DRV10983-Q1 Tuning Guide

The DRV10983-Q1 device is a three-phase sensorless motor drivers, featuring an I^2C interface that allows the user to reprogram specific motor parameters in registers and burn them into the EEPROM to help optimize the performance for a given application. This document helps customers quickly set up the DRV10983-Q1, enabling them to experience the powerful performance and flexible programmability of the device.

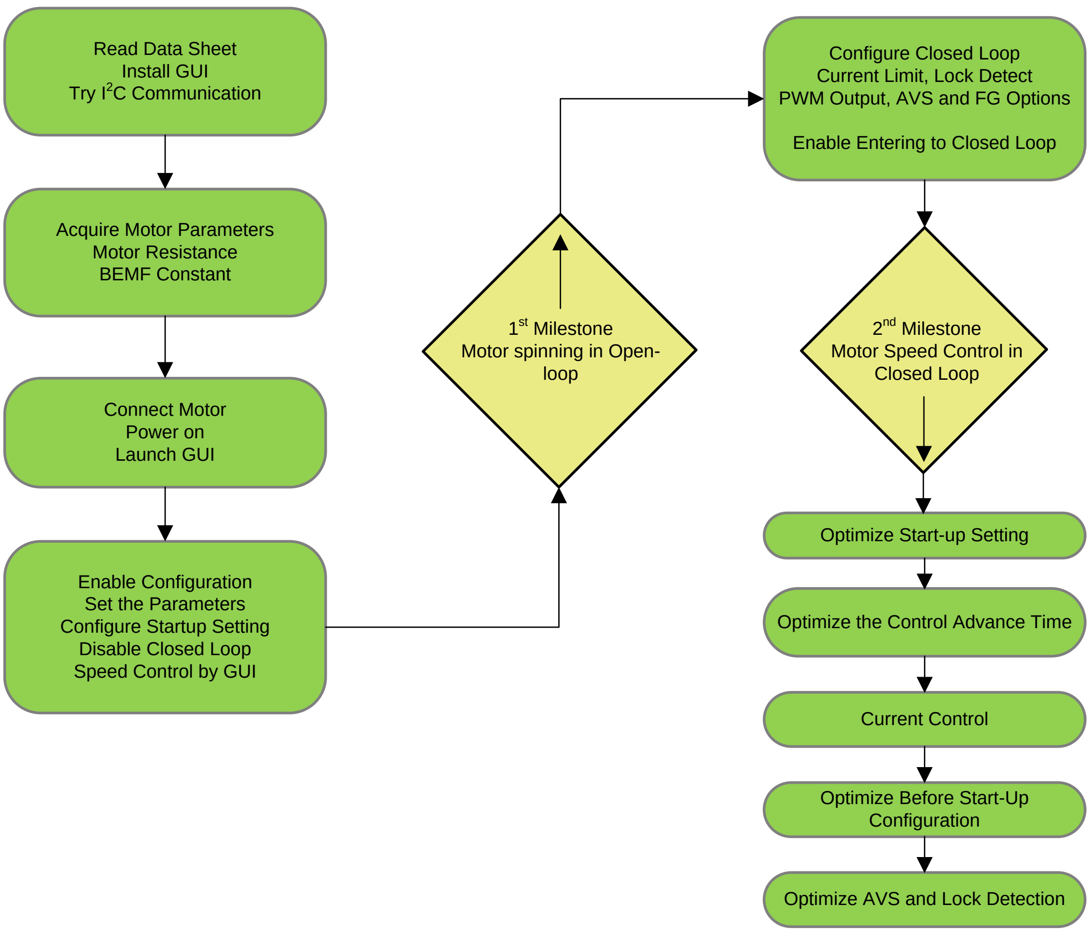


Figure 1. Sequence of Events

Trademarks

All trademarks are the property of their respective owners.

1 Bench Set Up

Before connecting a motor, read the following documents: *DRV10983-Q1 Automotive, Three- Phase, Sensorless BLDC Motor Driver* and *DRV10983-Q1 Evaluation Module User's Guide* . Also install the GUI software. Make sure the I²C communication is working.

2 Acquire Motor Parameter

When a new project is started, the first step is to acquire the motor parameters required by the DRV10983-Q1 to spin the motor (see Table 1). The motor parameters help determine whether it is suitable for the DRV10983-Q1 and what the proper settings are for this motor. The following sections describe how to measure each parameter.

Table 1. Motor Parameter Table

ITEM NUMBER	OPERATION VOLTAGE	MAXIMUM SPEED (RPM)	MAXIMUM CURRENT	R (PHASE-CT)	Kt (PHASE- PHASE)
Motor1				See Section 2.2	See Section 2.3

2.1 Operation Voltage, Number of Poles, Maximum Speed, and Maximum Current

The four parameters, operation voltage, number of poles, maximum speed, and maximum current, should be provided by the motor specification or the application specifications.

2.1.1 Operation Voltage

Typically, operation voltage is fixed with less than 10% error. Some applications adjust motor speed by adjusting the operation voltage. In this condition, record the maximum operation voltage.

2.1.2 Maximum Speed and Maximum Current

Maximum current means the current consumption at steady state (not during accelerating or startup), which normally happens at maximum speed. Maximum speed and maximum current depend on the load condition. Heavier loads cause higher current and slower speed. For a fan application, know the speed and current with the blades assembled. If the full-load speed and no-load speed are both shown in the motor specification, record both.

2.2 Motor Resistance

For a wye-connected motor, the motor phase resistance refers to the resistance from the phase output to the center tap (R_{PH_CT}). This resistance is labeled as R_{PH_CT} in Figure 2.

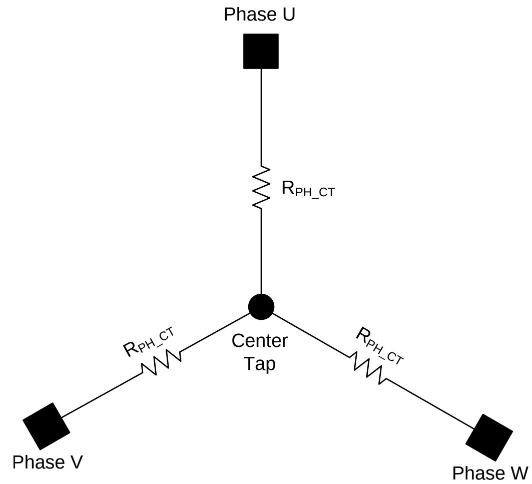


Figure 2. Wye-Connected Motor Resistance

For a delta-connected motor, the motor phase resistance refers to the equivalent phase to center tap in the wye configuration. In Figure 3, this resistance is calculated as $R_Y \times R_{PH\_CT} = R_Y$.

For both the delta-connected motor and the wye-connected motor, the easy way to get the equivalent R_{PH_CT} is to measure the resistance between two phase terminals (R_{PH_PH}) and then divide this value by two: $R_{PH\_CT} = ½\ R_{PH\_PH}$.

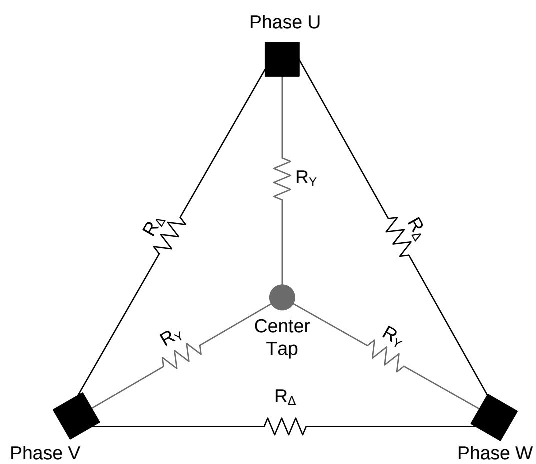


Figure 3. Delta-Connected Motor and the Equivalent Wye Connections

The maximum resistor value (R_{PH_CT}) that can be programmed for the DRV10983-Q1 is 18.624 Ω and the minimum resistor value is 0.0097 Ω (R_{PH_CT}).

2.3 Motor Velocity Constant

The motor velocity constant describes the motor's phase-to-phase back electromotive force (BEMF) voltage as a function of the motor velocity. The measurement technique for this constant as used in the DRV10983-Q1 is shown in Figure 4.

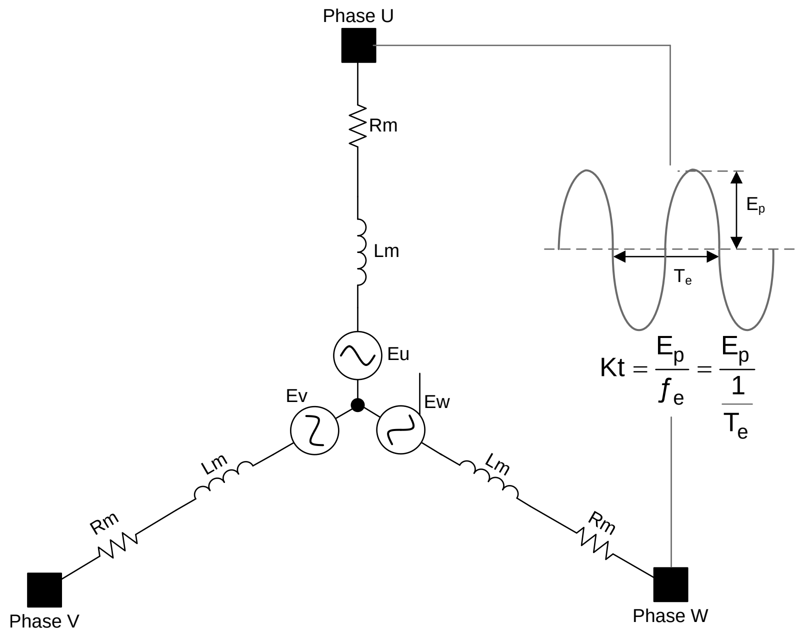


Figure 4. Kt_{PH} Definition

Manually spin the motor quickly or coast it, and use an oscilloscope to capture the differential voltage waveform between any two phases. Use Equation 1 to calculate the motor velocity constant used by the DRV10983-Q1 device.

$$Kt_{PH} = Ep \times Te$$

where

- Ep is ½ the peak-to-peak amplitude of the measured voltage
- Te is the electrical period (1)

The maximum Kt_{PH} that can be programmed in the DRV10983-Q1 is 1766 mV/Hz and the minimum that can be programmed is 0.92 mV/Hz.

If unable to determine the motor velocity constant from the previous steps, the value can be estimated by the maximum motor speed as shown in Equation 2.

$$\text{Motor Velocity Constant} \approx \frac{\text{VCC}}{\text{maximum motor speed (no load)}} \quad (2)$$

For example: the motor spins with 1500 rpm (4 pole) at 24-V power supply:

- 1500 rpm / (60 / 2 pole pairs) = 50 Hz
- 24 V / 50 Hz = 480 mv/Hz

or:

$$\text{Motor Velocity Constant} \approx \frac{\left(\text{VCC} - \text{maximum current} \times \text{resistance} \times \sqrt{3}\right)}{\text{maximum motor speed (full load)}} \quad (3)$$

For example: motor spins with 1000 rpm (4 pole) at 24-V power supply, the current is 1 A and phase resistance is 2 Ω.

- 1000 rpm / (60 / 2 pole pairs) = 33.3 Hz
- (24 V – 1 A × 2 Ω × √3) / 33.3 Hz = 620 mv/Hz

3 Motor Connection and Power On

For a sensorless controller, connect phases U, V, and W of the driver to the three terminals of the motor in any sequence. If the direction of the motor is not the expected direction, swap two of the three wires and the motor direction changes.

Before powering the device on, refer to the *DRV10983-Q1 Evaluation Module User's Guide* . Connect a current probe to phase V to observe the phase current during tuning. Power on the device without connect motor. The power consumption should resemble Table 2:

Table 2. Average Power Consumption

CONDITION	12-V VCC SWITCHING REGULATOR	12-V VCC LINEAR REGULATOR
VCC current	9 mA	11 mA

Launch the GUI software.

4 Spin the Motor With Open Loop Control

After following the steps from Section 4.1 to Section 6.1, the motor can be spun in open loop control.

4.1 Motor Parameters

Enter the motor parameters previously measured and recorded in the table.

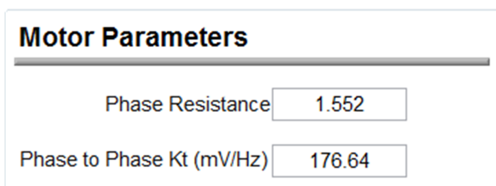


Figure 5. Motor Parameters

4.2 Disable IPD

Uncheck the *Enable IPD* box, disabling the initial position detection (IPD) function.

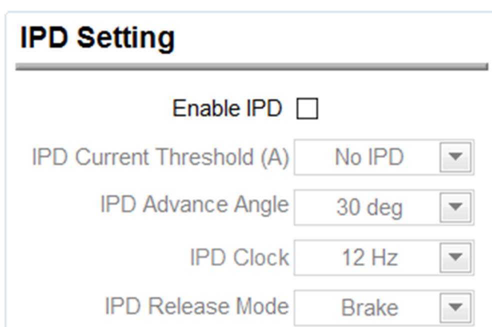


Figure 6. IPD Setting

4.3 Roughly Configure the Before Startup

Uncheck the *Enable Initial Speed Detect* and *Enable Reverse Drive* boxes. Set the *Brake Done Threshold* to *No Brake*.

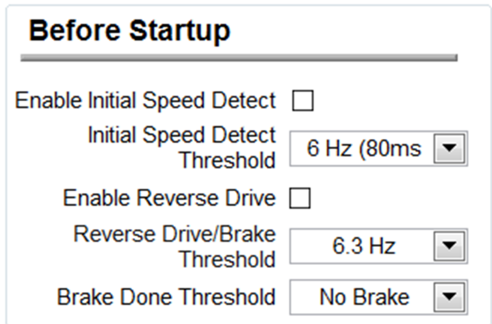


Figure 7. Before Startup

4.4 Configure the Startup Setting

Step 1. Set the *Align Time* to the maximum value (5.3 s) to allow time to measure the phase current.

Step 2. Set the *Open to Closed Loop Threshold* at around one-third to one-fifth of the maximum motor speed. For example, if the motor maximum speed is 80 Hz, set the *Open to Closed Loop Threshold* to 25.6 Hz.

Step 3. Set *First Order Accelerate* and *Second Order Accelerate* based on motor inertia and *Open Loop Current rate* settings. Heavier motors need to accelerate slower. If unsure, try the *Slow* acceleration setting: 2.4 Hz/s and 0.22 Hz/s^2, respectively. The *Open Loop Current rate* should remain the same for now.

Step 4. Based on the SoftStart requirement of the application, select the *Open Loop or Align Current*. Select the 0.4 A / 0.3 A option if the application has no requirement, or if unsure.

Step 5. Make sure the *CLoopDis* is checked, disabling the close loop control. This way the open loop control can be tested and verified first.

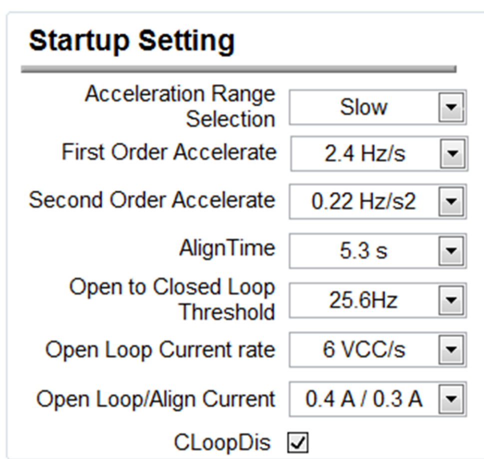


Figure 8. Startup Settings

4.5 PWM Output and Slew Rate Options

The DRV10983-Q1 device can be configured to work with slew rate to improve EMC and EMI performance. Dead time must be changed to account for this change in slew rate. Table 3 provides guidance on dead-time settings for various slew-rate settings. The recommendation is to start with 120 V/µs and a 440-ns setting. The PWM output has two options: 25 kHz and 50 kHz. Uncheck the *Double the output PWM frequency* box initially and operate at 25-kHz PWM output frequency. Motors that are required to operate at a higher speed (>500 Hz) or have lower inductance require higher PWM frequency. In such cases, check *Double the output PWM frequency* box to increase output PWM frequency to 50 kHz.

Table 3. Dead-Time Settings

Slew-Rate Setting	Dead-Time Setting
35 V/µs	1.08 µs
50 V/µs	800 ns
80 V/µs	600 ns
120 V/µs	440 ns

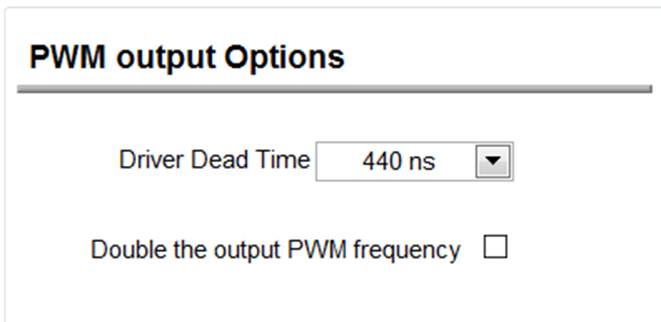


Figure 9. PWM Output Options

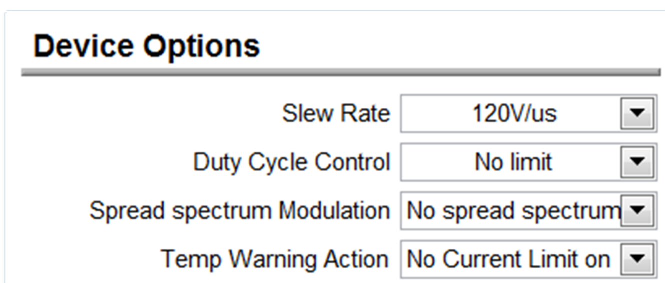


Figure 10. Device Options—Slew Rate

4.6 Spin the Motor in the Open Loop

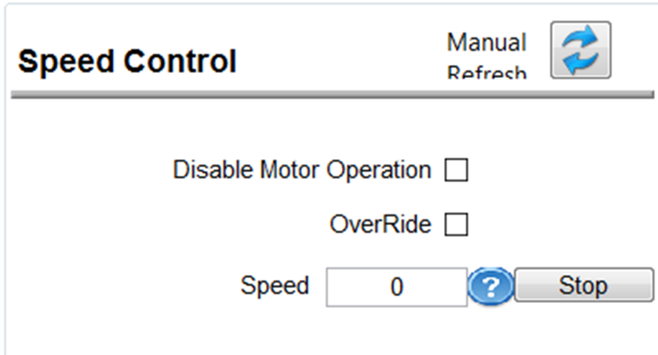


Figure 11. Speed Control Settings

Step 1. Check the *OverRide* box, enabling speed command by I²C.

Step 2. Input a non-zero speed. The motor should start to spin in open loop. The rotating speed should be the *Open to Closed Loop Threshold*. Note that the open loop operation speed is not determined by the value of *Speed* command; it is always at *Open to Closed Loop Threshold*.

Step 3. TI suggests using *300* as *Speed* command to smoothly transfer to closed loop (Section 5).

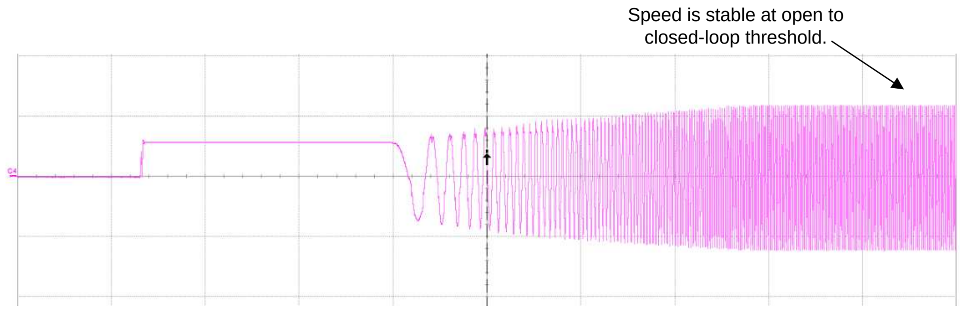


Figure 12. Speed Waveform

5 Enter the Closed Loop

If the BEMF of the motors is sinusoidal, a smooth sinusoidal current waveform should be seen in open loop operation. But the efficiency in open loop operation is not good. Also, open loop operation is very unstable, putting some external load on the motor makes it stop.

To spin the motor in closed loop, we need to configure the following parameters. These settings can be configured while the motor is spinning in open loop because it will not affect the open loop operation.

5.1 Configure the Closed Loop Setting

5.1.1 AdjMode

Set the *AdjMode* to *Full Cycle* adjustment.

Figure 13. AdjMode Setting

Choose *Half Cycle* adjustment only when the motor Kt is very large and the maximum speed is low. It can make the motor dynamic response faster but the potential risk is the phase sinusoidal current gets distorted.

Also, *Half Cycle* adjustment can solve the *BEMF Abnormal* lock detect issue described in Section 11.

5.1.2 Speed Input Mode

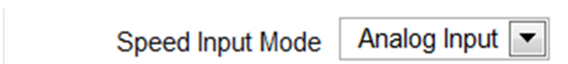


Figure 14. Speed Input Mode Setting

If the system provides an analog speed command, choose *Analog Input*. If the system provides PWM speed commands, choose *PWM Input*.

If the PWM frequency is beyond the DRV10983-Q1 application range (0.1 kHz to 100 kHz), convert the PWM input into an analog voltage using an RC filter and select the *Analog* mode instead.

If using I^2C speed commands (checking the *OverRide* box enables I^2C speed commands), both the analog input and digital input will be ignored. In the following description, I^2C speed commands are used.

5.1.3 Closed Loop Accelerate

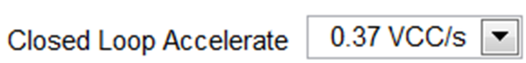


Figure 15. Closed Loop Accelerate Setting

To prevent sudden changes in the torque applied to the motor which could cause high currents (accelerate) or VCC voltage surges (decelerate), buffer the speed command with *Closed Loop Accelerate*.

Closed Loop Accelerate is a supplementary method to the software current limit and mechanical anti-voltage surge (AVS). If the software current limit and mechanical AVS are both working properly, *Closed Loop Accelerate* can be set at *Inf fast*. At some particular motor parameter conditions (refer to the Section 8.2 and Section 10); software current limit and mechanical AVS cannot work as expected. *Closed Loop Accelerate* is the direct way to adjust the buffered speed command.

TI suggests having some level of *Closed Loop Accelerate* setting in all applications. If the speed response is very strict in the application, increase the rate, otherwise, decrease the rate.

Set *Closed Loop Accelerate* to 0.37 VCC/s now; then continue to the next steps.

5.1.4 Control Coefficient

Set the control coefficient to 1.

5.1.5 Commutate Control Advance Mode and Setting

- In *Commutate Control Advance Mode*, choose *Constant Time*.
- For the *T Control Advance* setting, start with 220 µs. Section 7 describes how to optimize this parameter.

Figure 16. T Control Advance Setting

5.2 Roughly Configure for Other Settings

5.2.1 Configure the Current Limit

Disable the *Software Current Limit Function* and set the Current Limit for Lock Detection Control by selecting *Use Range 2* from the *Current Limit for Lock Detection Control* dropdown menu.

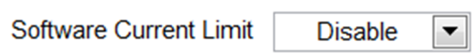


Figure 17. Current Limit Settings

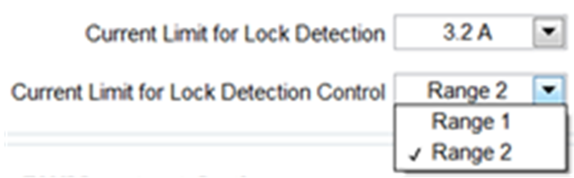


Figure 18. Hardware Current Limit Settings

5.2.2 Configure Lock Detect

Enable all the *Lock Detect* options except the *BEMF Abnormal*, this feature is discussed in the *Lock Detect* section. Set the *Abnormal Kt lock detect Threshold* to Kt_high = 2 Kt, and Kt_low = 1/2 Kt.

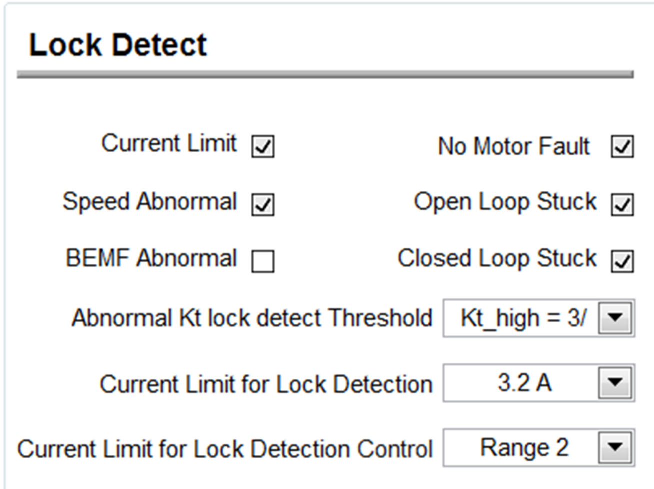


Figure 19. Lock Detect Settings

5.2.3 Clear Lock Detection Status Bits

After any lock event, the lock status bit is set until the bit is manually reset. To clear status bit, click the *Clear Fault* button.

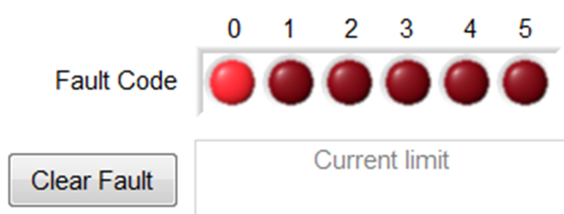


Figure 20. Fault Code for Lock Detection

5.2.4 Configure the Anti-Voltage Surge (AVS) Function

Start with *AVS Function* disabled:

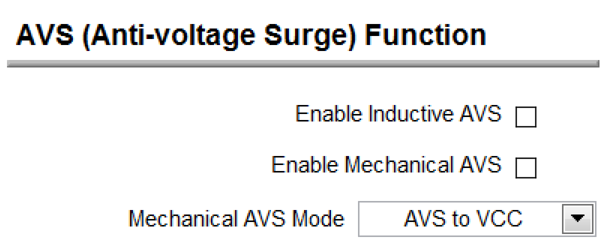


Figure 21. AVS Function Settings

5.2.5 Configure FG Options

Keep the *FG Options* at the default settings.

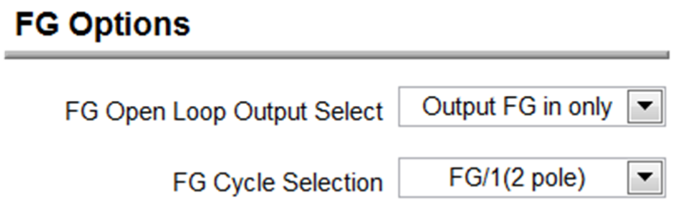


Figure 22. FG Options

5.3 Enter Closedloop

While the motor is spinning in the open loop, enable the closed loop control.

Figure 23. CLoopDis

The motor speed should increase (if speed command = 300) and the phase current during open-to-closed loop transition should resemble Figure 24:

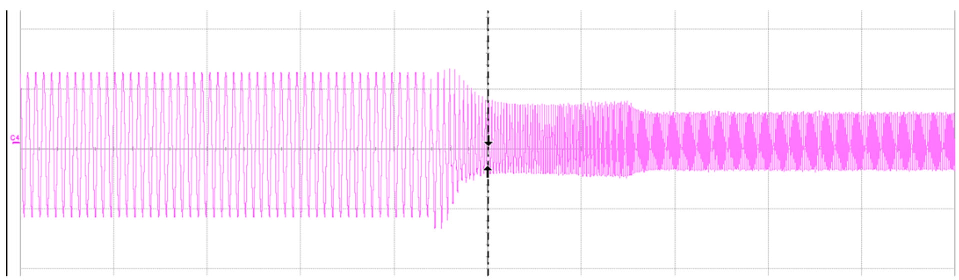


Figure 24. Waveform at Speed Command = 300

Now, adjust the motor speed by entering the *Speed* command in the box (from 0 to 511):

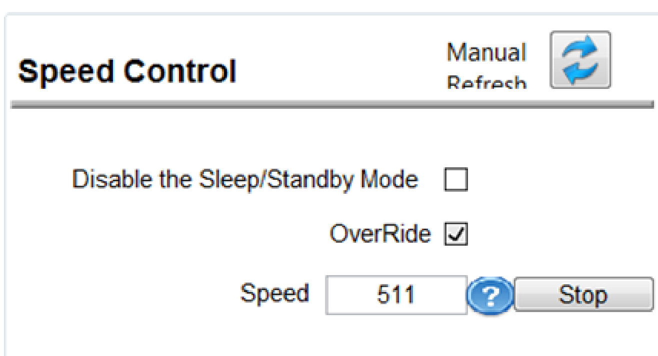


Figure 25. Speed Control Settings at 511

Figure 26 shows phase current with speed control:

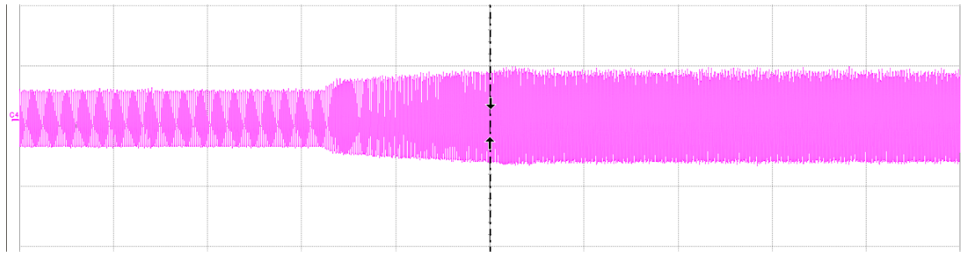


Figure 26. Phase Current With Speed Control Waveform

6 Optimize Motor Startup Setting

6.1 Evaluate the Calculated Kt Value

Observe the waveform for the phase V current and compare it to the following pictures in Figure 27:

- Picture 1: the parameter entered for Kt is correct.
- Picture 2: the Kt entered is too big.
- Picture 3: the Kt entered is too small.

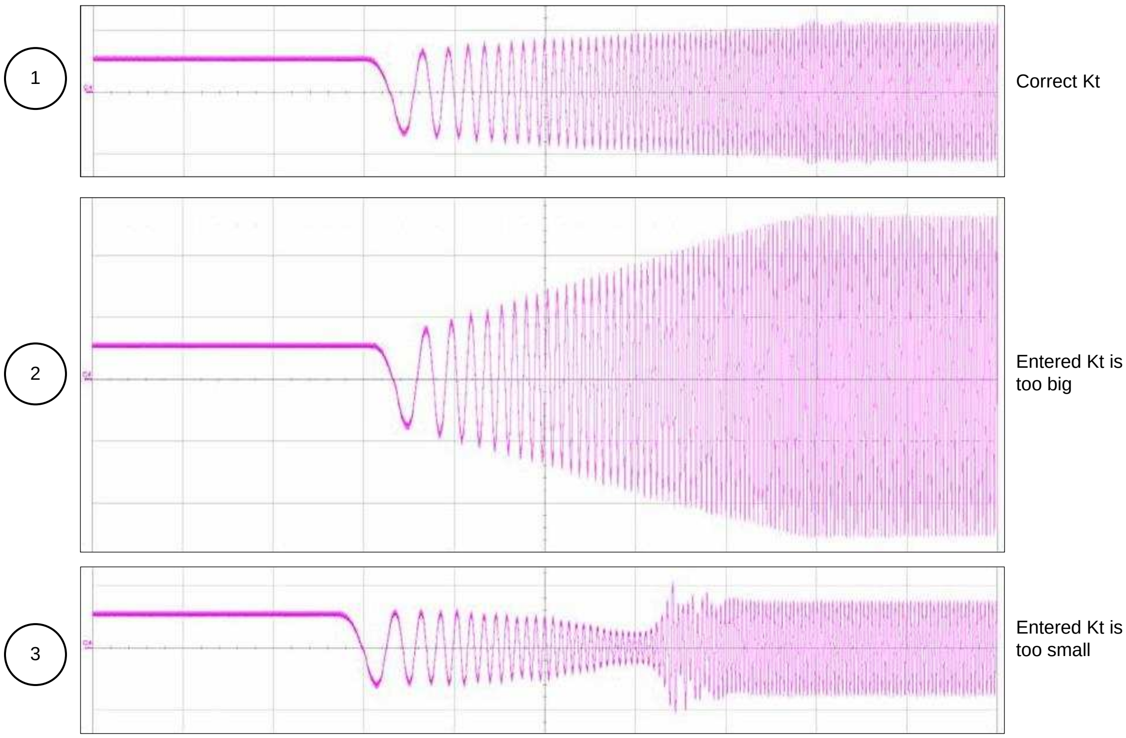


Figure 27. Waveform Comparison

The motor should have stopped at the time denoted in Figure 28.

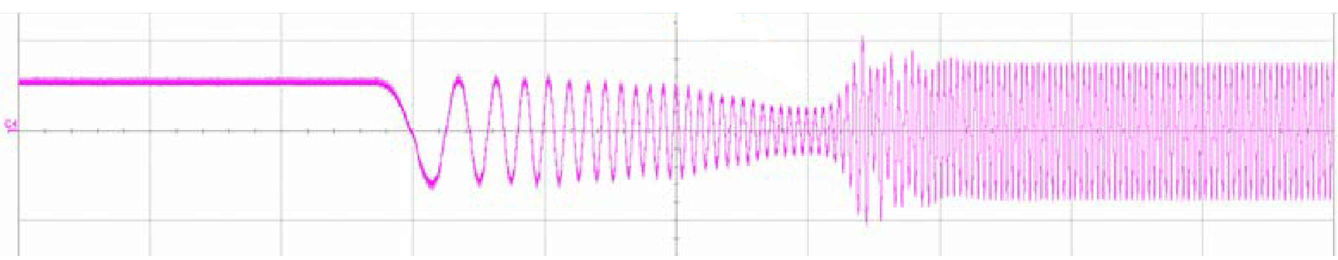


Figure 28. Motor Stopped Waveform

An alternative way to measure the Kt is by trying different Kt values until there is a fairly constant envelope current (Picture 1) during open loop. It is normal that the current envelope slightly increases while the speed is increasing.

During this measurement, to look at the startup again, set the speed to 0 (use the *Stop* button), after the motor stops, set it back to a non-zero value again.

Figure 29. Speed Settings

6.2 IPD

For applications where a reverse spin is not acceptable, the IPD function is an alternative way to initialize the motor. With the proper IPD setting, the motor startup is also faster.

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.

If IPD is chosen as initialization method, we need to enable IPD and configure the IPD setting section. At the same time, because the align method is not used, the AlignTime can be set to the minimum value.

Figure 30. AlignTime Setting

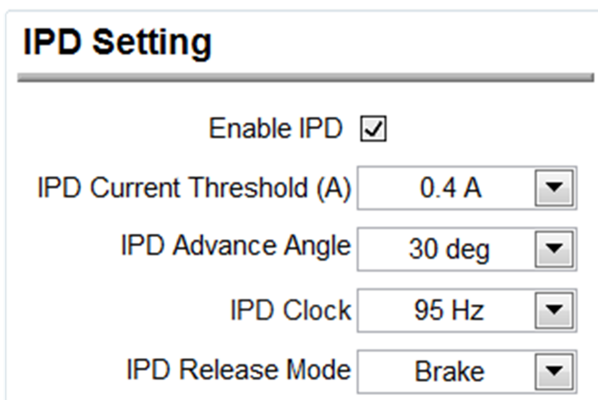


Figure 31. IPD Settings

6.2.1 IPD Enable and Current Threshold

IPD current threshold is selected based on the inductance saturation point of the motor. However, normally we are not able to find out this exact number by specification or calculation. We need to choose this number by experience.

A higher current has better chance to accurately detect the initial position. On the other hand, higher current will result in vibration and noise; also the current should not be higher than the maximum current the motor can handle.

Note that it is possible to not be able to find the proper settings for a particular motor for the IPD function. This is either because the current causes too much noise and vibration, or the current is not sufficient to accurately detect the initial position of the motor. In this case the align and go method should be used.

Modifying the motor can make the IPD work properly, contact a TI representative for further advice.

6.2.2 IPD Advance Angle

IPD advance angle is the driving angle after the initial position of the motor is identified. The suggested angle is 30 degrees.

6.2.3 IPD Clock

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.

Looking at Figure 32, the current does not settle completely, which means the clock is too fast for this motor. This will result in IPD not being able to reliably identify the initial position of the motor.

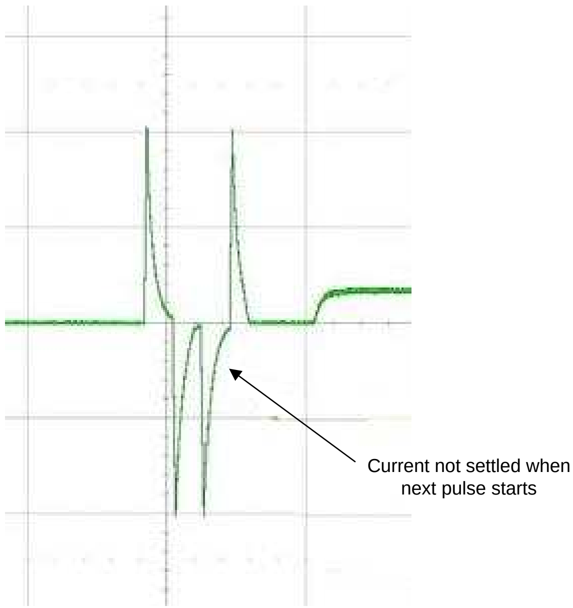


Figure 32. Current Graph

6.2.4 IPD Release Mode

IPD release mode can be selected either as *Brake* or *Tri-state*.

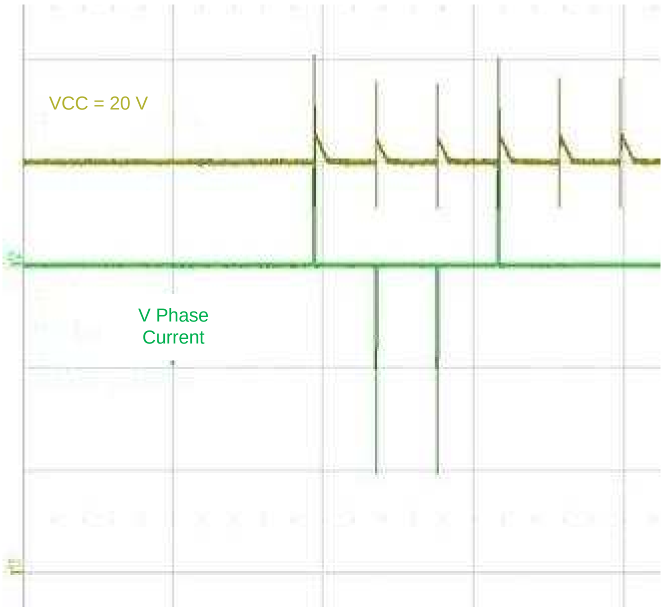


Figure 33. VCC Overshoot in Tri-State Mode

If the system input capacitor is not big enough, the *Tri-state* option will cause the VCC voltage to pump up, which should be avoided.

The advantage of *Tri-state* mode is that the current settle down time is significantly reduced.

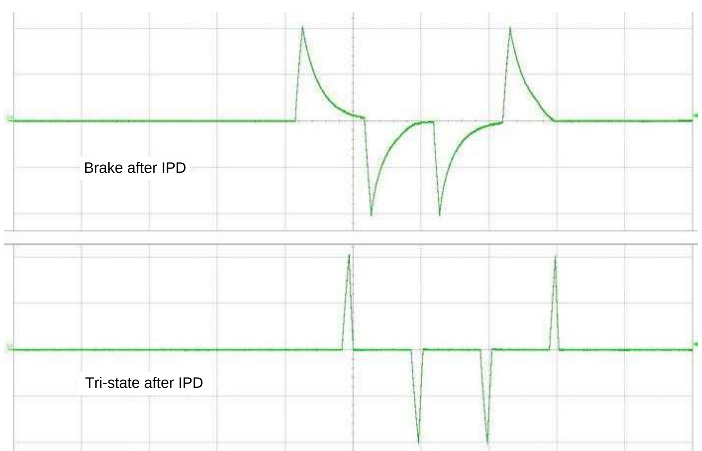


Figure 34. Brake versus Tri-State after IPD

6.2.5 Whether to Use IPD

To decide whether to choose IPD or align and go, compare the startup difference of the two techniques. Normally, if the part of the motor that rotates is not visible and the system can tolerate reverse rotation of the motor, for example a cooling fan in a laptop or a water pump, we suggest using align and go. If the rotating part of the motor is visible to the user or if the system cannot tolerate reverse rotation of the motor, use the IPD technique.

6.3 *Improve the Align Time and Accelerate Time*

The *Startup Settings* have been set to a safe value to make sure the startup was successful. However, it might not be sufficiently fast enough to meet the startup time requirement. Use the following sections to minimize the startup time.

6.3.1 Handoff Threshold Optimization

To start the motor reliably, there is a minimum speed that is required to achieve a certain BEMF voltage. A low handoff threshold causes the handoff to occur before the BEMF is at the adequate level, resulting in startup failure.

However, a high handoff threshold will result in an over shoot when the closed loop speed command is low. For example, the threshold is 100 Hz and speed command is 50 Hz, overshoot appears in the startup.

We suggest the handoff speed to be set between one-fifth and one-third of the maximum speed. The customer may also want to try a lower threshold so that the motor ramp from zero speed to full speed is faster. This is because the motor accelerates faster in closed loop. Please ensure the startup reliability at a different initial position when we set the handoff threshold lower than one-fifth of the maximum speed.

6.3.2 AlignTime and Accelerate

Disable the transition to closed loop to optimize the *AlignTime* and acceleration of the system so that any failures related to closed loop operation will not interfere with correctly tuning.

If IPD is implemented in the application with the *Acceleration Range Selection* set to *Slow*, set the *AlignTime* to 0.04 s, and set *Second Order Accelerate* to 0.22 Hz/s2. Keep increasing the *First Order Accelerate* step by step until there is startup failure; select the last value before the failure occurred.

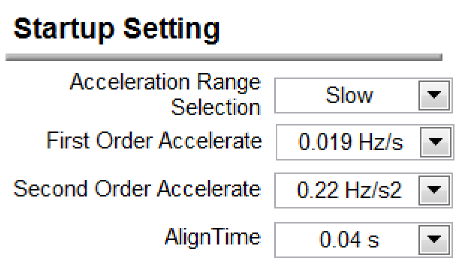


Figure 35. Startup Settings

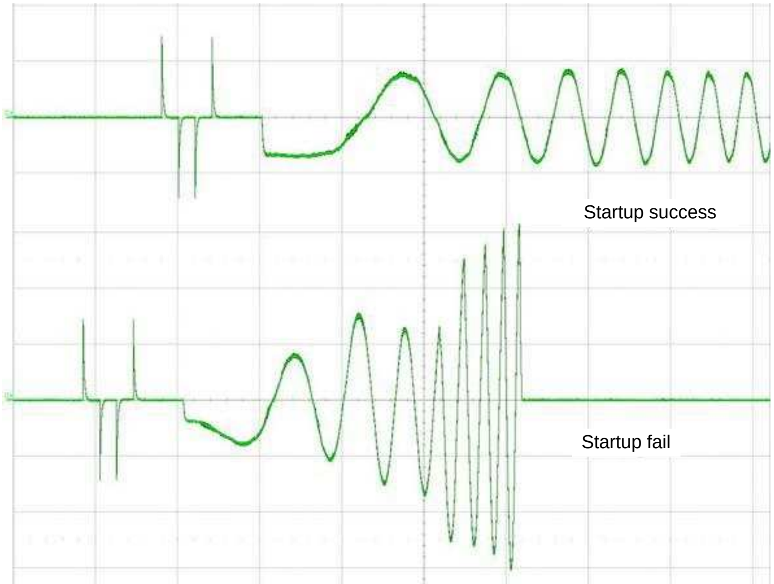


Figure 36. Startup Success versus Fail

If the IPD is not enabled, there are two options to align and accelerate the motor:

- The traditional align and go is to set the second order accelerate to 0.22 Hz/s2, then set the align time to be long enough that the motor can basically stop during align. Figure 37 show a sufficient align time and an insufficient align time.

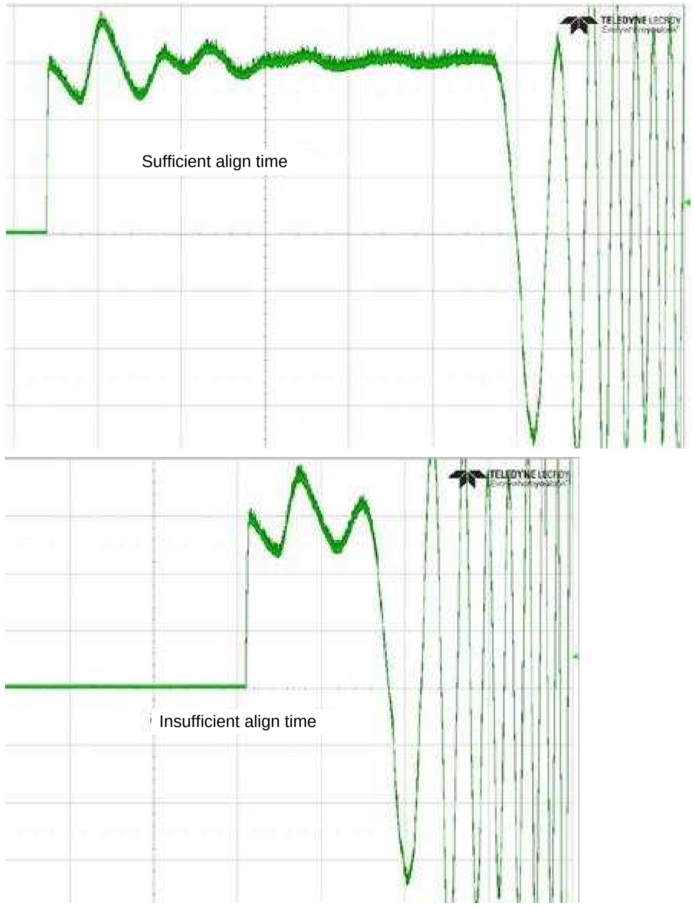


Figure 37. Sufficient versus Insufficient Align Time

After the align time is selected, increase the *First Order Accelerate* step by step until the startup fails.

Because of the complexity of the motor and the load (including friction, cogging, and initial position), the open loop setting is mainly based on experience. Many tests need to be taken to ensure the startup reliability. The simple method to verify the settings follows:

(a) First find the position where it is the most difficult to spin up the motor (the *Dead Point*).

(b) Label an initial position A on the fan, power on, look at the direction it starts to move. If the direction is counterclockwise, find another position B by tracking toward the clockwise direction. If it is clockwise, find the position B by tracking toward the counter-clockwise direction. Line up B in the prePower on, look at the direction it starts to move. If it is the same as position A, go further to find point B. If it is opposite, the *Dead Point* is between A and B.

(c) Find a point in between A and B, power on, look at the direction it starts to move. If the direction is towards B, the *Dead Point* is in between A and new point, relabel the new point B. If the direction the motor moves is towards A, the *Dead Point* is in between B and the new point, relabel the new point A.

(d) Repeat Step b. A and B become closer and closer until A and B almost overlap each other. This is the *Dead Point*.

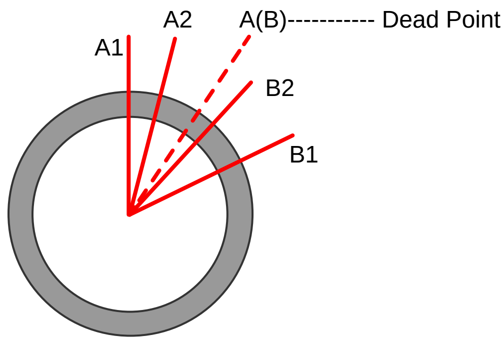


Figure 38. Dead Point

(e) Try start up at this *Dead Point* for several times, if it is safe at this point, it will be safe at other points.

In some applications, 100% successful startup at all the initial positions is not necessary. Especially when we want to have the *Dead Point* successfully startup, we will have to make the startup very slow. The DRV10983-Q1 can retry if the first start up fails.

Make the tradeoff between startup success rate and startup time.

- The other option is dynamic align and go, set the align time to 40 ms, set the *First Order Accelerate* and *Second Order Accelerate* to meet the startup time requirement. Here are the two steps for adjustment:

  (a) Set the second order acceleration to '0.22 Hz/s^2'. Make sure the startup at all positions is successful. If the startup fails at any positions, reduce the *First Order Accelerate*.

  (b) Increase the *Second Order Accelerate*, until the current has vibrations at the end of open loop. This is the indicator that the second order acceleration has reach the maximum value. If we continue to increase it, the startup will fail.

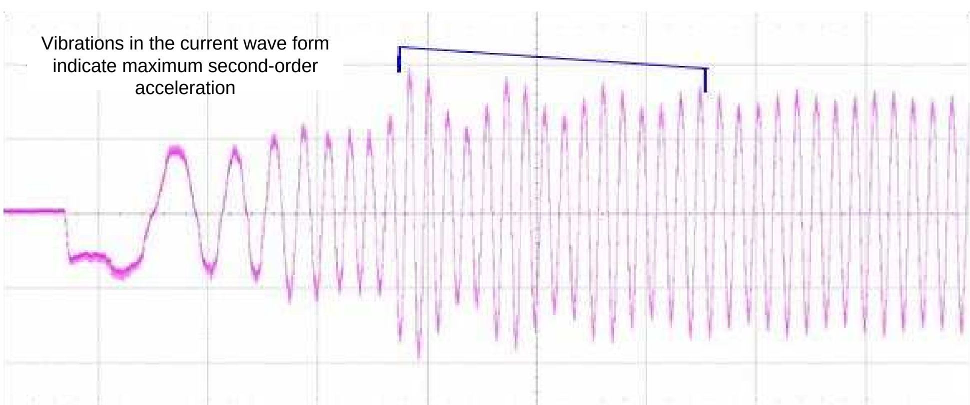


Figure 39. Current Waveform Vibrations

7 Optimize the Control Advance Time

7.1 Motor Electrical Time Constant (LR Time Constant)

The electrical time constant (T_{LR}) in a BLDC motor is defined as motor phase-to-phase inductance divided by phase-to-phase resistance.

Please note that phase-to-phase inductance and phase-to-phase resistance are used to calculate the T_{LR}. Phase-to-center tap (CT) inductance divided by phase-to-CT resistance does not have the same result:

$$L\ PH - PH\ R\ PH - PH \neq L\ PH - CT\ R\ PH - CT \quad (4)$$

- For example, winding U→V→W is put on the iron core:

– RUV = RVW. So, RUW = 2 × RUV
– LUV = LVW, but LUW ≠ 2 × LUV, because of the mutual inductance

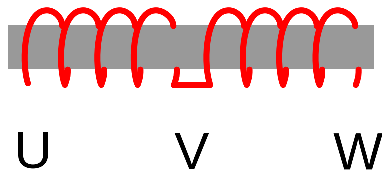


Figure 40. Mutually Coupled Windings

Some multi-meters and LRC meters have an inductance measuring function. However, the result is normally the equivalent impedance of the LR circuit, not the pure inductance value. When it is converted to the inductance value, it is not accurate.

To measure the LR time constant, do not record the L value or the R value; instead, measure the current ripple value when the motor is started by the DRV10983-Q1 . A larger LR time constant results in a smaller current ripple.

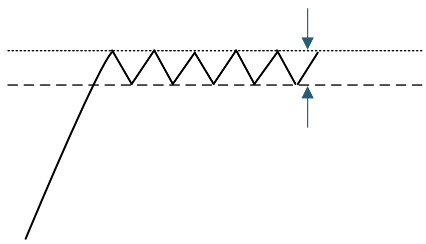


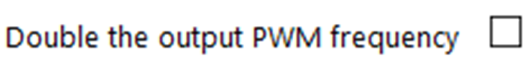
Figure 41. Current Ripple Value

When running the motor, set the align time to the maximum value and measure the V phase with a current probe. During the initial start, V phase current ripple represents the LR time constant.

7.2 Measure the LR Time Constant

The LR time constant is important to the system because it affects how long the control advance time is. A higher LR constant means a higher control advance time. This control advance time will be set in Section 7.

Set the *Align Time* back to 5.3 seconds for this measurement, this large of a value is not necessary for most applications. The reason we set it to 5.3 seconds is so the phase current can be captured during the align process to measure the LR time constant by finding the current ripple at this condition. The current ripple should be 25 kHz if the DRV10983-Q1 output PWM frequency is set to the default 25 kHz (but verify under advanced settings: *PWM Output Options* [Section 4.5]).



The unchecked box is 25 kHz PWM.

The amplitude of the ripple represents the LR time constant:

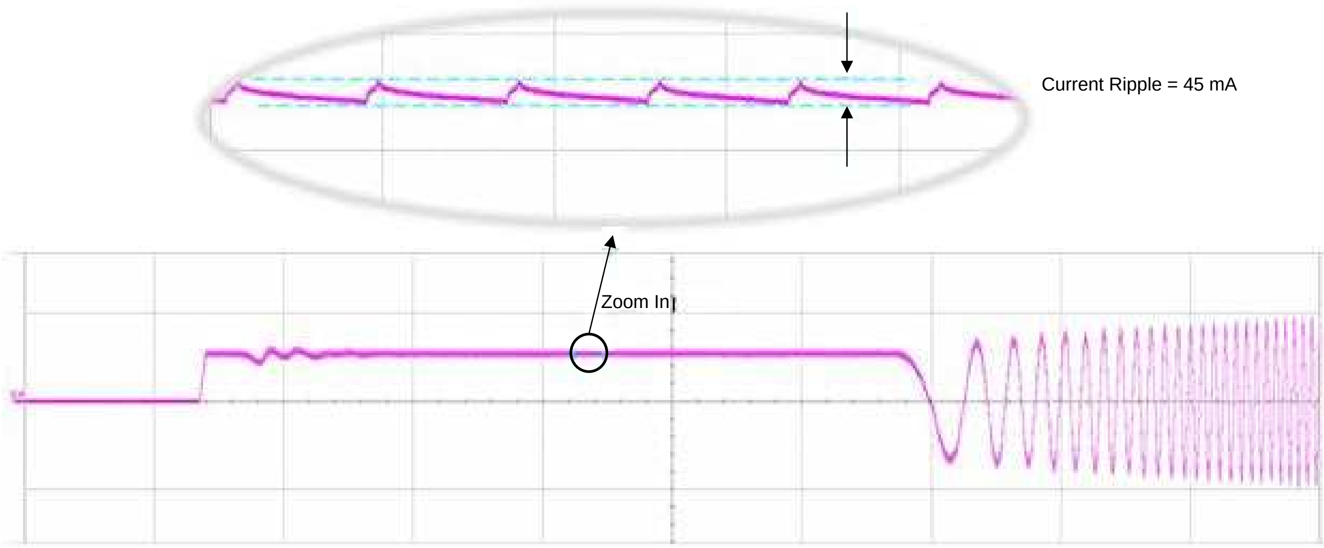


Figure 42. Ripple Amplitude

Put this current ripple value into the parameter table:

Table 4. Motor Parameters

MOTOR NAME	OPERATION VOLTAGE	MAXIMUM SPEED (RPM)	MAXIMUM CURRENT	R (PHASE-CT)	Kt (PHASE- PHASE)	LR CONSTANT
TI_M1	12 V	3000	450 mA	3 Ω	100 mV/Hz	45 mA

Use Equation 5 to calculate the estimated control advance times.

T Control Advance(s) ≈ 20 µs × Imax Current Ripple × Open Loop Current × R VCC

where

- Imax is the current at the maximum speed with full load.
- R is motor phase resistance.
- Current Ripple is current ripple measured during align state (in this example: 45 mA) (5)

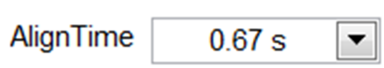


Figure 43. AlignTime Setting

7.3 *Control Advance Adjustment*

The DRV10983-Q1 have a control advance adjustment function to optimize the motor's operational efficiency. While the motor is spinning, the increases or decreases in the control advance angle will result in a phase voltage and phase current shift. At the same time, motor speed and supply current are also affected.

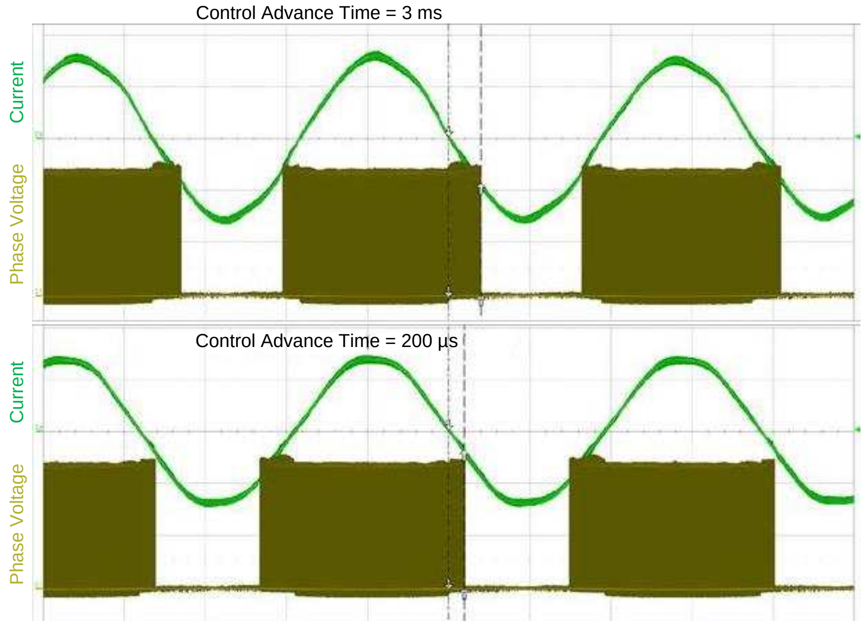


Figure 44. Control Advance Time Comparison

The criteria to find out the optimized control advance angle is

1. the motor should spin smoothly and the phase current waveform is stable, and
2. find out the smallest supply current at the same speed.

8 Current Control

The DRV10983-Q1 are able to control the motor phase current in startup and closed loop. It can also provide protection when current exceeds a set limit.

8.1 Open Loop Current Setting

The *Open Loop Current rate* setting starts the motor at the lowest current that will get the motor running. The higher the current setting, the more likely the motor will move; however, it is very inefficient to use such a large current for smaller devices.

The *Open Loop Current rate* setting has four options: 0.2 A, 0.4 A, 0.8 A, and 1.6 A. The selection depends on requirements for startup time, power supply capacity, and customer preference.

To have a SoftStart, set the *Open Loop Current* to 0.2 A; for a fast start, set it to a higher value such as 1.6 A. Normally, we suggest it be less than the full-speed operation current.

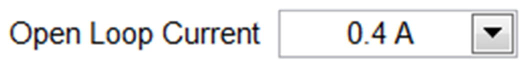


Figure 45. Current Setting

Figure 46 shows the different startup current settings:

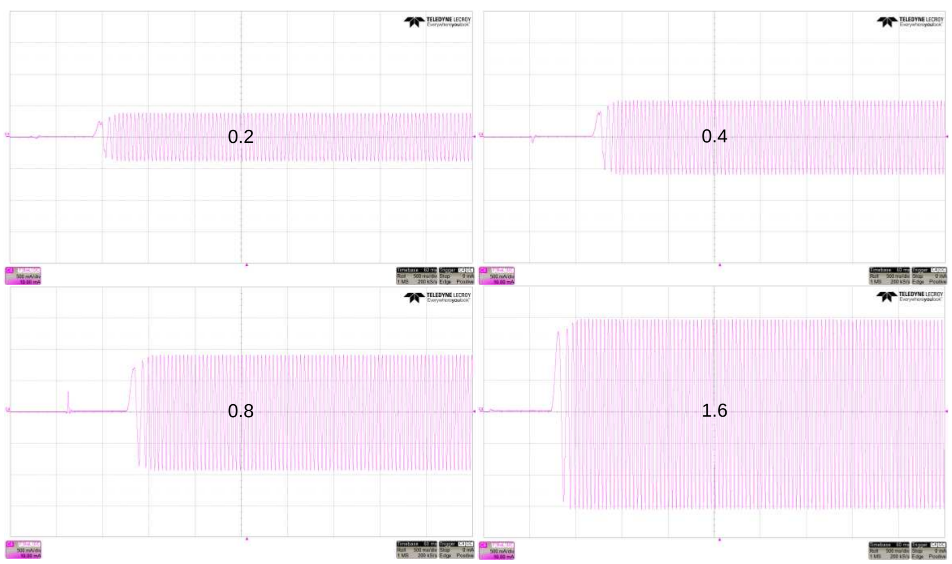


Figure 46. Waveforms of 0.2, 0.4, 0.8, and 1.6-A Currents

To avoid the acoustic noise caused by the fast slew rate of the driving phase, the DRV10983-Q1 provide current ramp rate options ranging from 0.023 VCC/s to 6 VCC/s. Refer to *DRV10983-Q1 Automotive, Three-Phase, Sensorless BLDC Motor Driver* for more information.

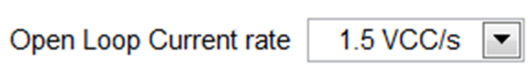


Figure 47. Open Loop Current Rate Setting

8.2 Software Current Limit

The software current limit is useful at transitions from open loop to closed loop and also when the motor is accelerating. During these two conditions, the current amplitude may increase without the software current limit. Please note that implementing the software current limit will slow down the motor's acceleration. Consider the tradeoff between the acceptable current and the acceleration based on their system requirements.

Because the *Closed Loop Accelerate* function works as a buffer to slow down the acceleration and limit the phase current, it actually provides a similar function as software current limit. We may not be able to see the effect of the software current limit if we have set *Closed Loop Accelerate*. Figure 48 is captured with closed loop accelerate as *Inf fast*.

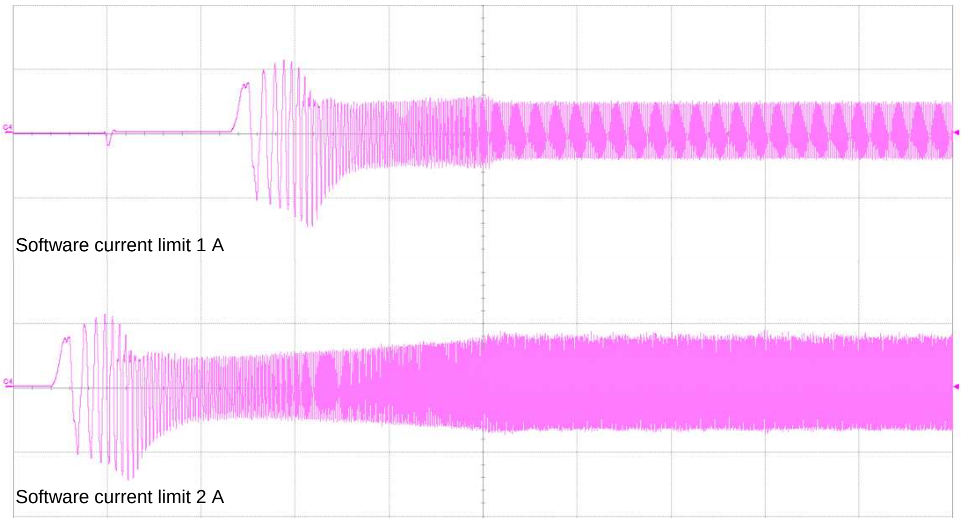


Figure 48. Software Current Limit Comparison

The criteria for setting the *Software Current Limit Threshold* are:

- The current setting should be higher than the current at the maximum speed with full load.
- The setting should be lower than the capability of the power supply and lower than the hardware current limit threshold.

For example, if the motor requires 1.5 A with full load at full speed, and the power supply provides no more than 2-A current, set the *Software Current Limit Threshold* to 1.8 A.

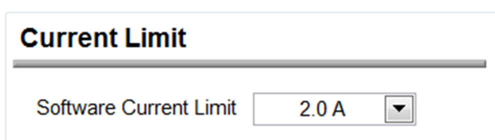


Figure 49. Software Current Limit Setting

Sometimes, because of the implementation of the software current limit, the motor is not able to accelerate to the target speed. In this case the *Speed Cmd Buffer* is lower than the *Speed Command*.

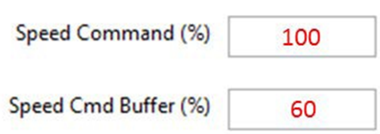


Figure 50. Speed Command Setting

The motor is not able to accelerate to the target speed is because:

- The programmed Kt value is wrong and set too low. The solution is increasing the Kt value and checking whether the motor startup improves. If the difference between *Speed Command* and *Speed Cmd Buffer* gets closer, we need to continue increasing the Kt until they are the same.
- The motor has very low phase resistance; the Kt programmed is slightly lower than the correct value. However, if the value is increased by even one step, the motor cannot be controlled at low speed. In this case, choose the higher Kt and disable the mechanical AVS, or choose the lower Kt and increase the *Software Current Limit Threshold* (or disable the software current limit).

If the software current limit is disabled, control the *Speed Command* slow enough (or select the slow *Closed Loop Accelerate* setting) to prevent a big inrush current.

Note that when the motor resistance and current are low, the software current limit is not always accurate because of the resolution of Kt programming. *Closed Loop Accelerate* is a complementary method in this condition.

8.3 Current Limit for Lock Detection

The Current Limit for Lock Detection is designed to protect the motor when it is blocked by an external force. It operates in both open loop and closed loop. The threshold is programmable and should be set to a value greater than 1.2 times the software current limit. It should be set less than the short time current the system can handle, which is determined by the motor, the device, the power supply and the system specification. There are two ranges to chose from: range 1 up to 2.5A and range 2 up to 3.2A.

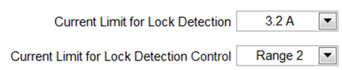


Figure 51. Current Limit for Lock Detection Threshold Setting

Figure 52 shows that when the hardware current limit feature is triggered, the device stops driving the motor and waits for five seconds to retry (In this figure, the startup failed again at retry because the current at this threshold is not sufficient to make the resynchronization).

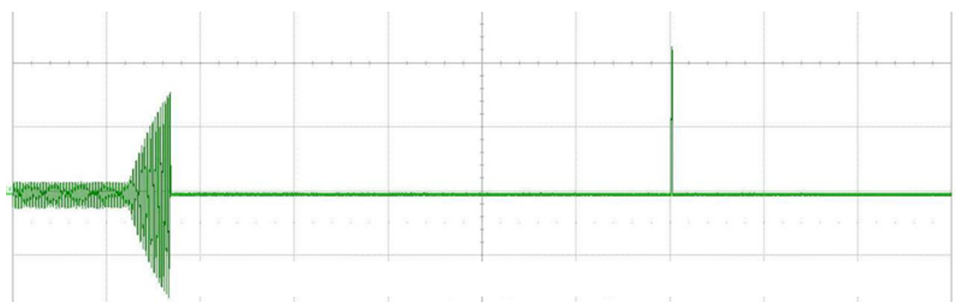


Figure 52. Hardware Current Limit (1.6 A) and Retry After 5 s

Also, the register bit will be set and the GUI indicator will turn on when the hardware current limit is triggered. Click the *Clear Fault* button to clear the fault.

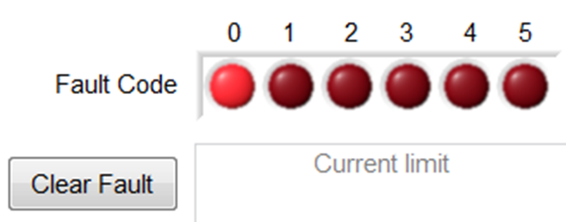


Figure 53. Fault Code for Lock Detection

9 Optimize Before Startup Configuration

9.1 *Initial Speed Detect (ISD)*

Keep *Enable Initial Speed Detect* checked all the time. This avoids waiting for the motor to stop in the condition when motor has initial speed.

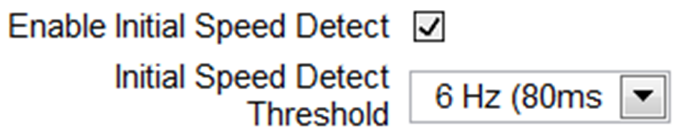


Figure 54. Initial Speed Detect Settings

The *Initial Speed Detect Threshold* is selected based on motor velocity constant and inertia. Theoretically, we should choose the threshold to be as small as possible, because we can resynchronize the motor with lower speed. However, a motor with a low velocity constant at a lower speed has a very small BEMF; it may not be able to correctly trigger the BEMF comparator, causing the resynchronization to fail. So, using a motor with Kt > 1 V/Hz, a 0.8 Hz threshold is chosen and using a motor with Kt < 100 mV/Hz, choose 6 Hz.

9.2 *Reverse Drive*

Unless the application requires two-direction spinning or it is possible for the motor to have reserve speed before startup, the *Enable Reverse Drive* function and *Brake* function should be disabled. Refer to *DRV10983-Q1 Automotive, Three- Phase, Sensorless BLDC Motor Driver* for detailed information.

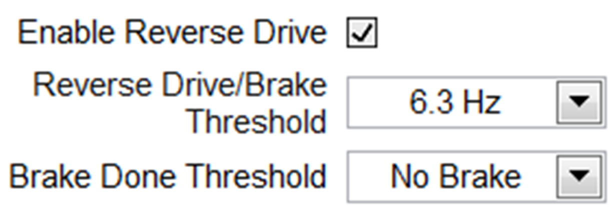


Figure 55. Reverse Drive and Brake Setting

The *Reverse Drive* function is very useful when an application contains a motor with big inertia and small friction. In this condition, it takes a very long time for the motor to coast down to zero speed without the *Reverse Drive* function (for example, a ceiling fan).

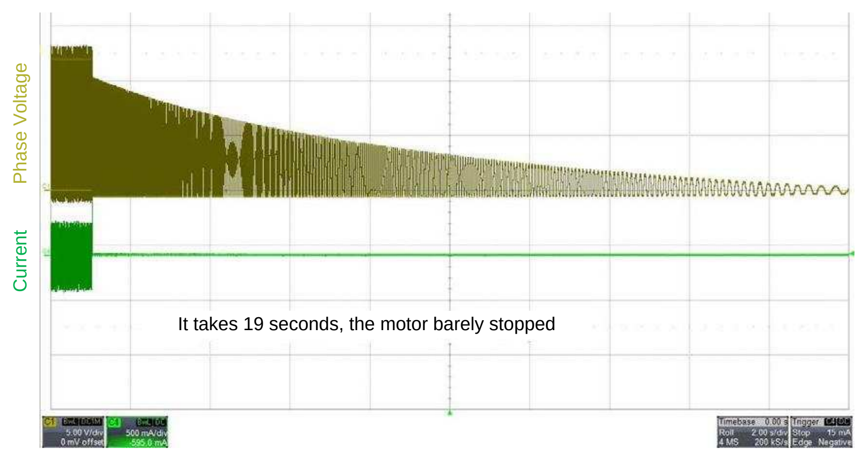


Figure 56. Reverse Drive Waveform

9.3 Brake

The *Brake* function and *Reverse Drive* function are mutually exclusive. The *Reverse Drive/Brake Threshold* needs to be reasonably low. If it is set too high, the current caused by the BEMF may damage the motor or the device.

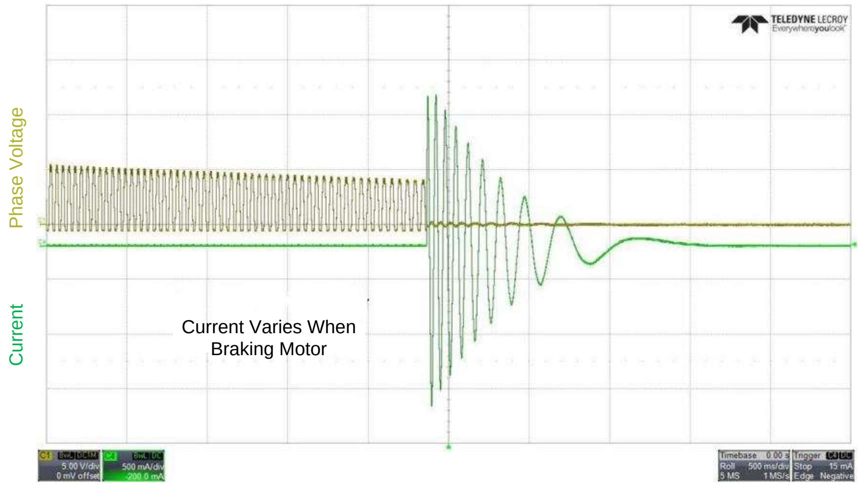


Figure 57. Brake Waveform

Brake Done Threshold determines how to finish the brake state. Bigger Kt and bigger inertia motors need a longer time while smaller Kt and smaller inertia motors need shorter time. If *Brake Done Threshold* is set too long, the startup time is unnecessarily long. If it is set too short, it is possible that the motor will not completely stop before attempting to start up again. This is especially important if we choose to use the IPD to start the motor. If the motor is not completely stopped, the IPD result will not be correct.

If the *Align&Go* is selected to start the motor, we don't need to brake the motor to a complete stop, we can even remove the brake state as long as the align state is sufficient to stop and position the motor.

9.4 Verify the Before Startup Settings

In order to verify these *Before Startup* functions, generate initial speed to the motor. For example, make the motor spin and control the speed command to zero, before the motor stops send the startup speed command (non-zero) again. Before the motor stops, if we change the DIR pin and send the startup speed command, we can verify the *Reverse Drive* function and the *Brake* function.

10 AVS

The last step is to enable the AVS function. The AVS function prevents the voltage from surging when motor is spinning from a high speed to low speed (Mechanical AVS) and when the motor transfers from a driving state to coasting state (Inductive AVS).

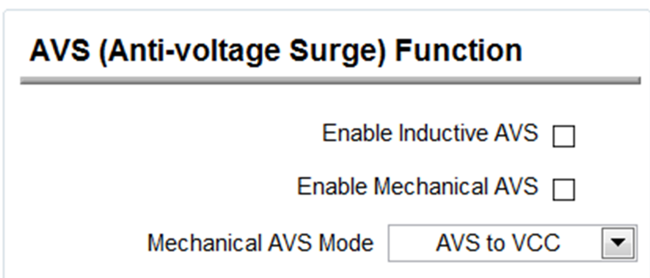


Figure 58. AVS Function Settings

There is a possibility that when the mechanical AVS is enabled, the motor is not able to decelerate. In this case the *Speed Cmd Buffer* is always higher than the *Speed Command*.

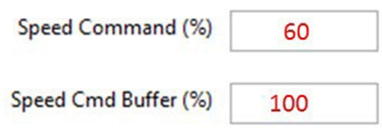


Figure 59. Speed Command Input

This is because:

- The programmed Kt value is wrong, it is too high. The solution is reducing the Kt value and checking whether it can be improved. If we see the difference between *Speed Command* and *Speed Cmd Buffer* getting closer, continue to reduce the Kt, until they are the same.
- The motor is operating with very low current, the Kt we have programmed is slightly higher than the correct value, but if we reduce the value by even one step, we can’t spin the motor at high speed. In this case, increase the software current limit threshold (or even disable the software current limit) or disable the Mechanical AVS.

If Mechanical AVS is disabled, the *Speed Command* should be controlled slow enough to prevent the VCC from pumping up. Please set the *Mechanical AVS Mode* to *AVS to VCC* if the voltage surging higher than VCC is not allowed in the application.

If VCC is less than 24 V, and the system allows VCC to go up, select "AVS to 24V", which means we protect the VCC only when VCC goes up to 24 V. Do not allow VCC to go even higher because it may damage the device.

Note that motor decelerating is slower by enabling the mechanical AVS.

11 Lock Detect

Put external torque on the motor and stop it. The device should be able to detect the lock condition and report the issue.

When the motor is blocked, the appropriate register bit is set and the corresponding GUI indicator is turned on. A description of the fault appears in the text box below the fault code indicators. Click the *Clear Fault* button to clear the fault.

Figure 60. Fault Code Indicator for Blocked Motor

If the device is not able to effectively detect the lock condition and continues driving output current after the motor is completely blocked, enable the *BEMF Abnormal* lock detect function.

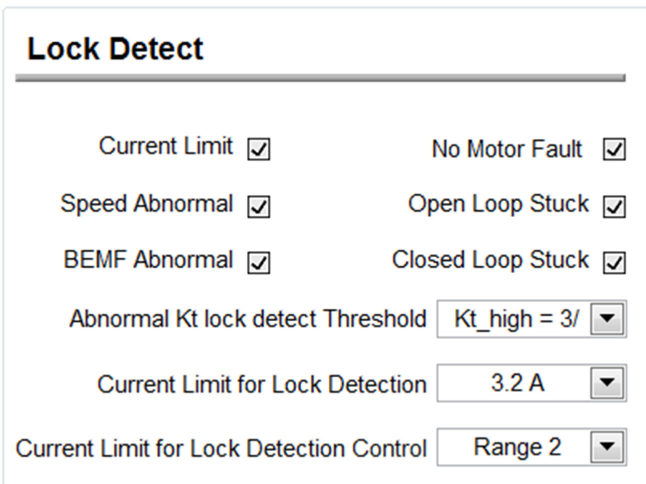


Figure 61. Lock Detect Settings

Sometimes enabling *BEMF Abnormal* could cause startup failure during transition from open loop to closed loop.

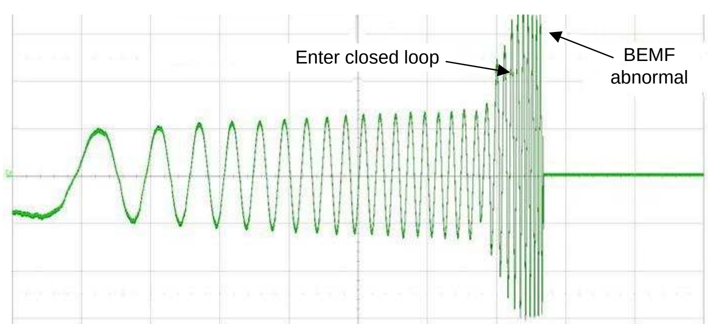


Figure 62. Startup Failure Using BEMF Abnormal

This is because during the transition, the calculated Kt is not accurate. Once the non-accurate Kt value lasts for longer than 300 ms, the device will treat it as a lock condition when actually the motor is accelerating normally. In this condition, two solutions are suggested:

- Increase the open to closed loop threshold, so that the transition happens at a higher speed. The device will pass the non-accurate Kt measurement period within less than 300 ms.
- Changing the closed loop AdjMode to *Half cycle adjustment* can also help solve this issue.

After exercising the preceding configuration, if there is not a combination which can start up the motor successfully under normal conditions and can reliably detect the condition when the motor is locked, an extra MCU may be needed to detect the lock, consult a TI representative for further advice.

Miscellaneous Motor Parameter Measurements

A.1 Number of Poles

The number of poles means the number of poles of the permanent magnet. The number of pole pairs is the number of poles divided by two.

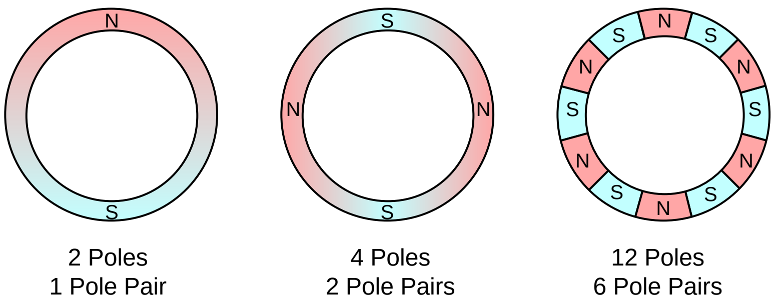


Figure 63. Pole Pairs in Permanent Magnets

If the number of poles is not listed in the motor specification, measure it with the following method:

Step 1. Use a lab power supply and inject current from phase U to phase V. Make sure the current amplitude is less than the motor rated current.

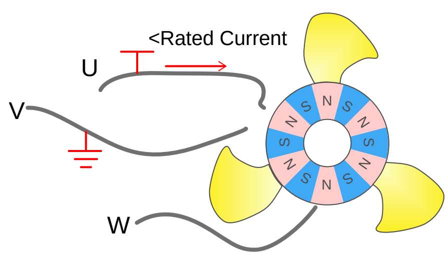


Figure 64. Injecting Current from Phase U to Phase V

Step 2. The rotor should have settled at one position with the injecting current. Manually rotate the rotor; it will have several *settle-down* positions around one mechanical cycle.

Step 3. Count the number of settle-down positions, which is the number of pole pairs. Multiplying by two calculates the number of poles (In Figure 65, there are six pole pairs in the motor).

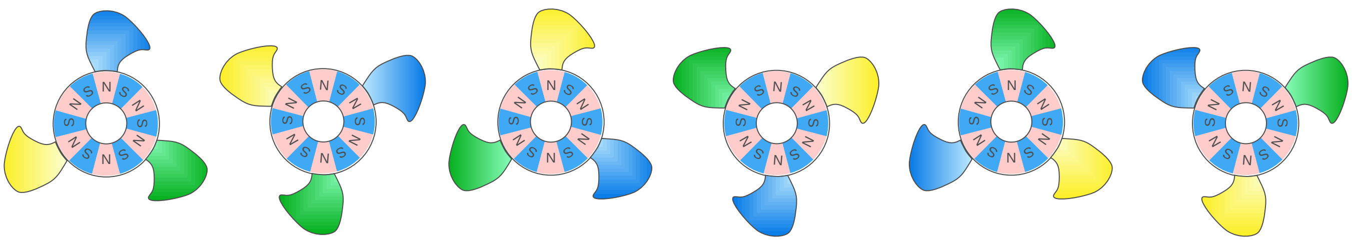


Figure 65. Multiple Motors

If unable to determine the pole pairs, leave this empty and spin the motor.

A.2 Motor Inertia

Motor inertia is the tendency of the motor to keep moving at the existing constant velocity, or to keep still. Motor inertia depends on the rotor structure, including the blades. Normally, motors with heavier blades or larger blades have greater inertia.

The unit of inertia is kg × m × m, but motor manufacturers usually do not provide this information. It is very difficult and not always necessary to find out the accurate inertia value. Instead, use the oscillation period value (T) to characterize the motor inertia (Large inertia motors have longer oscillation periods). The motor inertia affects the align time. Greater inertia means a longer align time.

The following steps can be used to measure the oscillation of the motor:

Step 1. Connect the oscilloscope to capture the voltage between V and W.

Step 2. Make sure the blades are assembled. Use a lab power supply with current limit function and inject 1 A of current into phase U and return from phase V of the motor. If the rated current of the motor is less than 1 A, inject the rated current and record a note in the table.

Step 3. The motor should have settled at one position.

Step 4. Manually rotate the fan blades off the settled position, then release. The motor will oscillate back and forth around the settle point.

Step 5. Measure the period of the swing using an oscilloscope.

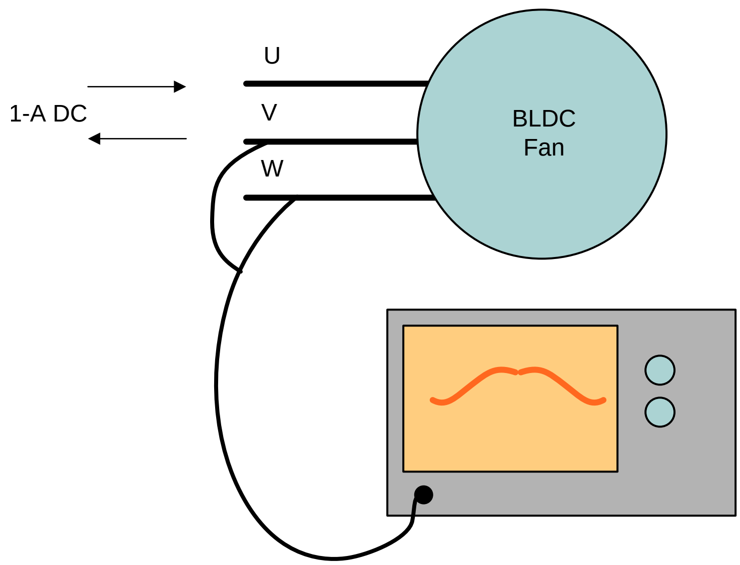


Figure 66. Oscilloscope Setup

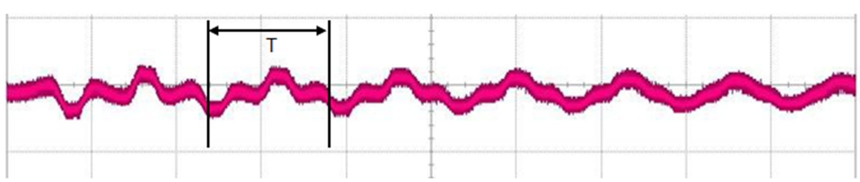


Figure 67. Oscilloscope Reading

IMPORTANT NOTICE FOR TI DESIGN INFORMATION AND RESOURCES

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265