User's Guide
MCF8316A Tuning Guide

TEXAS INSTRUMENTS

ABSTRACT

This Tuning guide provided step by step guidence to setup MCx8316AEVM, connect MCF8316A GUI to the EVM and tune a 3-phase Brushless DC motor using MCF8316A.

Table of Contents

List of Figures

List of Tables

1 Revision History

NOTE: Page numbers for previous revisions may differ from page numbers in the current version.

Trademarks

All trademarks are the property of their respective owners.

2 Introduction

The MCF8316A provides three half-H-bridge integrated MOSFET drivers with sensorless FOC control to drive a three-phase brushless DC (BLDC) motor at 12-V/24-V DC rails or battery powered applications. The device integrates three current-sense amplifiers (CSA) with integrated current sense for sensing the three phase currents of BLDC motors to achieve optimum FOC control. Simplified schematic of MCF8316A is shown in Figure 2-1.
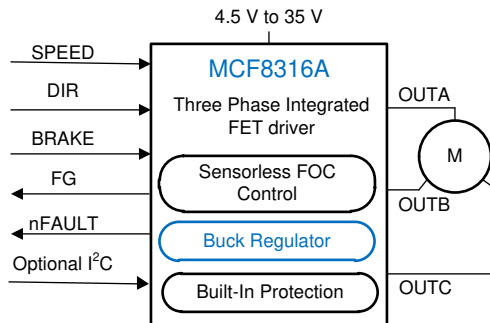
Figure 2-1. Simplified Schematic of MCF8316A

This tuning guide provides the steps to tune a 3-phase BLDC motor using the MCF8316A. The tuning process is classified into two sections: Essential controls and Basic controls. This process is also detailed in the Guided tuning section in the GUI.

- *Essential controls:* Tuning steps to successfully spin the motor in closed loop.
- *Basic controls:* Tuning steps to conform to use-case and explore features in the device.

2.1 Hardware and GUI Setup

Below are the items that are required to run through this tuning guide.

1. MCx8316AEVM Board.
2. A computer with the MCF8316A_GUI installed.
3. A motor to be tuned using this process. It will be good to have motor datasheet but it is not mandatory.
4. A DC Power Supply rated for the motor.
5. Basic lab equipment such as a Digital Multimeter (DMM), Oscilloscope, current probe and voltage probes.

2.1.1 Jumper Configuration

Ensure the hardware is configured according to the jumper configuration as shown in Figure 2-2. Red boxes indicate where jumpers should be installed on the EVM.

Figure 2-2. MCx8316A EVM Jumper Configuration

2.1.2 External Connections

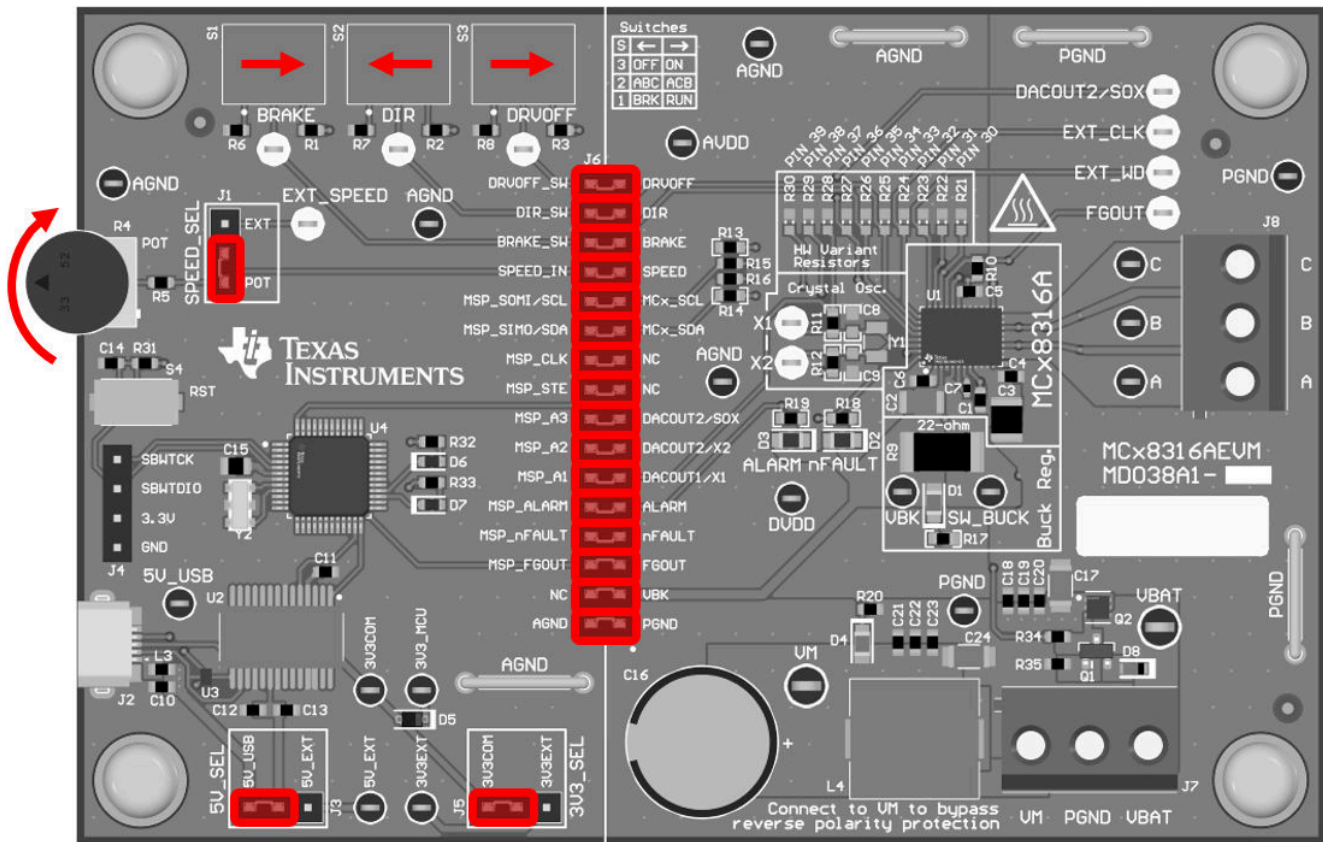Connect the motor to J8 (connections A, B, and C). If the motor has a center tap connection or wires for Hall-effect sensors, leave these wires unconnected. Supply a voltage compliant with the Power Supply Voltage (VM) range using J7. Recommended voltage range for the device is 4.5 V – 35 V.
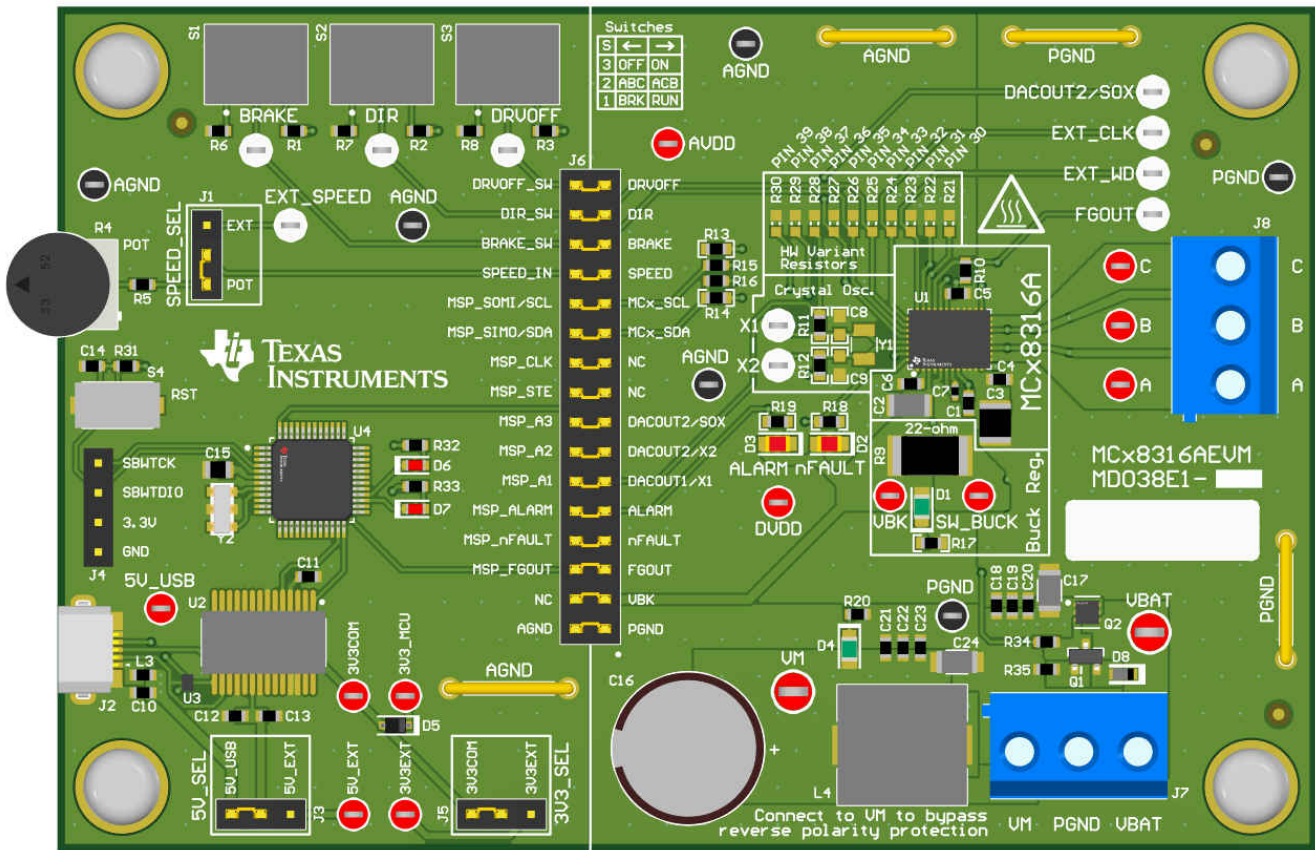
Figure 2-3. MCx8316A EVM External Configuration

2.1.3 Connecting to the GUI

2.1.3.1 Connect to computer

Ensure the EVM is powered on (VM) before connecting USB cable. Plug in the micro-USB, from the computer that holds the GUI, to the USB Connector (J2) on the EVM.

2.1.3.2 Connect to the GUI

The GUI should automatically connect to the EVM, as shown on the status bar located on the bottom of this GUI.

If the GUI doesn't connect automatically, please ensure that the "FTDI" serial port is selected by going to the top menu "Options -> Serial Port ...". Then click on the chain link icon on the left of the status bar to connect to the EVM.

2.1.3.3 Verify Hardware Connection

After the GUI establishes connection with the EVM through USB, you can verify the communication with the MCx8316 device by clicking on the "Read All Registers" button on the top right of this GUI. After a few seconds, the GUI should indicate with a pop-up notification that all registers have been read back successfully.

3 Essential Controls

The goal of this section is to help customers spin their motor successfully in closed loop. This section provides standardized steps to tune parameters for successful Motor spin-up in closed loop.

Closed loop is defined as the sensorless closed loop where motor spins at the commanded speed.

3.1 Recommended Default Values

Launch the MCF8316A EVM GUI. Load the recommended default values that are listed in Table 3-1.

Table 3-1. Recommended Default Values

Address Name	Address	Recommended Value
ISD_CONFIG	0x00000080	0x44638C20
REV_DRIVE_CONFIG	0x00000082	0x283AF064
MOTOR_STARTUP1	0x00000084	0x0B6807D0
MOTOR_STARTUP2	0x00000086	0x23066000
CLOSED_LOOP1	0x00000088	0x0C3181B0
CLOSED_LOOP2	0x0000008A	0x1AAD0000
CLOSED_LOOP3	0x0000008C	0x00000000
CLOSED_LOOP4	0x0000008E	0x0000012C
SPEED_PROFILES1	0x00000094	0x00000000
SPEED_PROFILES2	0x00000096	0x00000000
SPEED_PROFILES3	0x00000098	0x00000000
SPEED_PROFILES4	0x0000009A	0x00000000
SPEED_PROFILES5	0x0000009C	0x00000000
SPEED_PROFILES6	0x0000009E	0x00000000
FAULT_CONFIG1	0x00000090	0x5FE80206
FAULT_CONFIG2	0x00000092	0x74000000
PIN_CONFIG	0x000000A4	0x00000000
DEVICE_CONFIG1	0x000000A6	0x00000000
DEVICE_CONFIG2	0x000000A8	0x0000B000
PERI_CONFIG1	0x000000AA	0x40000000
GD_CONFIG1	0x000000AC	0x00000100
GD_CONFIG2	0x000000AE	0x00200000
ALGO_CTRL1	0x000000EA	0x00000000
INT_ALGO_1	0x000000A0	0x00B3407D
INT_ALGO_2	0x000000A2	0x000001A7

3.2 Device and Pin Configuration

3.2.1 Speed Input Mode

MCF8316A provides four options to configure the input speed command.

- PWM input on SPEED pin by varying duty cycle of input signal
- Frequency input on SPEED pin by varying frequency of input signal
- Analog input on SPEED pin by varying amplitude of input signal
- Over I^{2}C by configuring SPEED_CTRL

Configure the speed input [SPEED_MODE] register to select the appropriate speed input mode.

If PWM input on SPEED pin is chosen as the speed input, configure SPEED_RANGE_SEL to select the PWM frequency range.

If Frequency input on SPEED pin is chosen as the speed input, configure INPUT_MAXIMUM_FREQ to select the maximum PWM frequency. For example, if 10 kHz is chosen as maximum PWM frequency, 10 kHz PWM signal on SPEED pin will correspond to 100% speed command.

To command the speed via I²C, toggle the "Enable Speed Control via I²C" in the I²C Control Section in GUI.

3.3 Control Configuration - Motor Parameters

3.3.1 Maximum Motor Electrical Speed (Hz)

Using the motor's datasheet, the user can input the maximum motor electrical speed in Hz. If this data is not available, the user can input the number of pole pairs and motor mechanical speed in RPM. The GUI will convert the motor mechanical speed in RPM to motor electrical speed in Hz using Equation 1 .

$$f_{Electrical} = \frac{n_{PolePairs} \cdot \omega_{Mechanical}}{60} \tag{1}$$

Where:

- $\omega_{Mechanical}$ is the mechanical speed in units revolutions per minute (RPM)
- $f_{Electrical}$ is the electrical speed in units of hertz (Hz)
- $n_{PolePairs}$ is the number of motor pole pairs

Note

Determining number of motor poles without a motor datasheet:

1. Use a lab power supply and make sure the current limit is set to less than the motor rated current. Do not turn on the supply
2. Connect V+ of the supply to phase A and V- of the supply to phase B of the motor. Any 2 of the 3 phases can be chosen at random if they are not labeled
3. Turn on supply The rotor should have settled at one position with the injecting current.
4. Manually rotate the rotor until rotor snaps to another settle position. It will have several settle-down positions around one mechanical cycle
5. Count the number of settle-down positions for one fully mechanical cycle, which is the number of pole pairs. Multiplying by two calculates the number of poles.

Be careful of gearing systems within a motor. The gearing ratio will determine how many rotor revolutions correlate to the shaft's mechanical revolution.

3.4 Control configuration - Closed Loop

3.4.1 Current Limit for Torque PI Loop

The current limit for torque PI loop will be same as the rated peak phase current of the motor. Rated peak phase current is the peak phase current drawn by the motor at rated torque of the motor. Check the motor datasheet for the rated peak phase current of the motor in Amps. The range of current limit values that can be configured in MCF8316A is between 0.125 A and 8 A.

If your motor current rating falls between two dropdown settings, then choose the higher setting.

3.5 Testing for Successful Startup into Closed Loop

1. *Apply a nonzero speed command*

   a. If the Speed input mode is I2C, change the "I2C Speed Command Percentage" to a non-zero value. Once the speed command is issued, the device will start commutating and the motor should start spinning that is proportional to the I2C Speed Command Percentage.

    b.   If the speed input mode is Analog, rotate the Speed Control potentiometer (R4) clockwise to control the motor speed. Once the speed command is issued, the device will start commutating and the motor should start spinning that is proportional to the voltage on the Speed Pin.

    c.   If the speed input mode is PWM, apply PWM signal to the speed pin with a given duty cycle with a low of 0 V to a high of 2.2 V (min). Once the speed command is issued, the device will start commutating and the motor should start spinning that is proportional to the duty cycle.

2. *Wait for MPET to estimate motor electrical parameters.*

    The Motor Parameter Extraction Tool (MPET) will attempt to measure the motor resistance, inductance, and Motor BEMF constant (Ke) and populate these parameters automatically during the initial position detect (IPD) and open loop stage as the rotor speed attempts to accelerate. It usually takes around 10-15 seconds to estimate motor resistance, inductance and Motor BEMF constant Ke.

3. *Check if motor spins in closed loop at commanded speed.*

    In closed loop, the motor should spin at the commanded speed and there should not be any faults triggered.

    a.   Enable the "Auto read fault status" toggle button towards the top right corner of the GUI. Then monitor the Fault Status in the side panel to the right.

    If no fault gets triggered, skip Section 3.6 and proceed to Section 4.

4. Prepare device for fault handling if any fault gets triggered.

    If the motor failed to spin successfully in closed loop, check the fault status.

    a.   Set zero speed command by turning off the PWM, turning the potentiometer completely counterclockwise, or setting the I2C command percentage to 0%.

    b.   Clear the fault status registers by clicking on the "Clear" button in the Fault Status side panel to the right as shown in Figure 3-1

    Check Section 3.6 for steps to debug faults.

This completes the essential controls section. At the end of this section, user should be able to spin the motor in closed loop. Save the register settings by clicking File->Save Registers in the GUI. If there were no faults, skip to Section 4.
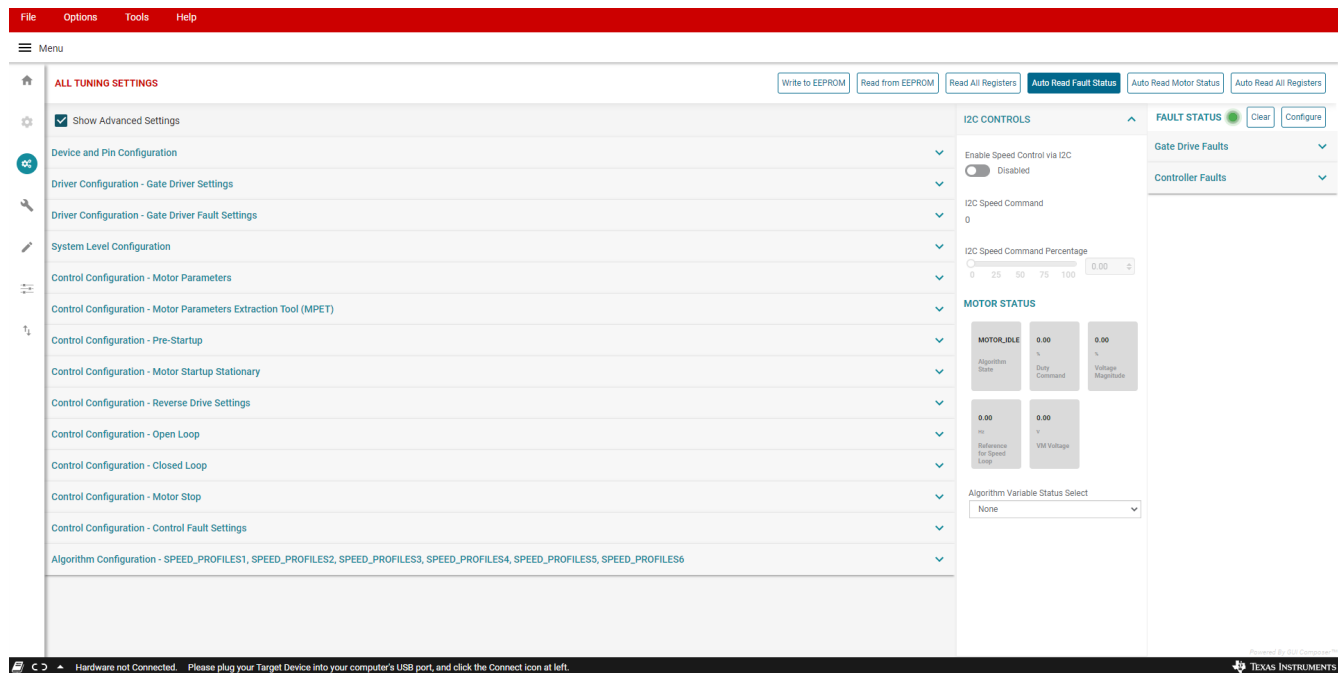

Figure 3-1. Fault Status

3.6 Fault Handling

Below are the faults that can possibly get triggered based on the default register configuration specified in Table 3-1.

3.6.1 MPET IPD Fault [MPET_IPD_Fault]

This fault gets triggered when there is no motor connected to the phase pins of the device or when the MPET IPD Current threshold [MPET_IPD_CURRENT_LIMIT] is set low for motors with high resistance. If this fault is triggered, then please follow the below suggestions.

Step 1: After confirming this fault triggers consistently from multiple start up attempts, check if all the phases of the motor are connected to the phase pins of the device.

Step 2: If the fault persists, increase the MPET IPD Current threshold [MPET_IPD_CURRENT_LIMIT]

Step 3: If the fault still persists, check motor datasheet for motor resistance and inductance. Open the "Control configuration – Motor Parameters" tab in All tuning settings section in the GUI and program motor phase resistance and motor phase inductance.

Step 4: If the motor does not have a datasheet, then measure the phase to phase resistance across any two phases using a Digital Multimeter and calculate the phase resistance by dividing the phase to phase resistance by 2 as shown in equation Equation 2. The motor phase resistance refers to the equivalent phase to center tap resistance, R_{PH} as shown in Figure 3-2. This measurement holds good for both star wound and delta wound motors.

$$R_{PH}\left(Phase\ resistance\right) = \frac{R_{PH\_PH}\left(Phase\ to\ Phase\ resistance\right)}{2} \tag{2}$$
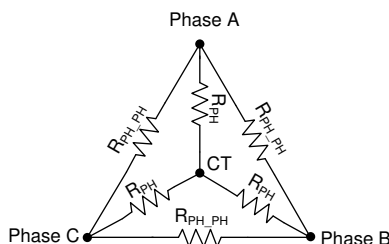

Figure 3-2. Motor Resistance

To know the motor inductance, measure the phase to phase inductance at 1 kHz across any two phases using an LCR meter. Calculate the phase inductance by dividing the phase to phase inductance by 2. Motor phase inductance refers to the inductance from the phase output to the center tap, L_{PH} as shown in Figure 3-3. For motors with different phase-phase inductances, measure all three phase to phase inductances and calculate their average and use this value as the phase to phase inductance. This measurement holds good for both star wound and delta wound motors.
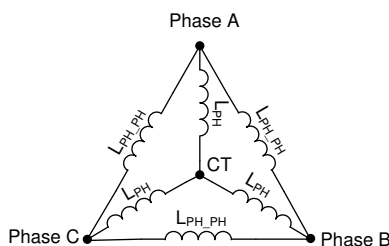

Figure 3-3. Motor Inductance

3.6.2 MPET BEMF Fault [MPET_BEMF_Fault]

This fault gets triggered when motor stalls while running in open loop for BEMF estimation. If this fault is triggered, then please follow the below suggestions.

Step 1: Increase MPET Open loop current reference [MPET_OPEN_LOOP_CURR_REF]

Step 2: Decrease MPET open loop slew rate [MPET_OPEN_LOOP_SLEW_RATE]

Step 3: If this fault persists, check motor datasheet for motor BEMF constant in mV/Hz and program Motor BEMF Constant [MOTOR_BEMF_CONST] in the "Control configuration – Motor Parameters" tab in All tuning settings section in the GUI.

Equation 3 and Equation 4 can be used to convert Ke in mV/rpm and torque constant to Ke in mV/Hz.

$$BEMF\ Constant\ in\ \left(\frac{mV}{Hz}\right) = BEMF\ Constant\left(\frac{mV}{rpm}\right) \times \frac{60}{Pole\ pairs} \tag{3}$$

$$BEMF\ constant\ \left(\frac{mV}{Hz}\right) = Torque\ constant\left(\frac{mNm}{A}\right) \times \frac{2\pi}{Pole\ pairs} \tag{4}$$

Step 4: If the motor does not have a datasheet, then measure the voltage across any two phases of the motor using an oscilloscope by manually spinning the motor. A sinusoidal or trapezoidal voltage should appear on the oscilloscope. Measure the peak voltage E_{PH} in milli-volts and time period T_E in seconds as shown in Figure 3-4. Calculate BEMF constant Ke as shown in Equation 5.
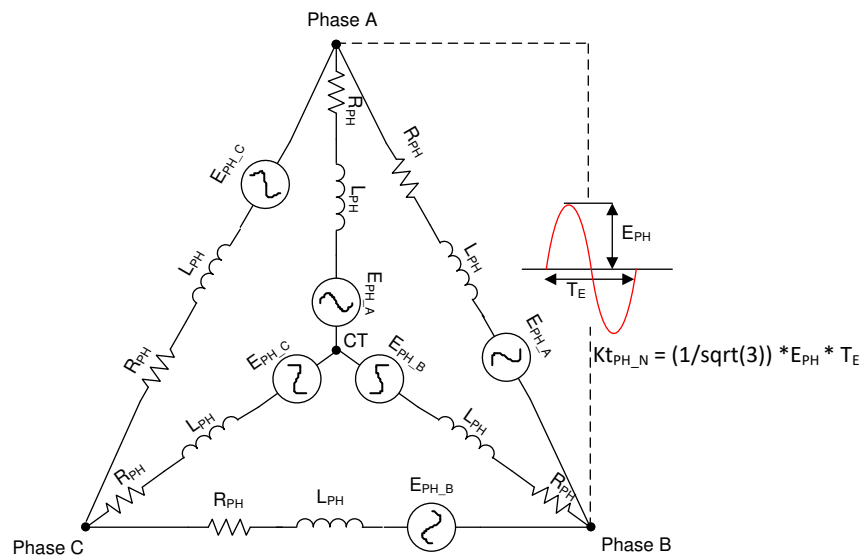

Figure 3-4. Motor BEMF Constant

$$BEMF\ Constant\ Ke = \frac{E_{PH} \times T_E}{\sqrt{3}} \tag{5}$$

3.6.3 Abnormal BEMF Fault [ABN_BEMF]

This fault gets triggered when the estimated BEMF voltage drops below the programmed Abnormal BEMF threshold % [ABNNORMAL_BEMF_THR] of the programmed or MPET estimated BEMF constant Ke [MOTOR_BEMF_CONST] which is programmed as 40% (default value). For example, if the estimated / measured Ke is 5 mV/Hz and programmed Abnormal BEMF threshold is 40%, this fault gets triggered when the estimated Ke drops below 2 mV/Hz. This fault can also get triggered when the programmed Ke is inaccurate.

Step 1: Estimated BEMF voltage drops when the motor speed drops. Motor speed can drop due to load dynamics (sudden change in load). For applications with load dynamics, we expect speed to drop and recover back. Because the speed drops, the BEMF voltage will also drop and can trigger this fault. For such applications, it is recommended to set the Abnormal BEMF threshold to 10% to avoid triggering this fault.

Step 2: This fault can get triggered if the programmed Ke is inaccurate. Follow steps recommended in Section 3.6.2 to obtain accurate Ke.

3.6.4 Lock Current Limit [LOCK_LIMIT]

This fault gets triggered when the motor is loaded above the stall torque of the motor.

We recommend to check the motor datasheet for stall torque and load the motor below the stall torque specified in the datasheet. If the load torque is still within the stall torque, we recommend to increase the Lock_ILIMIT in the Control Configuration – Control Fault settings in the All tuning Settings page.

3.6.5 Hardware lock Current Limit [HW_LOCK_LIMIT]

This fault gets triggered when there is a spike in the phase current due to short circuit as shown in Figure 3-5. Short circuit can be due to phase to phase, phase to GND or Vm to GND short. Another reason to trigger this fault could be incorrect speed loop and current loop PI controller gains.
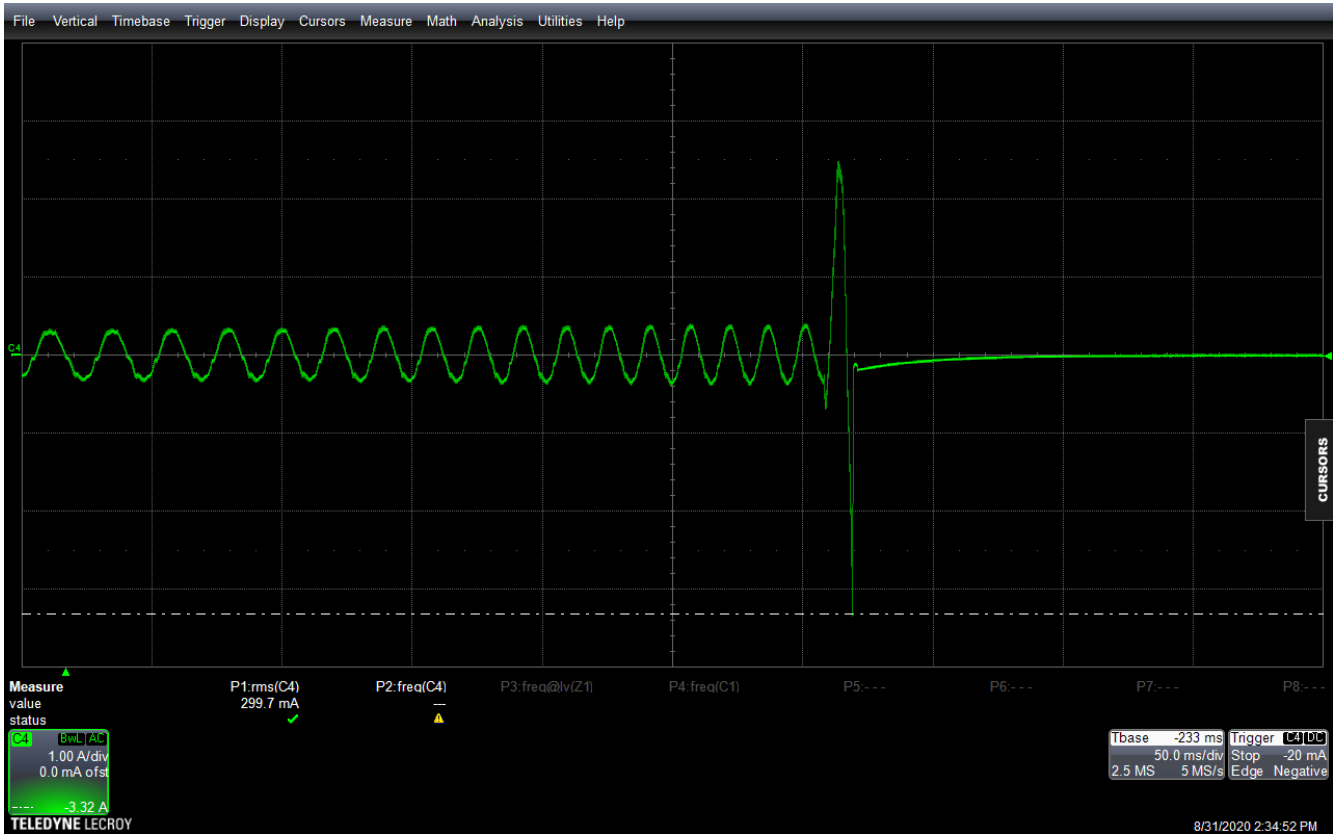

Figure 3-5. Phase Current Waveform Showing a Spike due to Short Circuit

Step 1: Program Speed loop Kp [SPD_LOOP_KP], Speed loop Ki [SPD_LOOP_KI], current loop Kp [CURR_LOOP_KP] and current loop Ki [CURR_LOOP_KI] to zero. This enables auto calculation of speed loop and current loop PI controller gains.

Step 2: Increase the MCC Dead time [MCC_DEAD_TIME] and retry again.

Step 3: If the fault still persists, check the continuity across phase to phase, phase to GND or Vm to GND to make sure there is no short across these terminals.

3.6.6 No Motor Fault [NO_MTR]

This fault gets triggered when the phase current is below No motor lock threshold % of base current.

Step 1: Make sure the motor phases are connected to J5 or TP9, TP10, or TP11.

Step 2: If the fault persists, set the No-Motor lock current threshold [NO_MTR_THR] to 0.05A.

4 Basic Controls

The goal of this section is to provide customers tuning guidance to conform to use-case and explore various features in the device.

Note, the user is expected to skip the subsection use-cases and scenarios that do not apply to the system or end equipment.

4.1 Device and Pin Configuration

4.1.1 Power Saver or Sleep Mode for Battery Operated Applications

MCF8316A can be configured to be in Sleep mode or Standby mode. For applications that operate using batteries, it is important to save power while the device is not spinning the motor.

MCF8316A enters sleep mode when speed input is held continuously below the speed threshold for SLEEP_ENTRY_TIME.

In sleep mode, all motor outputs are disabled, the charge pump is disabled, the AVDD regulator is disabled, and the serial communication bus is disabled. MCF8316A can be configured to be in sleep mode by configuring register bits DEV_MODE. Please note that in Sleep mode, GUI will loose connectivity as there will be no I2C communication.

4.1.2 Direction and Brake Pin Override

Direction and Brake pin override will be helpful in applications where it is not intended for the users to affect the direction and brake pins in the device. In such cases, it is recommended to override both pins and configure them via EEPROM.

Direction pin in MCF8316A can be overwritten by configuring DIR_INPUT. Configuring DIR_INPUT to 01b will overwrite Hardware pin to clockwise rotation if motor phases A, B and C are connected to OUTA, OUTB and OUTC respectively. Commutation sequence in clockwise direction will be OUTA-OUTB-OUTC. Configuring DIR_INPUT to 10b will overwrite Hardware pin to anti-clockwise rotation if motor phases A, B and C are connected to OUTA, OUTB and OUTC respectively. Commutation sequence in anti-clockwise direction will be OUTA-OUTC-OUTB.

BRAKE pin in MCF8316A can be overwritten by configuring BRAKE_INPUT. Configuring BRAKE_INPUT to 01b will overwrite Hardware pin to low side brake/Align according to BRAKE_PIN_MODE. Configuring BRAKE_INPUT to 10b will overwrite Hardware pin and not low side brake/align.

In Low side braking mode, all the LS FETs will be turned ON. In Align braking mode, MCF8316A aligns a motor by injecting dc current through a particular phase pattern.

4.2 System Level Configuration

4.2.1 Tracking Motor Speed Feedback in Real Time

MCF8316A device provides information about the motor speed through the frequency generator (FG) pin, which is also known as a TACH (tachometer) out. In MCF8316A, the FG pin behavior is configured through FG_CONFIG. Configure FG_SEL to output FG signal only in closed loop, both open loop and closed loop, or only in open loop for the first try. Configure FG_DIV to number of motor poles so that the FG output matches the motor mechanical speed in Hz.

When FG_CONFIG is configured to 0 (FG active as long as motor is driven), the FG output is active as long as MCF8316A is driving the motor. FG will not be active during a motor stop and coasting condition. In this mode, FG is released high when MCF8316A enters sleep or standby mode. This mode is useful in applications that require real time motor speed information as long as MCF is driving the motor.

When FG_CONFIG is configured to 1 (FG active till BEMF drops below FG_BEMF_THR), MCF8316A provides FG output until BEMF falls below FG_BEMF_THR. The FG output will continue to indicate motor speed even if the motor is not being actively driven. This mode is useful in applications that require motor speed information above a certain speed, and rotor motion information during the coast and braking conditions.

For example, if the motor Ke is 5 mV/Hz and the application require measuring motor speed above 4 Hz, then the user can configure FG_BEMF_THR to 20 mV. Once when the motor speed reaches 4 Hz, the device will output FG as the BEMF voltage will be 20 mV at 4 Hz.

4.2.2 Improving Acoustic Performance

Acoustics is audible noise contributed by motor commutation and harmonic frequencies in motor drives. Stator excitation in motor drivers can generate mechanical resonance at audible frequency ranges leading to audible noise. Improving acoustic performance is important for applications that require quiet operation. Any distortion in phase current can translate to audible noise. Phase current distortion can be due to dead time, PWM modulation scheme or low MOSFET slew rate.

Step 1: Configure PWM Modulation scheme [PWM_MOD] to "Continuous Space Vector Modulation".

Note

1. In Continuous Space vector modulation, phase current waveform shaping will be better but the MOSFET switching losses will be higher.
2. In Discontinuous Space vector modulation, MOSFET switching losses will be lower. However, we can expect phase current distortion for low inductance motors.

Step 2: Enable Dead time compensation [DEADTIME_COMP_EN]

For speeds above 125 Hz, decrease MCC dead time [MCC_DEAD_TIME] till the phase current distortion gets better.

Note

Decreasing the dead time might trigger Hardware lock detection current limit fault [HW_LOCK_ILIMIT]. If this fault gets triggered, increase the dead time.

Figure 4-1 shows the phase current waveform when dead time compensation is disabled. Fundamental frequency of phase current is 40 Hz. Fast Fourier transform (FFT) of phase current plot shows harmonics at 160 Hz and 220 Hz. Figure 4-2 shows the phase current waveform when dead time compensation is enabled. Phase current looks more sinusoidal and the FFT of phase current plot does not have any harmonics.
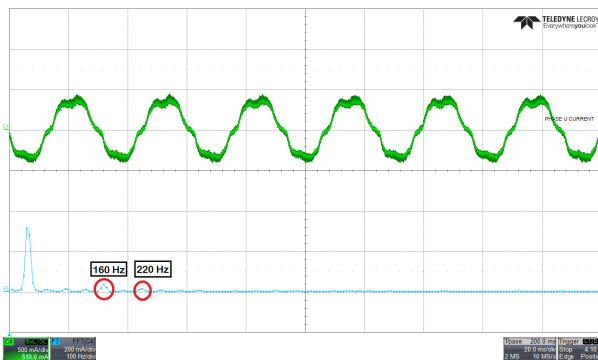

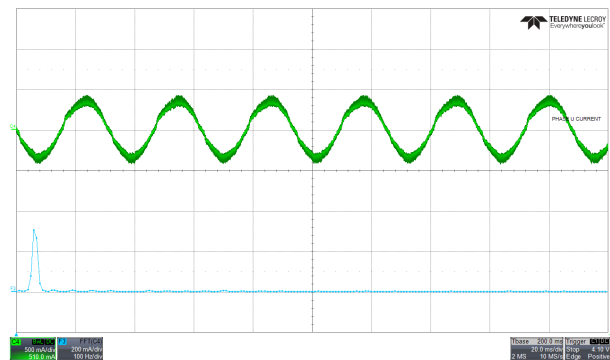**Figure 4-1. Phase current and FFT - Dead time compensation disabled**

Figure 4-2. Phase current and FFT - Dead time compensation enabled

4.2.3 Protecting the Power supply

Protecting the power supply from drawing higher current or potential voltage overshoots is important in battery operated applications or applications that do not have an internal overcurrent or overvoltage protection built into the power supply.

Step 1: When the load on the motor increases, device draws higher current from the power supply. For applications that may want to limit the current drawn from the power supply, it is recommended to Enable Bus current limit [BUS_CURRENT_LIMIT_ENABLE] and configure the Bus current limit [BUS_CURRENT_LIMIT] to protect the power supply from drawing higher current.

For example, it is required to limit the current drawn from power supplies such as batteries as the battery life depends on the charge or discharge cycles. Enabling bus current limit limits the power supply current by limiting the speed of the motor.

Step 2: When a command is issued for the motor to decelerate, based on the deceleration rate, the energy from motor may be pumped back to the power supply, increasing the supply voltage to possibly unsafe levels for electronics. Enable the Antivoltage surge [AVS] to protect the power supply from voltage overshoots which will override any deceleration limit set by any other register and automatically apply a safe deceleration rate.

Figure 4-3 shows overshoot in power supply voltage when AVS is disabled. Motor decelerates from 100% duty cycle to 10% duty cycle at a deceleration rate of 70,000 Hz/sec. Figure 4-4 shows no overshoot in power supply voltage when AVS is enabled.
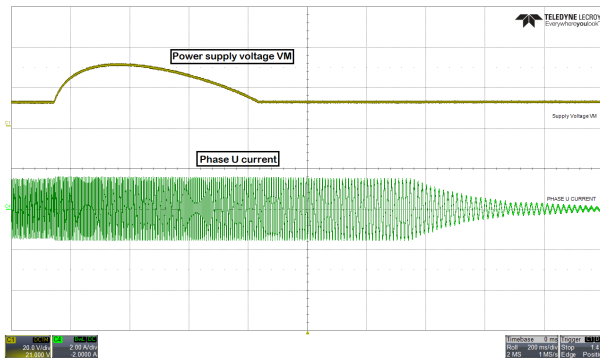

Figure 4-3. Power Supply voltage and phase current waveform when AVS is disabled
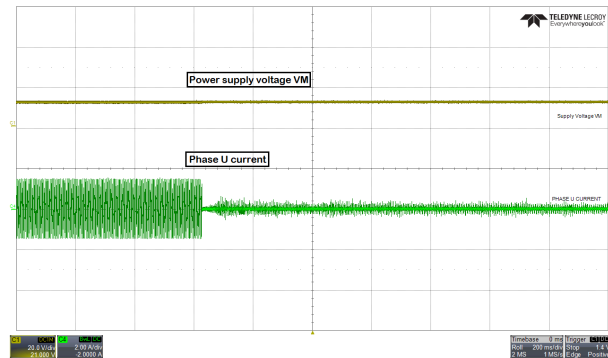

Figure 4-4. Power Supply voltage and phase current waveform when AVS is enabled

4.2.4 Monitoring Power Supply Voltage Fluctuations for Normal Motor Operation

In applications where the power supply may fluctuate, it is recommended to specify the minimum and maximum power supply voltage range. During an undervoltage condition, the motor might operate in overmodulation region to achieve the target speed leading to current distortion, inefficiency or noise. During an overvoltage condition, the MOSFETs, MCF8316A and motor may be stressed with continued operation in high voltage.

Step 1: Keep decreasing the supply voltage till we see a drop in the speed. Measure the bus voltage at which the speed drops and set MIN_VM_MOTOR to that value. Range of minimum bus voltage that can be configured is between 4.5 V and 12.5 V.

Step 2: Keep increasing the bus voltage to a point where the motor phase voltage reaches the maximum rated voltage of the motor. MAX_VM_MOTOR will be the bus voltage at which motor phase voltage reaches the maximum rated voltage of the motor. Range of maximum bus voltage that can be configured is between 20 V and 35 V.

Note

MCF8316A provides an undervoltage recovery mode [MIN_VM_MODE] and an overvoltage recovery mode [MAX_VM_MODE]. Undervoltage recovery mode can be configured to either automatically clear Undervoltage fault [MTR_UNDER_VOLTAGE] or latch on Undervoltage fault. Overvoltage recovery mode can be configured to either automatically clear Overvoltage fault [MTR_OVER_VOLTAGE] or latch on Overvoltage fault.

4.3 Control Configurations

4.3.1 Motor Parameter Estimation to Minimize Motor Parameter Variation Effects

The MCF8316A uses motor resistance, motor inductance and the motor Back-EMF constant to estimate motor position in closed loop operation. The MCF8316A has capability of automatically measuring motor parameters in offline state, rather than having the user enter the values themselves. Offline measurement of parameters takes place before normal motor operation. Variation in motor parameters can impact the accuracy of motor

position estimation and therefore the motor commutation and rotation. It is recommended to estimate the motor parameters before motor startup to minimize the impact caused due to possible parameter variations.

The MCF8316A has Motor parameter extraction tool (MPET) which estimates the motor phase resistance, the motor phase inductance and the motor Back-EMF constant. Go to "Control Configuration - Motor parameter extraction tool (MPET)".

Step 1: Enable Measure motor resistance [MPET_R], Measure motor inductance [MPET_L] and Motor BEMF constant [MPET_Ke].

Step 2: Enable MPET Start command [MPET_CMD]. Make sure the motor is stationary before enabling MPET start command.

Step 3: After successful motor parameter extraction, copy the estimated parameters to the respective shadow registers by enabling MPET write to shadow register flag.

Note

1. If the device triggers MPED_IPD_FAULT, increase the MPET IPD Current threshold till the rated current of the motor.
2. If the device triggers MPED_IPD_FAULT, decrease MPET Open loop slew rate [MPET_OPEN_LOOP_SLEW_RATE].
3. If the fault still persists, increase MPET Open loop current reference [MPET_OPEN_LOOP_CURRENT_REF] and decrease MPET Open loop speed reference [MPET_OPEN_LOOP_SPEED_REF]

Figure 4-5 shows the phase current waveform during motor parameter measurement. Figure 4-6 shows the IPD current waveform during R, L and Ke measurement. Bottom half of Figure 4-6 shows the IPD current waveform during R and L measurement. R is measured during the rising of phase current and L is measured during the falling of phase current. After R and L measurement, motor spins in open loop. Once the speed reaches MPET open loop speed reference [MPET_OPEN_LOOP_SPEED_REF], motor is coasted. BEMF voltage of all three phases are measured and Ke is calculated.
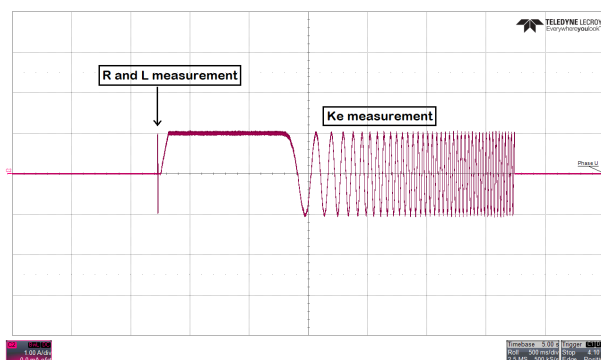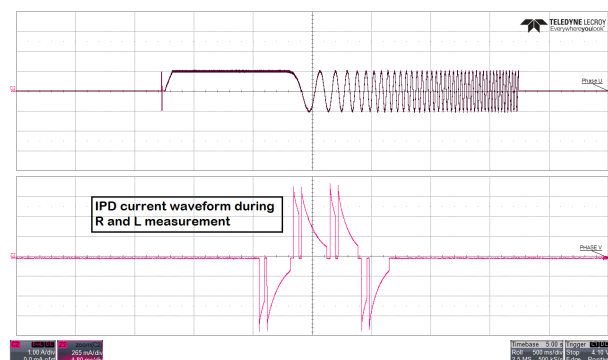

Figure 4-5. MPET - Phase current

Figure 4-6. IPD current waveform during R and L measurement

4.3.2 Initial Speed Detection of the Motor for Reliable Motor Resynchronization

The Initial Speed Detection (ISD) function is used to identify the initial condition of the motor. It is important to know the initial condition of the motor for reliable resynchronization. Motor resynchronization failures can occur when the device attempts to start the motor while the motor is coasting or spinning in the direction opposite to the intended direction of spin. Motors can coast in applications that require frequent motor starts and stops, or if the motor Is being forced externally or if there is a power interruption. Motors can spin in the direction opposite to the intended direction of spin if motor phase wires are connected to OUTA, OUTB and OUTC in wrong sequence or when wrong direction command is issued. Motors with higher inertia coast for a longer period of time. It is recommended to have ISD enabled in applications that require frequent motor starts and stops and use higher inertia motors.

For example, ceiling fan motors have higher inertia due to the fan blades and can coast for long time before stopping.

Step 1: Enable ISD [ISD_EN]

Step 2: Enable Motor ISD Resynchronize [RESYNC_EN]

Note

Follow the below recommendations if the motor fails to startup.

1. Increase the Motor Stationary BEMF Threshold [STAT_DETECT_THR].
2. Increase the Motor Stationary Persistence Time [ISD_STOP_TIME].
3. Increase the Motor Run Persistence Time [ISD_RUN_TIME].
4. Increase the Minimum speed threshold to resynchronize to closed loop

---

4.3.3 Unidirectional Motor Drive Detecting Backward Spin

For applications that require spinning the motor in a specific direction, it is important to know if the motor is coasting or spinning in the direction opposite to the intended direction of spin. MCF8316A reverse drive function acts to reverse decelerate the motor through zero speed and to accelerate after changing direction until it transitions into closed loop as shown in Figure 4-7.
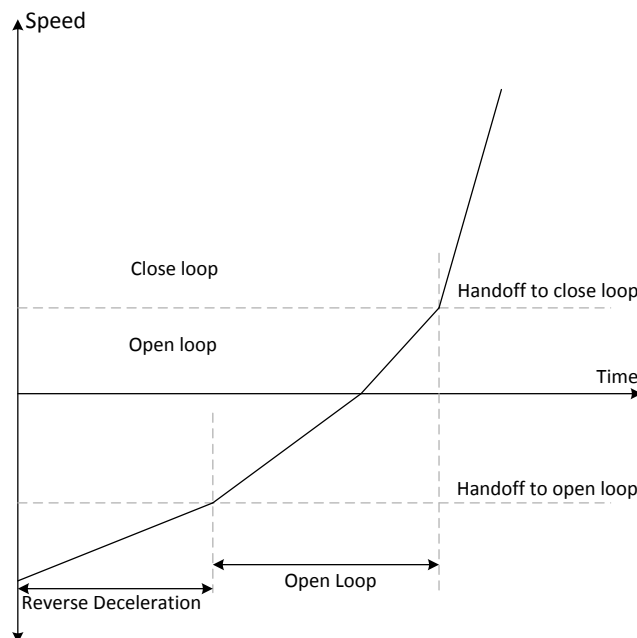

Figure 4-7. Reverse Drive Function

MCF8316A provides an option to apply brakes and stop the motor while the motor is coasting or spinning in reverse direction and then accelerate into closed loop after changing the direction.

In applications such as ceiling fans and pumps, it is required to spin the motor in specific direction for desired results. For such applications, it is recommended to follow the below recommendations.

Step 1: Enable ISD [ISD_EN]

Step 2: Enable Motor ISD Reverse drive [RVS_DR_EN]

Step 3: Enable reverse resynchronization [RESYNC_EN]

Step 4: Enable Active brake reverse drive [ACTIVE_BRAKE_REV_DRV_EN]

Step 5: Enter the Kp coefficient [ACTIVE_BRAKE_KP] and Ki coefficient [ACTIVE_BRAKE_KI] for Active braking loop. We recommend to start with Kp coefficient [ACTIVE_BRAKE_KP] as 700 and Ki coefficient [ACTIVE_BRAKE_KI] as 100.

Step 6: Set bus current limit during active braking [ACTIVE_BRAKE_CURRENT_LIMIT]

Note

Follow the below recommendations if the motor fails to resynchronize in reverse direction.

1. Increase the reverse deceleration speed threshold to transition to open loop
2. Enable Open loop reverse drive configuration [REV_DRV_CONFIG]
3. Increase the Reverse Drive Open Loop Current Reference [REV_DRV_OPEN_LOOP_CURRENT]
4. Decrease open loop acceleration coefficient A1 and A2 during reverse drive
5. If the device triggers Lock current limit [LOCK_LIMIT], then increase the open loop current limit during speed reversal [REV_DRV_OPEN_LOOP_CURRENT]

4.3.4 Preventing Back Spin of Rotor During Startup

For applications where a reverse spin is not acceptable, the Initial Position Detection algorithm (IPD) function is an alternative way to start up the motor. With the proper IPD setting, the motor startup can be faster than using align. While this function is suitable for motors with high inertia, such as heavy blades (for example: a ceiling or appliance fan), it is not suitable for motors with low inertia, such as small blades (for example: a computer fan), because the current injection will cause the motor to shake, resulting in the IPD not being accurate.

In applications where the acoustic noise ("chirp") generated by IPD is not acceptable during startup, it is recommended to select Slow first cycle as the startup method.

Option 1: IPD

Step 1: If IPD is chosen as startup method, select IPD in the Motor startup option [MTR_STARTUP] in "Control Configuration – Motor Startup Stationary" tab in the GUI.

Step 2: Select the IPD Current threshold [IPD_CURR_THR]. IPD current threshold is selected based on the inductance saturation point of the motor. A higher current has better chance to accurately detect the initial position. However, higher current might result in rotor movement, vibration and noise. It is recommended to start with 50% of the rated current of the motor. If the motor startup is unsuccessful, then we recommend increasing the threshold till the motor starts up successfully. Note that the IPD current threshold should not be higher than the rated current of the motor.

Step 3: Select IPD clock value [IPD_CLK_FREQ]. IPD clock defines how fast the IPD pulses are applied. Higher inductance motors and higher current thresholds need a longer time to settle the current down, so we need set the clock at a slower time. However, a slower clock makes the IPD noise louder and it lasts longer, so we suggest setting the clock as fast as possible as long as IPD current is able to settle down completely.

Note

Device triggers IPD timeout faults [IPD_T1_FAULT] and [IPD_T2_FAULT] for motors with very high inductance, or if the motor is not connected. If this fault gets triggered, it is recommended to check if motor is connected to the device. If the fault still persists, it is recommended to set the IPD release mode [IPD_RLS_MODE] to Tri-state if any overshoot in DC bus voltage is acceptable.

Device triggers IPD Frequency fault [IPD_FREQ_FAULT] if the IPD clock frequency is set too high. If this fault gets triggered, it is recommended to decrease the IPD Clock value [IPD_CLK_FREQ].

Step 4: Select IPD Advance Angle [IPD_ADV_ANGLE]. It is recommended to Start with 90º to get maximum startup torque. If there is sudden jerk observed during startup, then it is recommended to reduce the angle to 60º or 30º for a smoother startup.

Option 2: Slow first cycle

Follow below steps if Slow first cycle is chosen as the startup method.

Step 1: Select Slow first cycle in the Motor startup option [MTR_STARTUP] in "Control Configuration – Motor Startup Stationary" tab in the GUI.

Step 2: Select Align or Slow first cycle current reference [ALIGN_OR_SLOW_CURRENT_ILIMIT]. Lower current reference may lose synchronization of motor. Higher current may lead to sustained oscillations for high inertia motors, or sudden jerky motion for low inertia motors. It is recommended to start with 50% of the rated current of the motor. In applications where the startup torque is high, the motor might lose synchronization. In such applications, it is recommended to increase the current reference. In applications where, sustained oscillations or sudden jerks are observed, it is recommended to decrease the current reference.

Step 3: Select Align or slow first cycle current ramp rate [ALIGN_SLOW_RAMP_RATE]. Current reference is ramped to avoid reverse rotation of the motor. Lower current ramp rate may lose synchronization of motor. A higher current ramp rate may lead to sustained oscillations for high inertia motors, or sudden jerking motion for low inertia motors. It is recommended to start with setting up the ramp time to 0.5 sec to ramp to rated current of the motor. In applications where the startup torque is high, the motor might lose synchronization. In such applications, it is recommended to increase the current ramp rate. In applications where, sustained oscillations or sudden jerks are observed, it is recommended to decrease the current ramp rate.

Step 4: Select Frequency of first cycle [SLOW_FIRST_CYC_FREQ]. Lower frequency may give a jerky motion at startup. Higher frequency may not be able to synchronize the motor. It is recommended to start with 20% of the maximum speed of the motor. In applications where the startup torque is high, the motor might lose synchronization. In such applications, it is recommended to decrease the frequency. In applications where jerky motions are observed, it is recommended to increase the frequency.

4.3.5 Faster Startup Timing

Startup time is the time taken for the motor to reach the target speed from zero speed. For applications that require quick startup time, we recommend choosing either Initial Position Detection (IPD) or Slow first cycle as the startup method.

Option 1: Initial Position Detection (IPD)

Step 1: Select IPD [MTR_STARTUP] as the motor startup method.

Step 2: Increase IPD current threshold [IPD_CURR_THR] to rated current of the motor.

Step 3: Increase IPD clock value [IPD_CLK_FREQ] to higher frequency up to a value where the device does not trigger IPD frequency fault. Check Section 4.3.4 (Step 3) for more details.

Step 4: Select IPD repeating times [IPD_REPEAT] to 1 time.

Step 5: Select Open loop current limit [OL_ILIMIT] to be the same as Current limit for Torque PI Loop [ILIMIT].

If the device triggers Lock current limit [LOCK_LIMIT], it is recommended to increase [LOCK_ILIMIT] upto the stall current of the motor. Configuring this to a value higher than motor stall current will overheat or damage the motor.

Step 6: Increase Open loop acceleration coefficient A1 [OL_ACC_A1] and Open loop acceleration coefficient A2 [OL_ACC_A2].

Note

A1 and A2 can be increased until open loop current reaches Lock detection current threshold [LOCK_ILIMIT]. Open loop current can be measured using oscilloscope. Increasing Open loop acceleration coefficient A1 [OL_ACC_A1] and Open loop acceleration coefficient A2 [OL_ACC_A2] might trigger LOCK_LIMIT. If this happens, reduce A1 and A2 until LOCK_LIMIT no longer triggers.

Step 7: Select Minimum BEMF for handoff [AUTO_HANDOFF_MIN_BEMF] to 0 mV.

If the device triggers Abnormal BEMF [ABN_BEMF] fault, then it is recommended to increase the [AUTO_HANDOFF_MIN_BEMF].

Step 8: Keep increasing ramp rate for reducing difference between estimated theta and open loop theta to 2 deg/ms.

Step 9: Increase Closed loop acceleration rate [CL_ACC]

Note

LOCK_LIMIT fault handling:

Closed loop acceleration rate [CL_ACC] can be increased until closed loop current reaches Lock detection current threshold [LOCK_ILIMIT]. Closed loop current can be measured using oscilloscope. Increasing closed loop acceleration rate [CL_ACC] might trigger LOCK_LIMIT. If this happens, reduce closed loop acceleration rate [CL_ACC] until no longer triggers.

Option 2: Slow first cycle

Step 1: Select Slow first cycle as the motor startup method in [MTR_STARTUP].

Step 2: Select Align or slow first cycle current limit [ALIGN_OR_SLOW_CURRENT_ILIMIT] to be the same as Current limit for Torque PI loop [ILIMIT].

Step 3: Keep increasing Align or slow first cycle current ramp rate [ALIGN_SLOW_RAMP_RATE] until the open loop current reaches 100% of the rated current of the motor.

Step 4: Follow Step 5 to Step 9 in Option 1.

4.3.6 Gradual and Smooth Start up Motion

For applications that require slow and gradual startup with lower speed overshoots during handoff, follow the below recommendations.

Step 1: Decrease Open loop acceleration coefficient A1 [OL_ACC_A1] and Open loop acceleration coefficient A2 [OL_ACC_A2].

Step 2: Enable Iq ramp down after transition to closed loop [IQ_RAMP_EN]

If there is speed overshoots, it is recommended to decrease ramp rate for reducing difference between estimated theta and open loop theta [THETA_ERROR_RAMP_RATE].

4.3.7 Improving Speed Regulation

For applications that require better speed regulation, it is recommended to tune Speed loop PI controllers [SPD_LOOP_KP] and [SPD_LOOP_KI]. Kp coefficient of speed loop [SPD_LOOP_KP] controls the settling time and speed overshoots. Ki coefficient of Speed loop [SPD_LOOP_KI] controls speed overshoot and ensures regulation of speed at set value and drives the error to zero. Speed loop PI controller gains can either be auto-tuned by MCF8316A or tuned manually.

Auto Tuning: MCF8316A auto calculates Speed loop PI controller gains by setting [SPD_LOOP_KP] and [SPD_LOOP_KI] to zero.

Manual Tuning: Follow below steps to tune Speed loop PI controller gains manually

Step 1: Disable Speed loop [SPEED_LOOP_DIS]

Step 2: Disable closed loop [CLOSED_LOOP_DIS]

Step 3: Issue non-zero speed command to start the motor (refer to Section 3.5, step 1 on how to issue non-zero speed command). Motor will spin in open loop.

Step 4: Allow the open loop current to settle down and then measure the peak open loop current.

Step 5: Enter 110% of measured peak phase current as the Iq reference when speed loop is disabled [FORCE_IQ_REF_SPEED_LOOP_DIS]

Step 6: Enable closed loop [CLOSED_LOOP_DIS]. Now the motor will enter closed loop in torque mode.

Stp 7: Adjust the Iq reference [FORCE_IQ_REF_SPEED_LOOP_DIS] such that the motor speed reaches the maximum speed.

Step 8: Speed loop Kp [SPD_LOOP_KP] is calculated using Equation 6.

$$Speed\ loop\ Kp\ =\ \frac{Iq\ reference\ at\ maximum\ speed}{Maximum\ Electrical\ Speed\ in\ Hz} \tag{6}$$

Step 9: Speed loop Ki [SPD_LOOP_KI] is calculated using equationEquation 7.

$$Speed\ loop\ Ki = 0.1\ \times Speed\ loop\ Kp \tag{7}$$

Step 10: Enable speed loop [SPEED_LOOP_DIS]

Note

Tuning speed loop Kp and Ki is experimental. If the above recommendation doesn't work, then we recommend to manually tune Speed loop Kp and Ki till the desired results are achieved.

4.3.8 Stopping Motor Quickly

For applications that require stopping the motor quickly, it is recommended to configure Motor stop options [MTR_STOP] to either High side braking or Low side braking.

Step 1: Configure Motor stop options [MTR_STOP] to either High side braking or Low side braking.

Step 2: Select Speed threshold for BRAKE pin and Motor STOP options. Setting speed threshold to higher speed would result in FETs carrying large current. Setting speed threshold to lower speed would increase the stop time of the motor. It is recommended to start with 50% of the maximum speed. If the device triggers Overtemperature warning [OTW] or Overtemperature shutdown [OTS], it is recommended to decrease the threshold. If the stop time is too high, it is recommended to increase the threshold.

4.3.9 Preventing Supply Voltage Overshoot During Motor Stop.

For applications that require preventing supply voltage overshoots during motor stop, it is recommended to select active spin down or recirculation brake as Motor stop options. Recirculation brake can be used as a motor stop option in applications where fast stop is not required, but any inductive energy going back to power supply is not acceptable. Active spin down can be used as a motor stop option in applications where fast stop is not required but some amount of inductive energy going back to power supply is acceptable

Option 1: Recirculation Mode

Step 1: Configure Motor stop options [MTR_STOP] to Recirculation Mode.

Step 2: Configure Recirculation Brake Time [RECIR_BRK_TIME].. If there is voltage overshoot on Vm during motor stop, configure RECIR_BRK_TIME long enough to prevent overshoot on Vm pin.

Option 2: Active spin down

Step 1: Configure Motor stop options [MTR_STOP] to Active spin down

Step 2: Configure active spin down speed threshold [ACT_SPIN_THR]. It is recommended to set the ACT_SPIN_THR to 50% of the maximum speed. If there is voltage overshoot seen on the power supply, it is recommended to decrease the ACT_SPIN_THR till the voltage overshoot reaches acceptable limit.

4.3.10 Protecting Against Rotor Lock or Stall Condition

Follow below recommendations if the motor stalls or gets locked in closed loop when the motor is overloaded.

Increase lock detection current threshold [LOCK_ILIMIT]

Note

MCF8316A provides options to either latch LOCK_LIMIT fault or auto retry for [AUTO_RETRY_TIMES] after every [LCK_RETRY] seconds. This can be configured in LOCK_ILIMIT_MODE. We have defaulted the auto retry time and lock retry.

4.3.11 Maximizing Thermal Efficiency and Increasing Thermal Performance

Thermal performance can be improved by minimizing power dissipation. Power dissipation across $R_{DS(on)}$, Switching losses and operating supply current dissipation are the major sources of Power dissipation in MCF8316A. Power dissipation across $R_{DS(on)}$ is fixed as MOSFETs are integrated in MCF8316A. Please follow below recommendations to minimize switching losses and operating supply current dissipation.

Step 1: Increase Slew rate [SLEW_RATE]

Please note that increasing Slew rate will increase EMI noise.

Step 2: Decrease PWM output frequency [PWM_FREQ_OUT]

Please note that decreasing PWM output frequency may lead to discontinuous phase current for very low inductance motors.

4.3.12 Mitigating Electromagnetic Interference (EMI)

Please follow the below recommendations to mitigate EMI in your system.

Step 1: Decrease Slew rate [SLEW_RATE]

Please note that decreasing Slew rate will increase power dissipation due to switching losses.

Step 2: Enable Spread spectrum Modulation [SSM_DIS].

4.3.13 Faster deceleration

Decelerating the motor quickly requires motor mechanical energy to be extracted and disposed - input DC voltage increases if this energy is returned to the DC input supply. When active braking is enabled, energy taken from DC power supply is used to brake the motor - this prevents DC voltage spike during fast deceleration. The mechanical energy of the motor and energy taken from DC source, both are dissipated within the motor itself.

Follow below steps to decelerate the motor quickly.

Step 1: Enable active braking [ACTIVE_BRAKE_EN].

Step 2: Configure ACTIVE_BRAKE_CURRENT_LIMIT. This sets the maximum limit on the current sourced from the DC bus during active braking. Active brake current limit should be configured to a value lesser than the power supply current limit.

Step 3: Configure ACTIVE_BRAKE_BUS_CURRENT_SLEW_RATE. This sets the slew rate of the DC bus current starting from 0A to ACTIVE_BRAKE_CURRENT_LIMIT.

Step 4: Configure ACTIVE_BRAKE_MOD_INDEX_LIMIT. This sets the modulation index limit below which active braking will be applied. Above this modulation index, device will apply AVS if AVS is enabled. Setting a higher modulation index limit will decelerate the motor much faster.

Note

Modulation index [VOLT_MAG] can be read back from ALGO_STATUS register.

Step 5: MCF8316A uses a PI controller to generate the D-axis reference current. Manually tune ACTIVE_BRAKE_KP and ACTIVE_BRAKE_KI if the active brake current looks unstable.