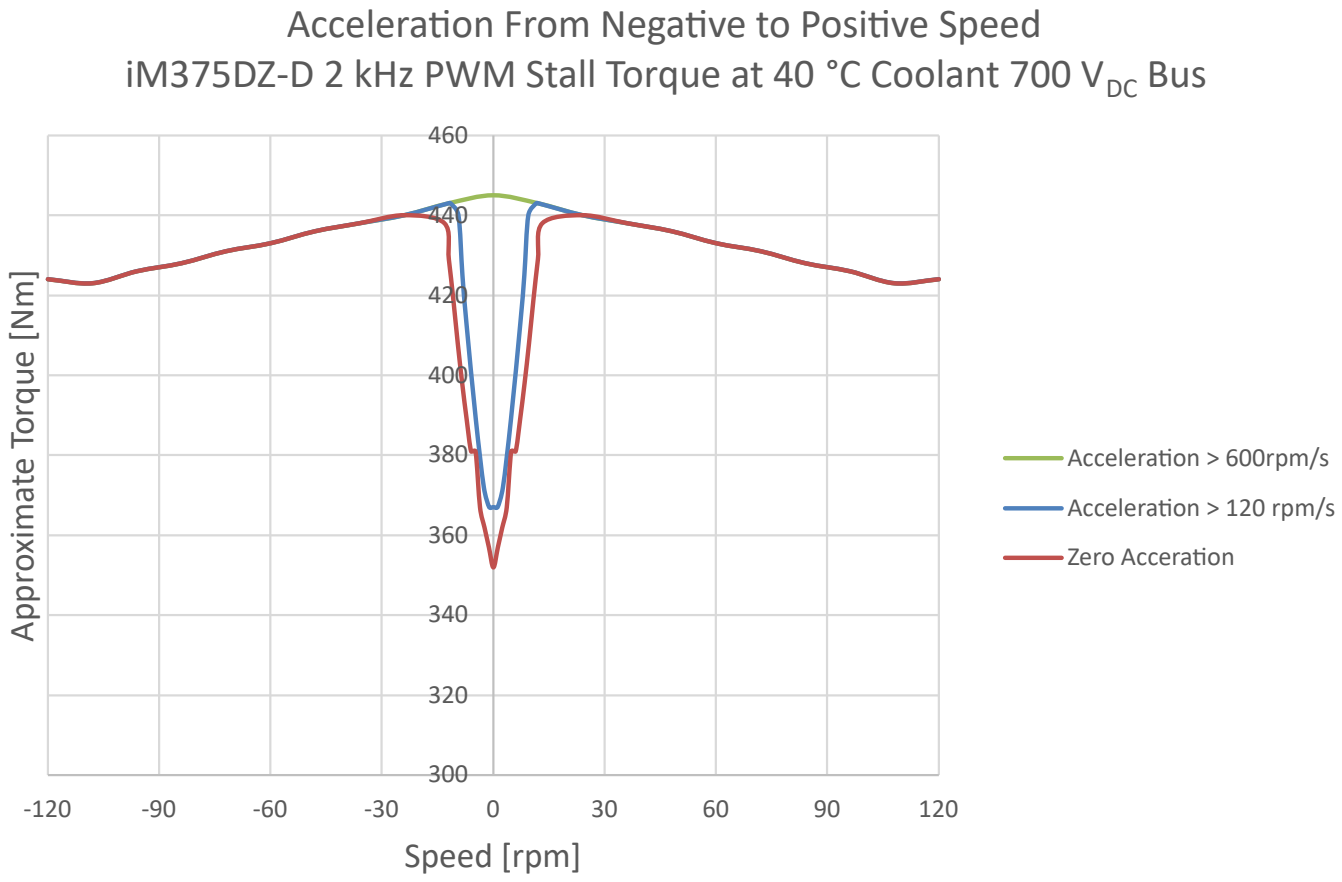


Figure 35: iM375DZ-D Stall Torque vs Speed.

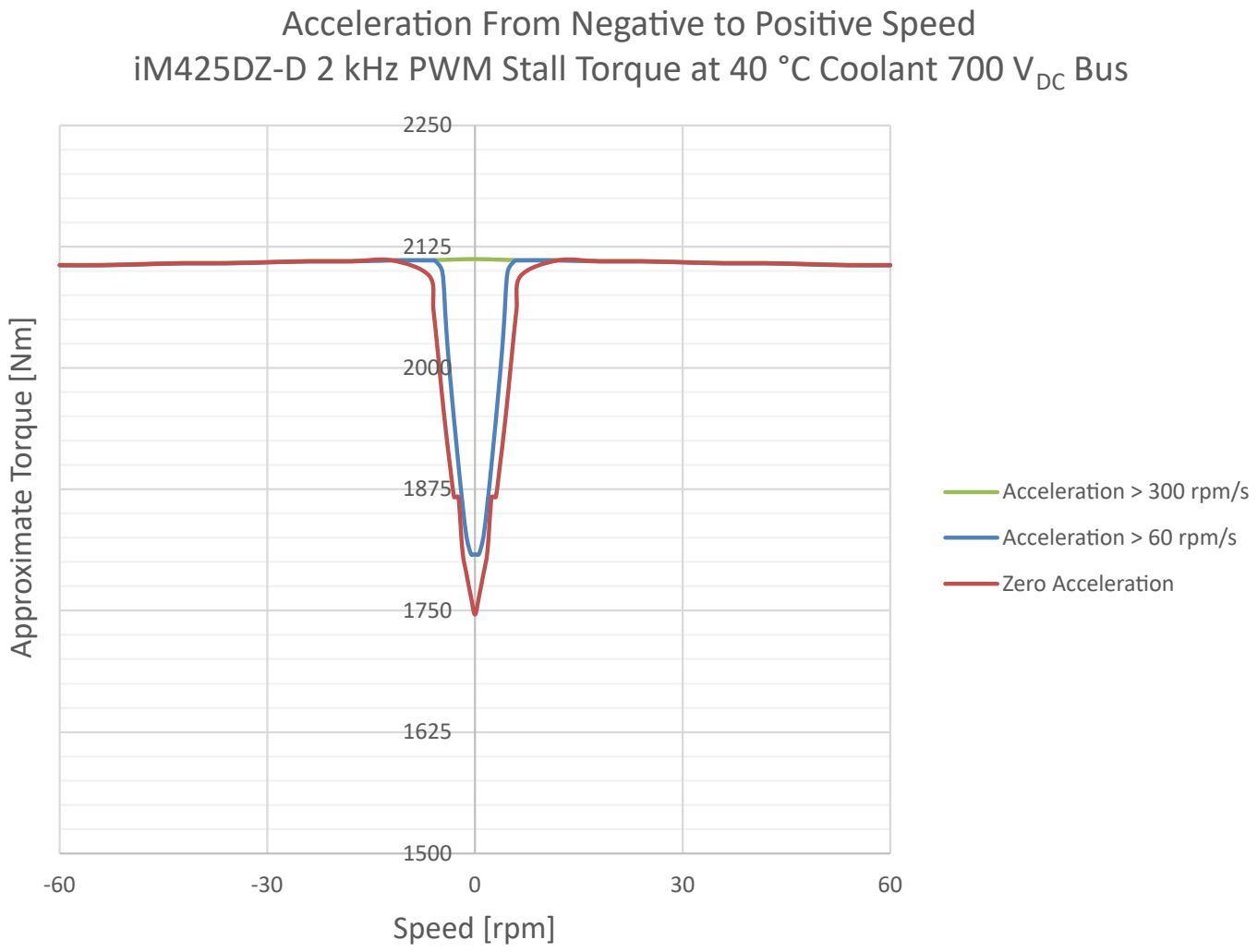


Figure 36: iM425DZ-D Stall Torque vs Speed.

F Appendix: Inverter Hot Spot Performance Data

F.1 CM200DX Hot Spot Performance

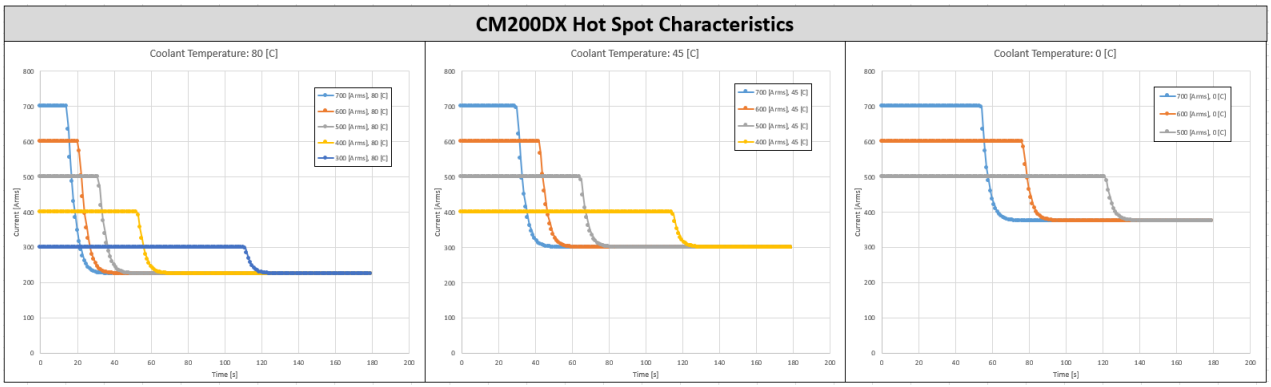


Figure 37: CM200DX Hot Spot Regulator Output Current

Inverter Hot Spot performance varies with coolant temperature and applied current. The above figure 37 shows the expected performance at various current levels and coolant temperatures assuming the inverter starts from rest/low output such that the inverter hot spot model temperature is the same as the coolant temperature.

Note that the maximum inverter current is determined by the PWM frequency, bus voltage, and coolant temperature as well. Depending on the PWM and bus voltage the max current above may not be possible. Refer to Section H for the max inverter current at various input conditions.

F.2 CM200DZ Hot Spot Performance

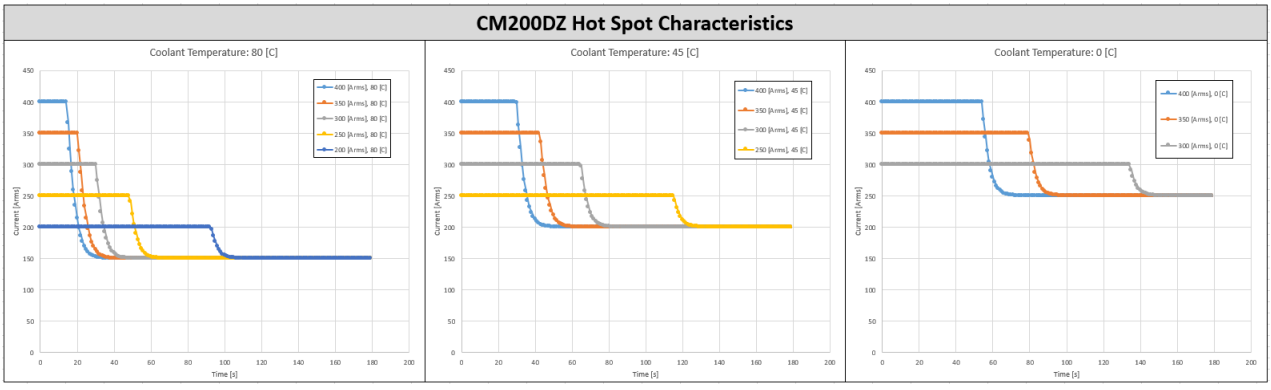


Figure 38: CM200DZ Hot Spot Regulator Output Current

Inverter Hot Spot performance varies with coolant temperature and applied current. The above figure 38 shows the expected performance at various current levels and coolant temperatures assuming the inverter

starts from rest/low output such that the inverter hot spot model temperature is the same as the coolant temperature.

Note that the maximum inverter current is determined by the PWM frequency, bus voltage, and coolant temperature as well. Depending on the PWM and bus voltage the max current above may not be possible. Refer to Section H for the max inverter current at various input conditions.

F.3 CM350DZ Hot Spot Performance

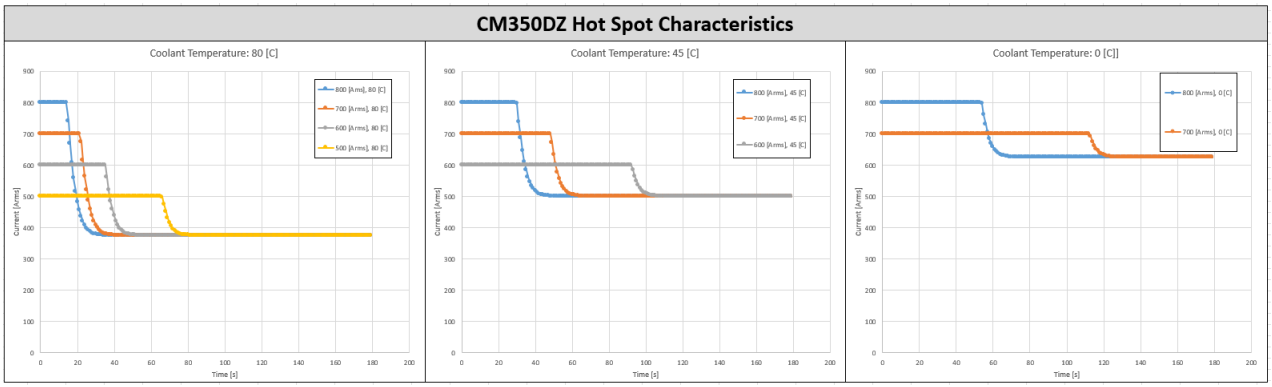


Figure 39: CM350DZ Hot Spot Regulator Output Current

Inverter Hot Spot performance varies with coolant temperature and applied current. The above figure 39 shows the expected performance at various current levels and coolant temperatures assuming the inverter starts from rest/low output such that the inverter hot spot model temperature is the same as the coolant temperature.

Note that the maximum inverter current is determined by the PWM frequency, bus voltage, and coolant temperature as well. Depending on the PWM and bus voltage the max current above may not be possible. Refer to Section H for the max inverter current at various input conditions.

F.4 CM350SiC Hot Spot Performance

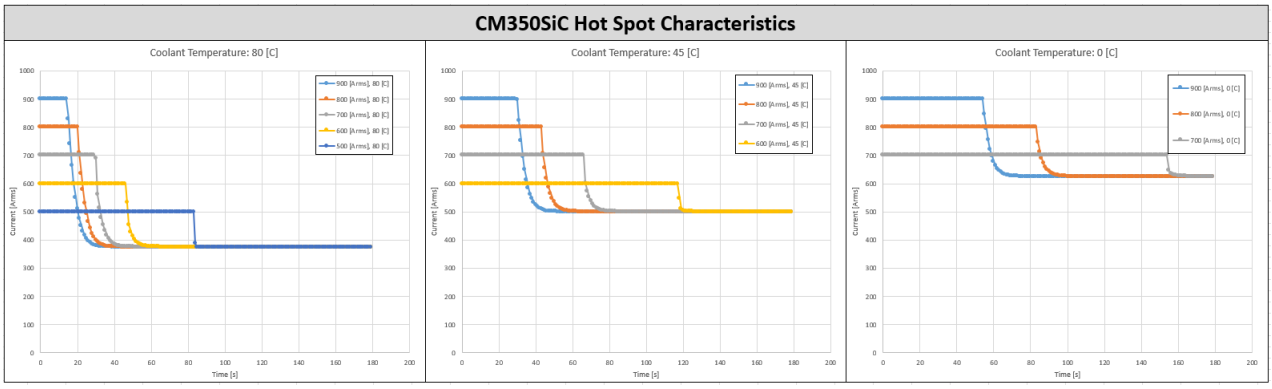


Figure 40: CM350SiC Hot Spot Regulator Output Current

Inverter Hot Spot performance varies with coolant temperature and applied current. The above figure 40 shows the expected performance at various current levels and coolant temperatures assuming the inverter starts from rest/low output such that the inverter hot spot model temperature is the same as the coolant temperature.

Note that the maximum inverter current is determined by the PWM frequency, bus voltage, and coolant temperature as well. Depending on the PWM and bus voltage the max current above may not be possible. Refer to Section H for the max inverter current at various input conditions.

G Appendix: Motor Hot Spot Performance Data

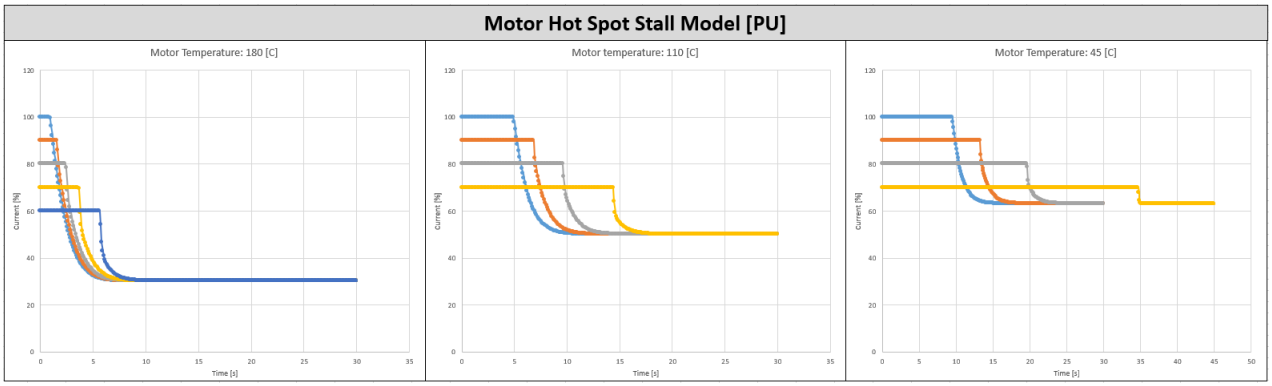


Figure 41: Motor Hot Spot Regulator Output Current

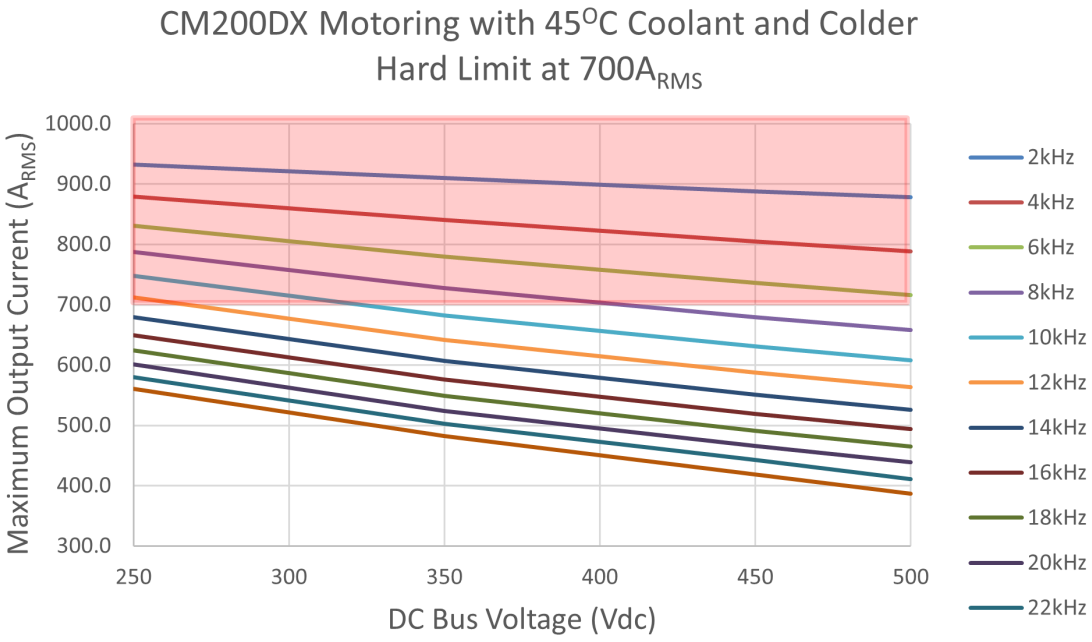
Motor Hot Spot performance varies with motor temperature and applied current. The above figure 41 shows the expected performance at various current levels and motor temperatures assuming the drive system enters stall at the specified motor temperature. The current possible in the motor is expressed in a per unit value of the maximum motor current, specified in section 6.6.

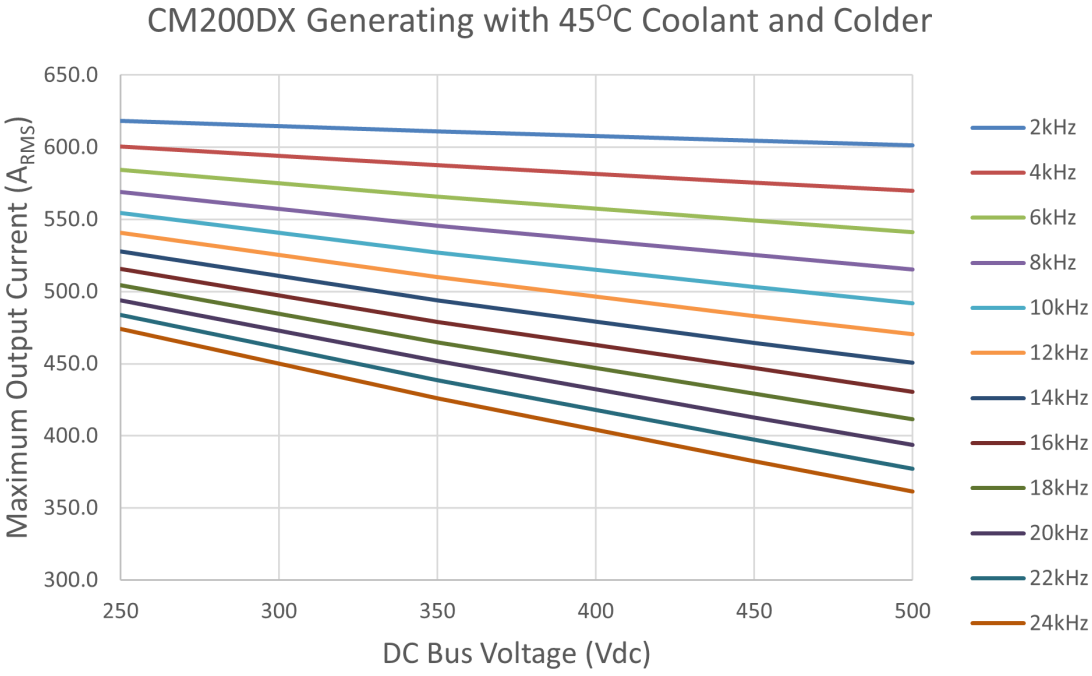
Note that this model is only active during stall.

H Appendix: Maximum Output Current Performance Data

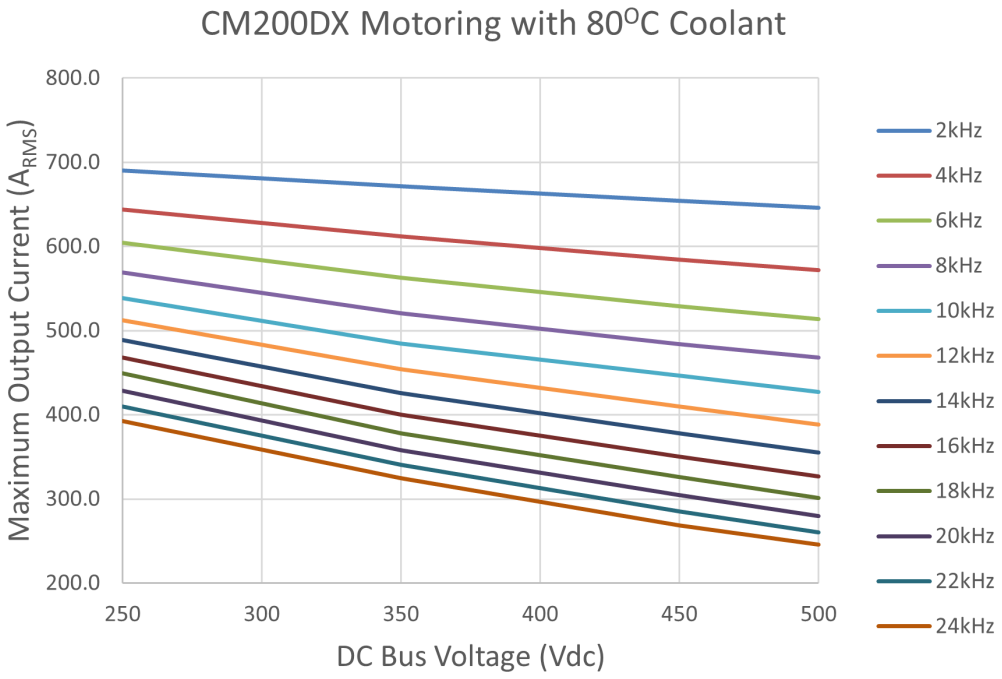
H.1 CM200DX Maximum Current Output

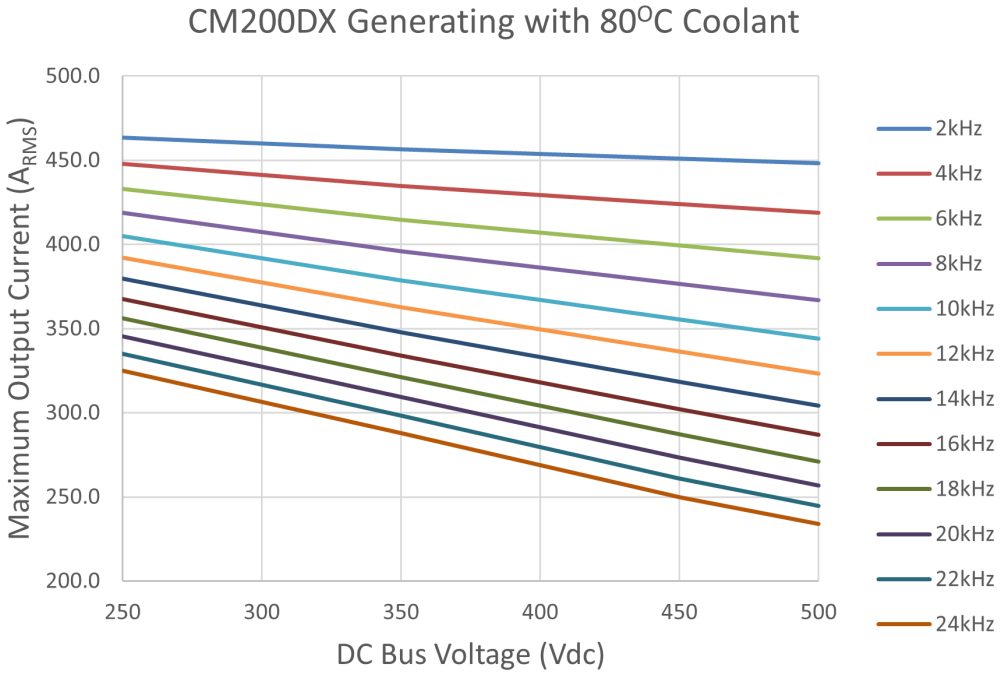
H.1.1 Coolant Temperature Less Than o Equal to 45°C





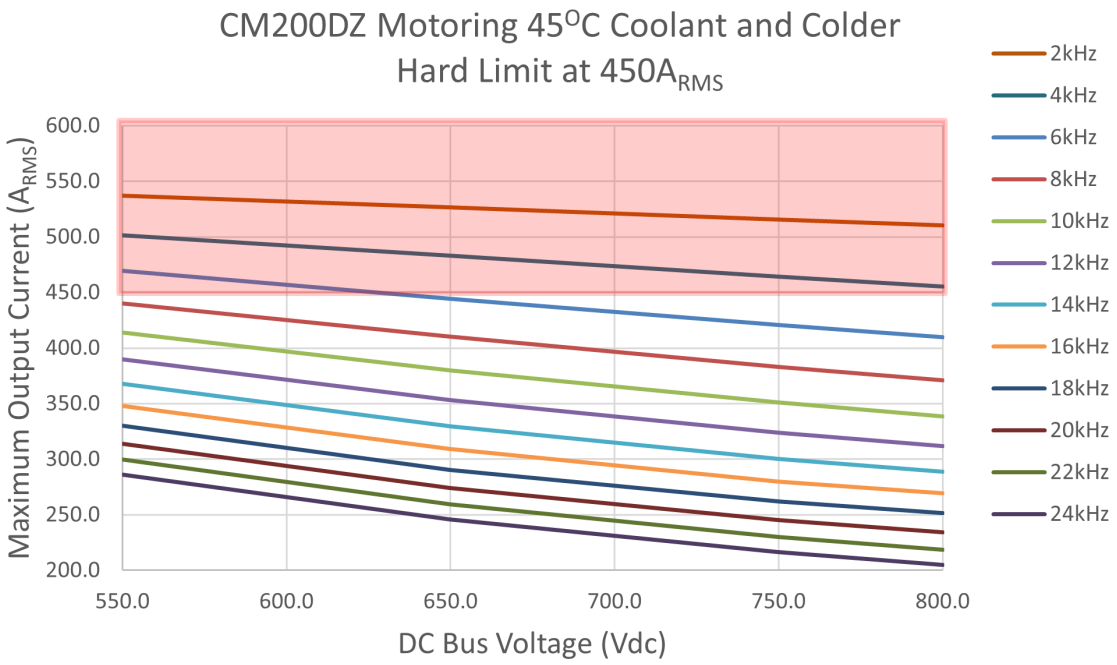
H.1.2 Coolant Temperature at 80°C

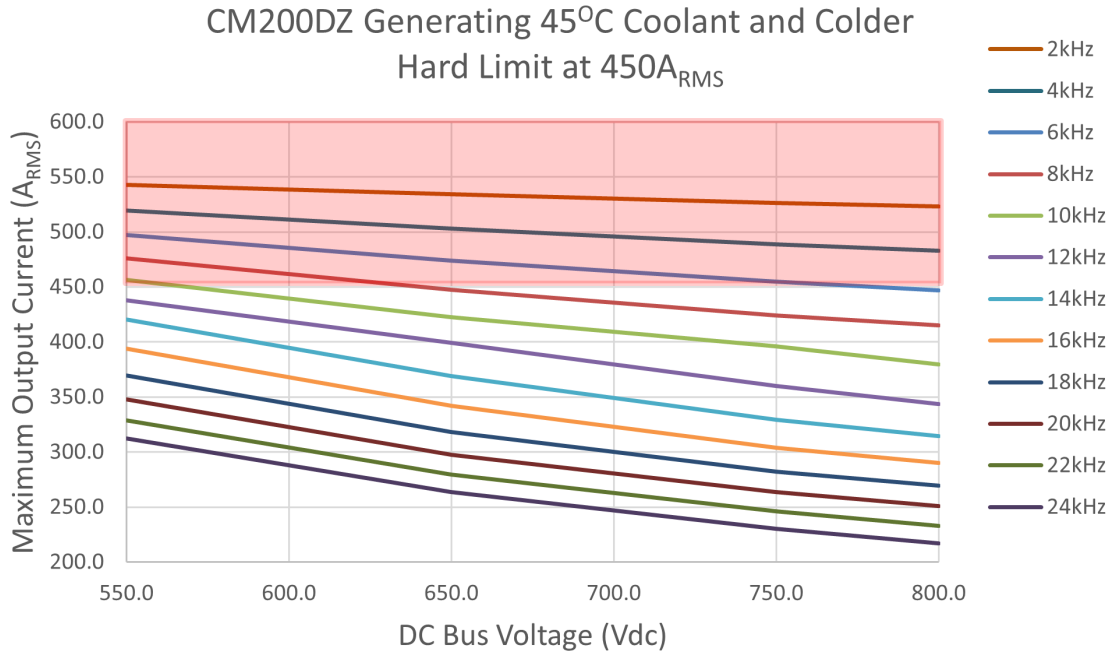




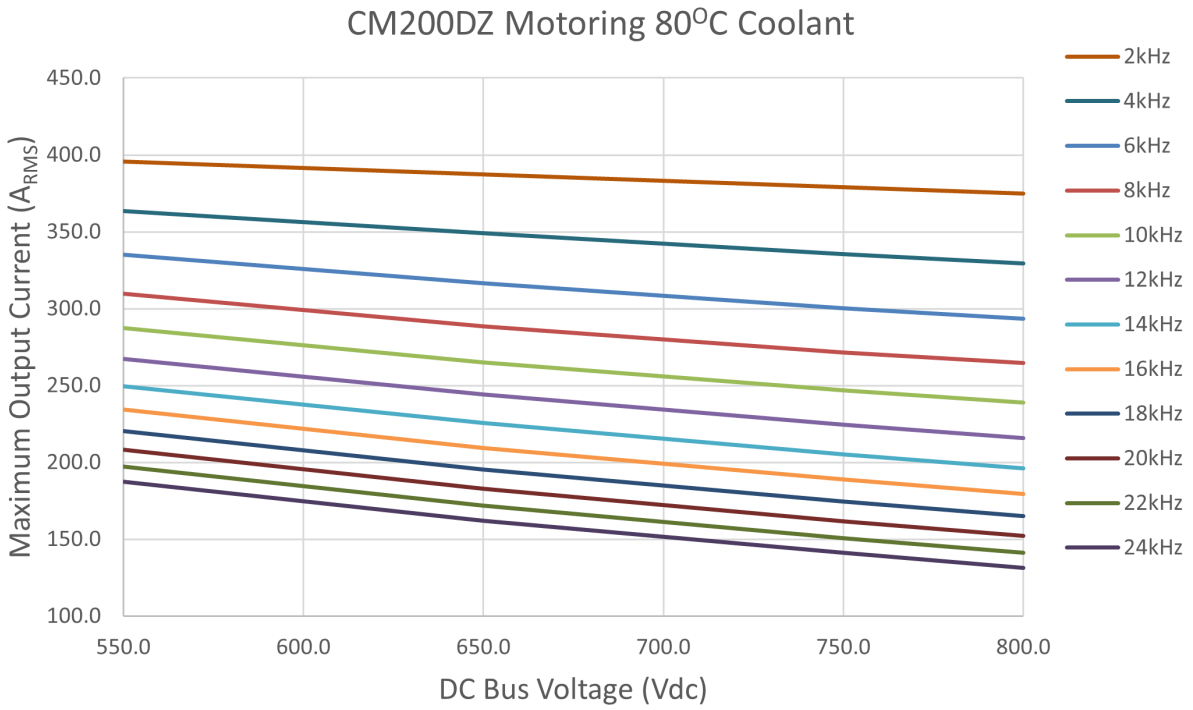
H.2 CM200DZ Maximum Current Output

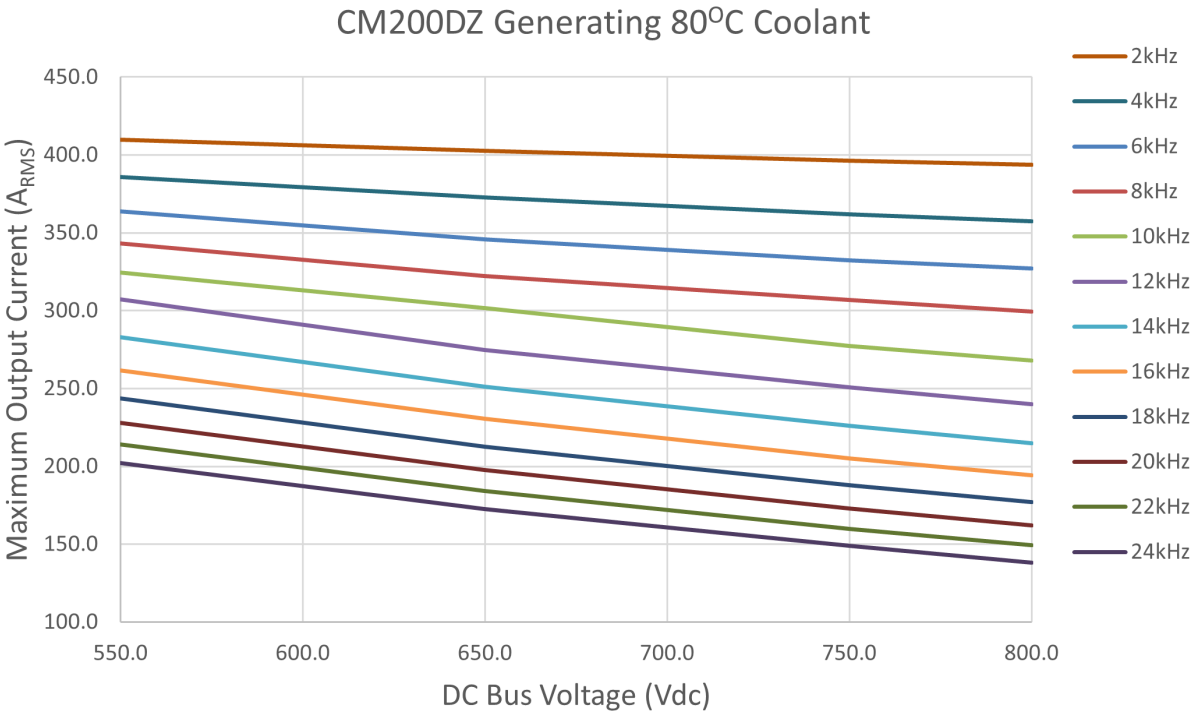
H.2.1 Coolant Temperature Less Than o Equal to 45°C





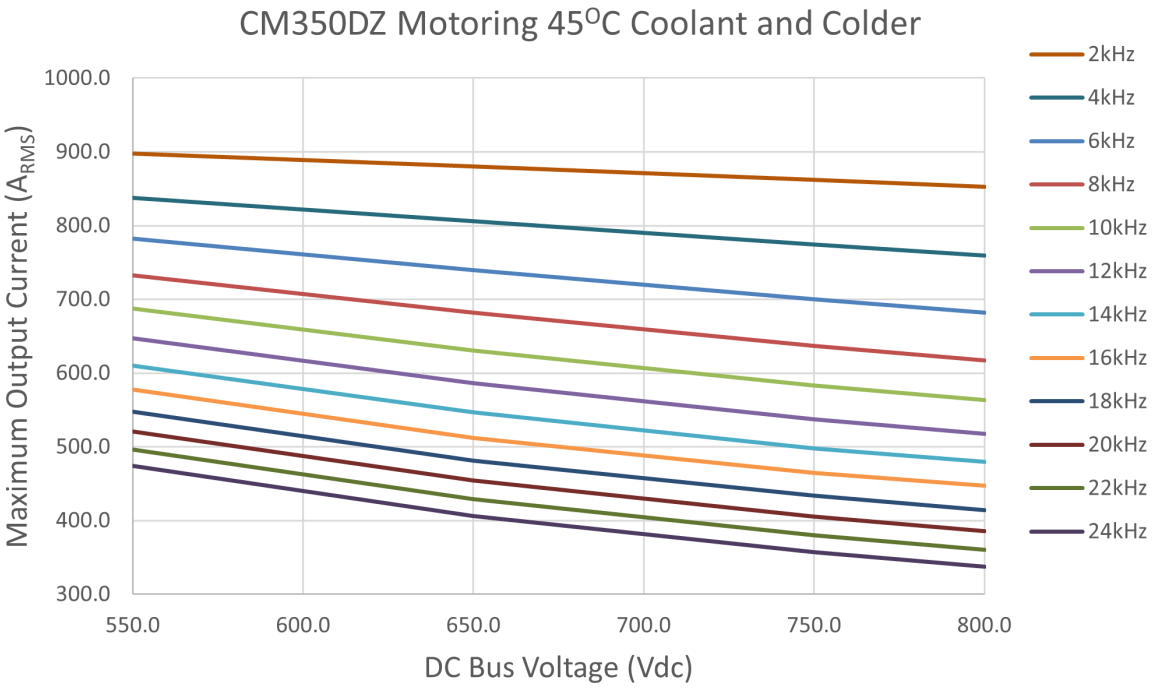
H.2.2 Coolant Temperature at 80°C

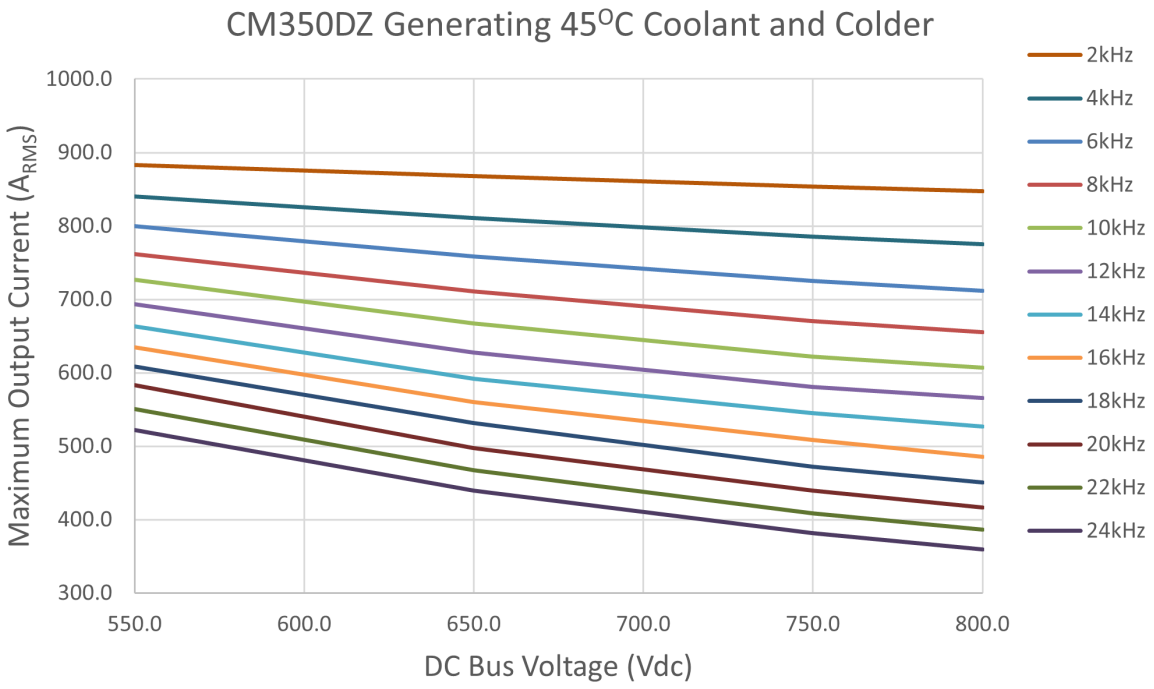




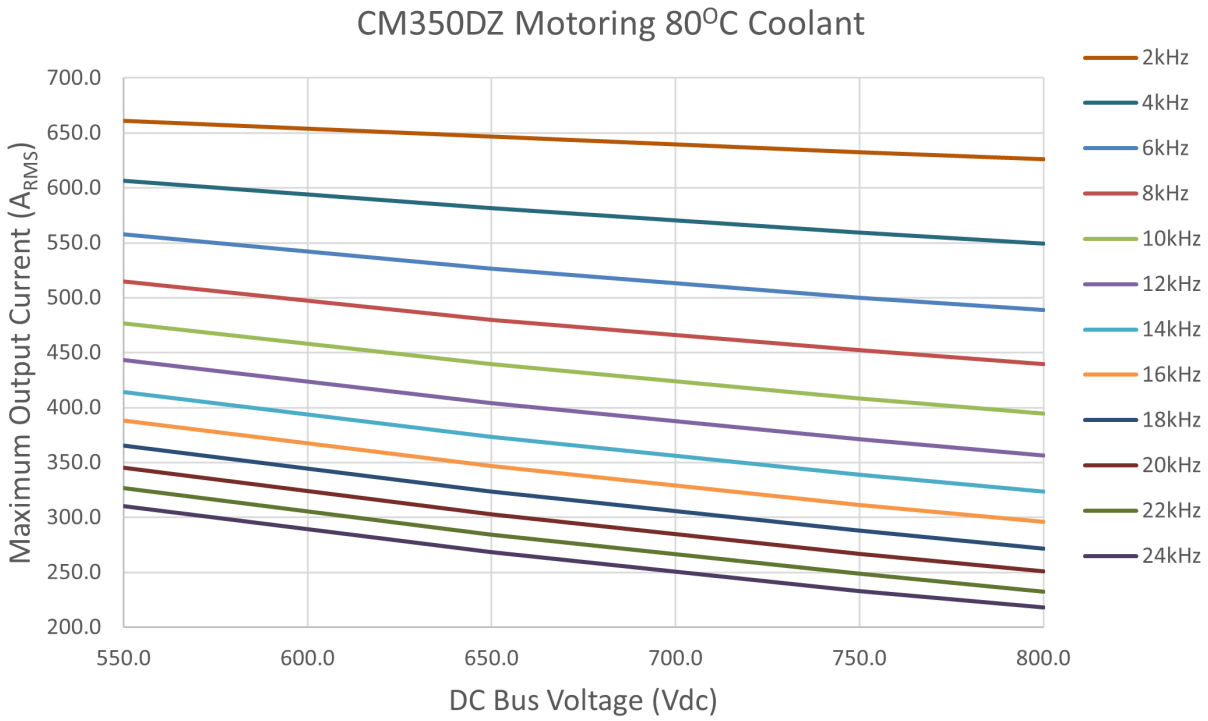
H.3 CM350DZ Maximum Current Output

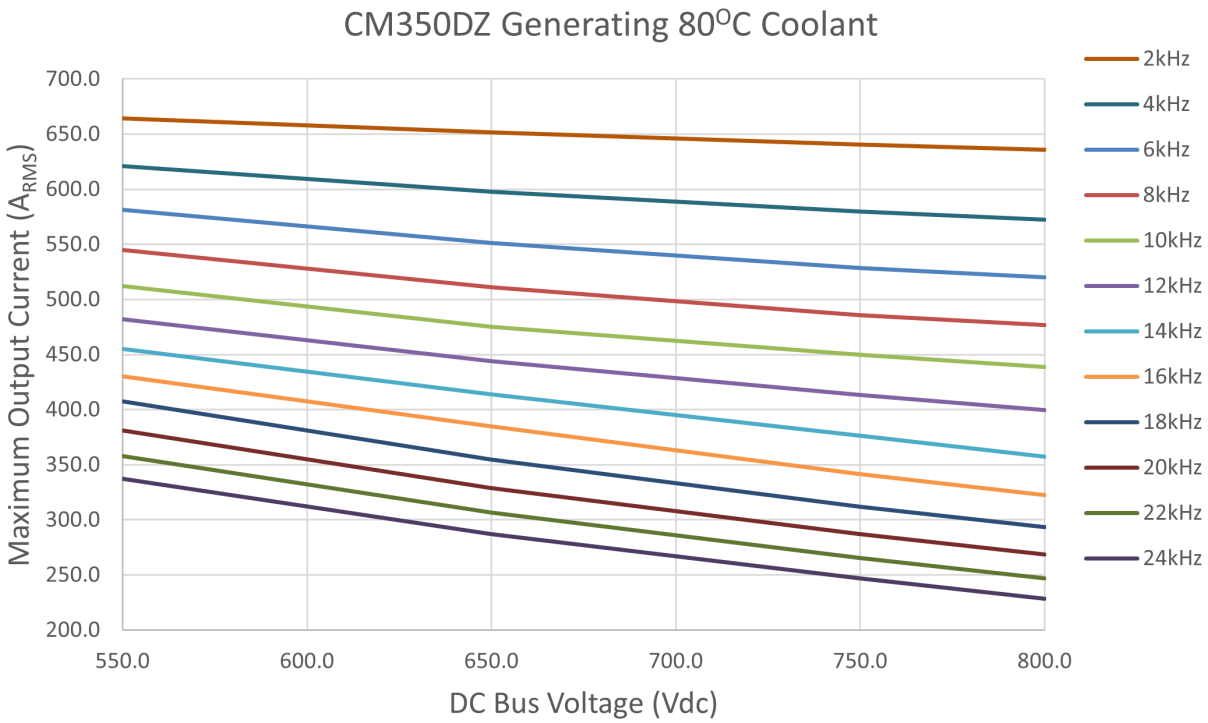
H.3.1 Coolant Temperature Less Than o Equal to 45°C





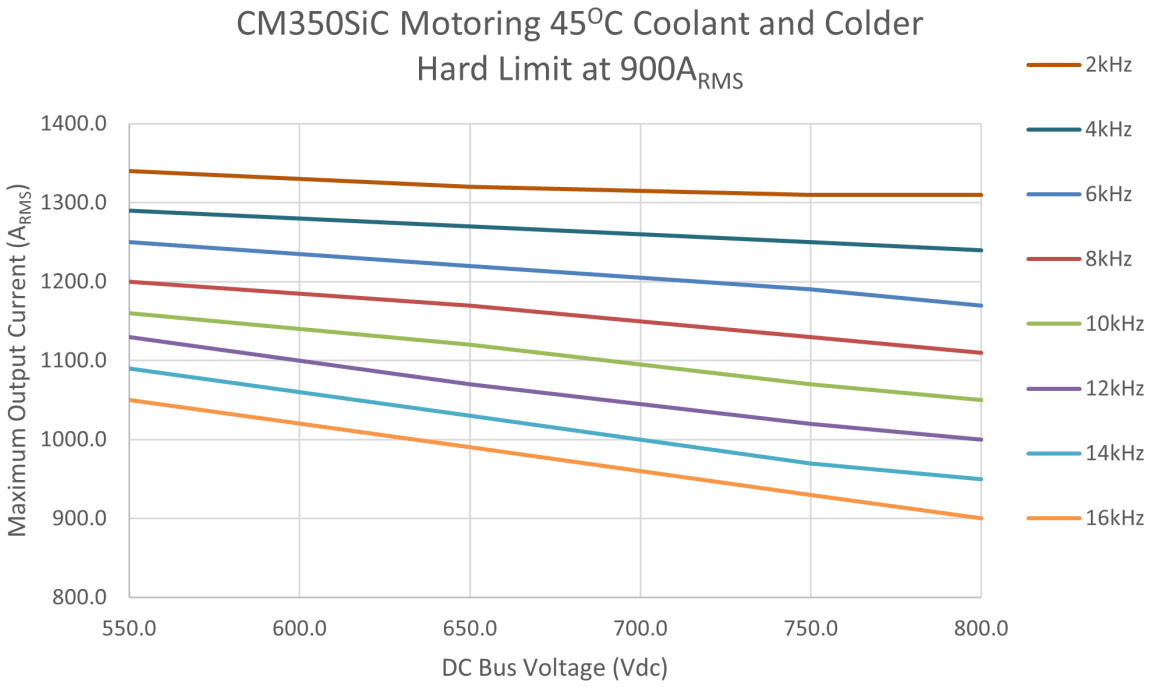
H.3.2 Coolant Temperature at 80°C

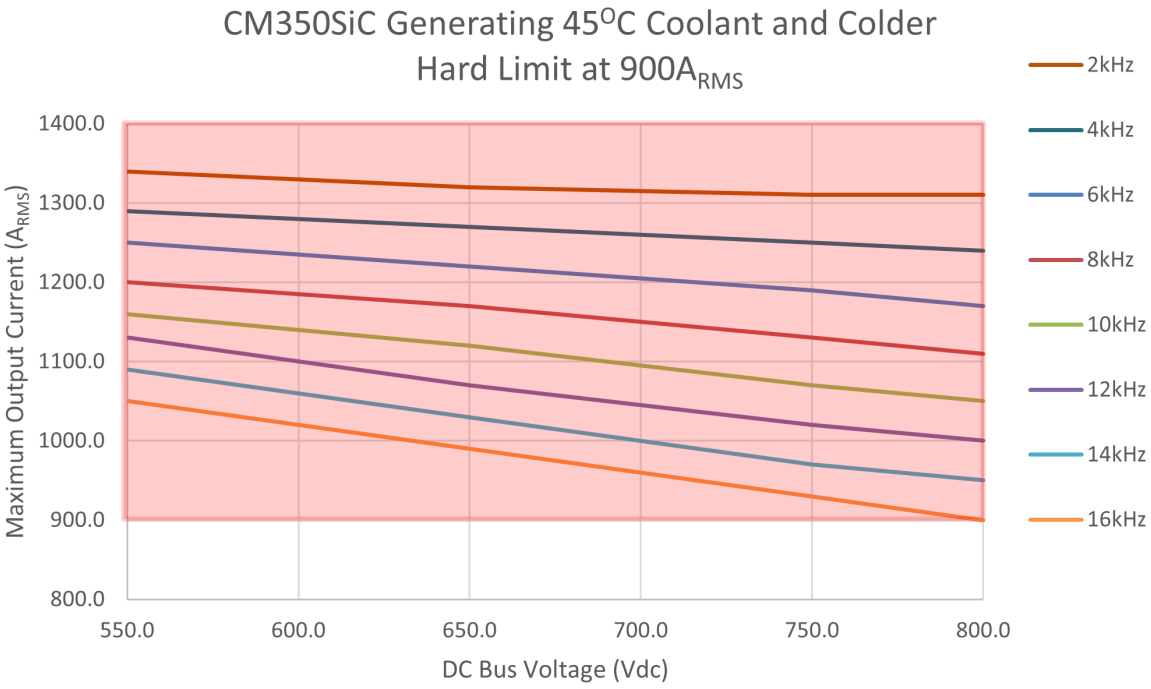




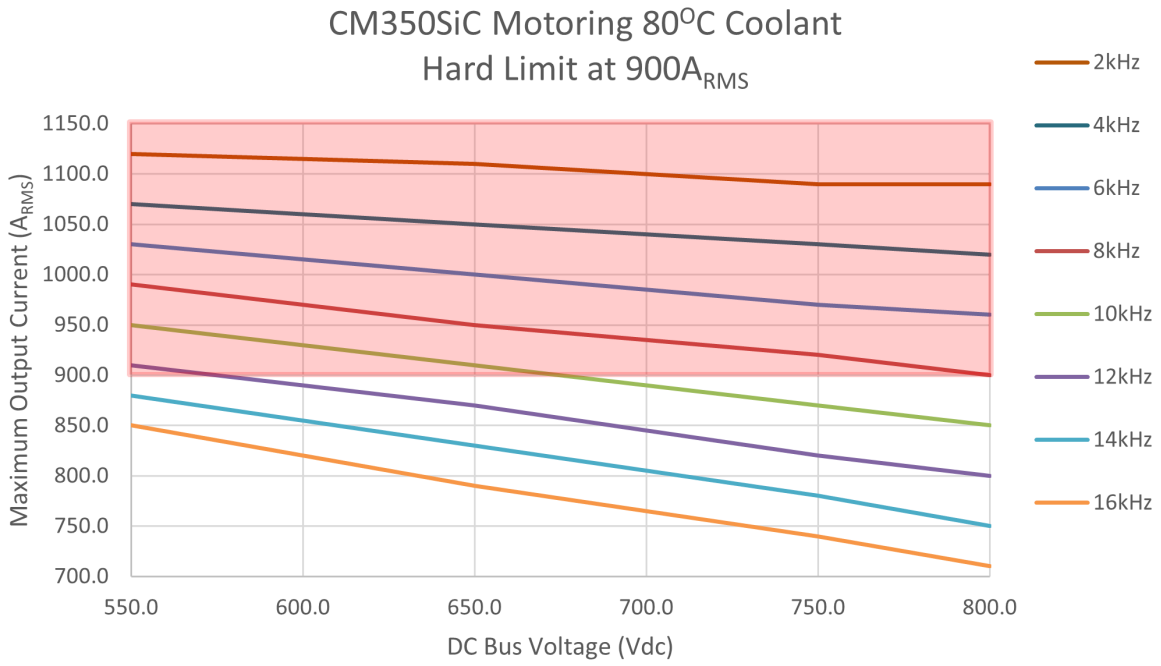
H.4 CM350SiC Maximum Current Output

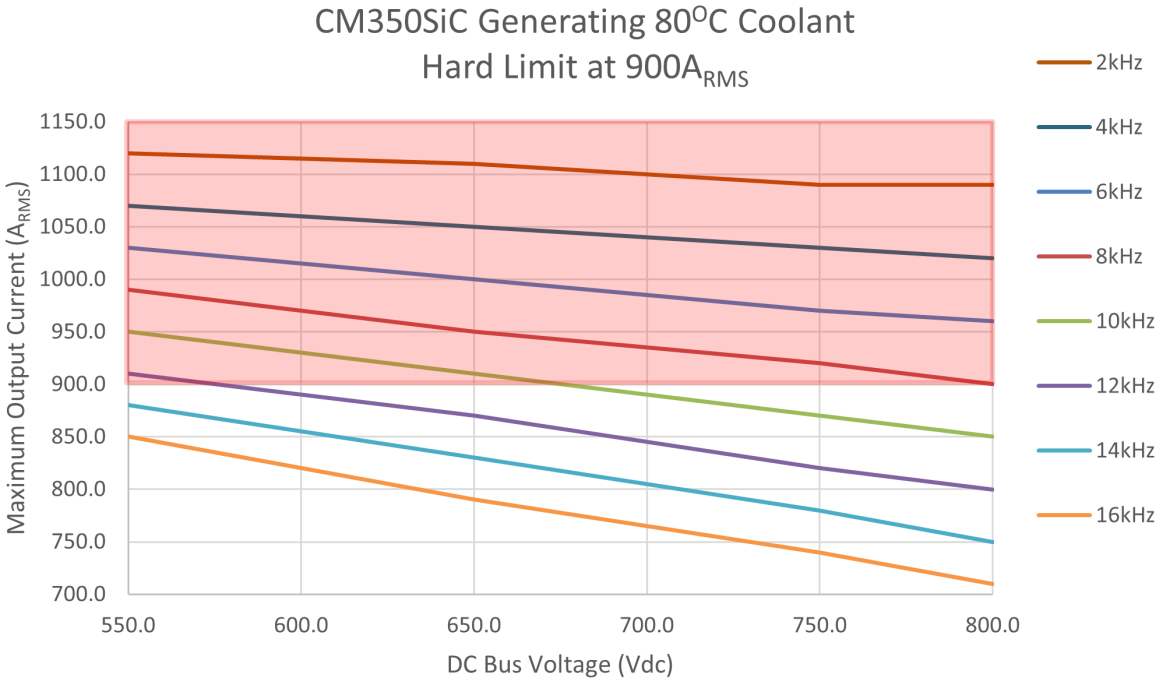
H.4.1 Coolant Temperature Less Than o Equal to 45°C





H.4.2 Coolant Temperature at 80°C





I Appendix: SCI Data Acquisition Guide

I.1 Required Hardware

RS232 cable or RS232-USB Adapter (based on PC's port availability)

I.2 Required Software and Configuration

To be able to capture the SCI data is necessary to use a program that can capture data from the RS232 port of the inverter. An example of this software is Realterm. The software should be configured for the COM port that is being used by the inverter and with the RS232 settings shown in the table below:

Baud Rate	57600
Parity	None
Data Bits	8
Stop Bits	1
Hardware Flow Control	None

Table 74: SCI Settings

I.2.1 Data Records

Each parameter is 16-bits long and each nibble (4-bits) in a parameter is sent as an ASCII character. A 'record' consists of total five characters, that is, the four nibbles in a parameter and a space character. After sending all records, two additional characters, a carriage return and a linefeed, are sent.

0	0	6	8	<space>

Table 75: Example Data Record

Data Record 1	Data Record 2	Data Record 3	Data Record N	<cr>	<lf>

Table 76: A complete set of data records

I.2.2 Update Rate

The inverter will send out records at a periodic rate. The rate is dependent on the number of data records in each set and the availability of processor resources.

I.3 Data Acquisition Parameters

The following data records are transmitted over the serial bus:

Count	Parameter
1	The low word of the Power On Timer (increments every 3ms)
2	Filtered Accel-pot input voltage (V) times 100
3	Motor Torque feedback (Nm) times 10
4	Vehicle Torque Command (Nm) times 10
5	DC Voltage (V) times 10
6	DC Current (V) times 10
7	Motor Speed (rpm)
8	Flux Weakening Regulator Output (Apk) times 10
9	Motor Voltage Magnitude (Vpk) times 10
10	IQ Command (Apk) times 10
11	IQ Feedback (Apk) times 10
12	ID Command (Apk) times 10
13	ID Feedback (Apk) times 10
14	Modulation times 10000
15	Module A Temperature (°C) times 10
16	Motor Temperature (°C) times 10
17	Run Fault Low Word
18	Run Fault High Word
19	Torque Shudder (Nm) times 10
20	Filtered Brake pot (V) times 100

Table 77: SCI Settings

I.3.1 Utilizing the Captured Data

Once the data is captured in a text file, it should be imported into a Microsoft Excel spreadsheet as space delimited data. After importing all data, it can be copied into the *SCI Template.xls* spreadsheet which provides conversion formulae for each data record and allows the user to plot graphs to analyze the vehicle performance in more detail.

J Revision History

Version	Description	Updated By	Date
0A-0163-01	Initial Content	Noah Erickson	10/23/2024
0A-0163-02	Updates for 6532 release, ECO 1106.	Noah Erickson	2/6/2025