

GUI Quick Start Guide: *InstaSPIN_FOC_F2806xM*

Version 1.0.4

Motor Solutions

Overview

InstaSPIN_FOC_F2806xM is a GUI Composer developed GUI that allows you to instrument and interact with the supplied Code Composer Studio (CCS) developed target binary to evaluate InstaSPIN™-FOC.

Supports:

- TMDSCNCD28069MISO Piccolo F28069M (ROM) controlCARD paired with
- 3-phase Inverter
 - o Low Voltage / Low Current: DRV8312_REVD
 - PN: DRV8312-69M-KIT
 - o Low Voltage / High Current: DRV8301_REVD
 - PN: DRV8301-69M-KIT
 - Do not try to use with BOOSTXL-DRV8301
 - o High Voltage: hvmtrkit_r1p1 [Yellow PCB with black enclosure]
 - PN: TMDSHVMTRINSPIIN
 - Do **NOT** use TMDX or TMDSHVMTRKIT5X [Yellow PCB with light grey enclosure] with CNCD28069MISO

----- NOTICE -----

The C code project (CCS developed target binary) that runs on these kits is **FROZEN**. It is at a fixed PWM & control frequency and fixed user scalings which are **NOT** appropriate for all motors. It also does not include the **CRITICAL** updates we have made to the solution since introduction in early 2013. You **MUST** use MotorWare projects to get these important features.

This GUI should only be used for general demonstration. Any performance testing should be done using the MotorWare based projects with careful consideration paid to setting up hardware and software scaling for proper motor ID and best performance.

----- NOTICE -----

TI Spins Motors



Version: 1.0.4

Revision History:

1.0.4	April 23, 2014	Updates regarding Warnings, Scalings, and direction for using MotorWare projects instead of the GUI.
1.0.3	May 9, 2013	Updated appProgram.out to support ADC Clock fix from MotorWare 1_01_00_07 Changed name of GUI installer that supports both the –FOC and –MOTION guis to GUI_Composer_InstaSPINs_F2806xM_install.exe Latest GUI installer includes patch to support PCs with multiple cores
1.0.2	March 21, 2013	Added Revision History
1.0.1	February 26, 2013	First release

TI Spins Motors

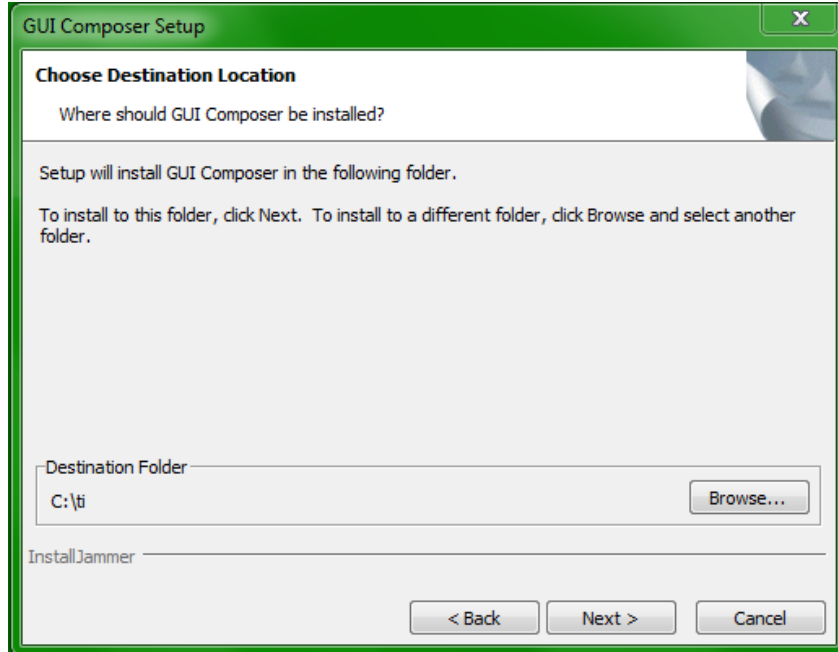


Installation

1. If this is the first time to install a GUI composer project, Run GUI_Composer_InstaSPINs_F2806xM_install.exe
 - a. Accept the license agreement



- b. Recommended to keep default destination location



2. Once the installer has been run, for any future updates to the GUI (available through [MotorWare](http://www.ti.com/tool/motorkitscncd69miso) or at the kit contents download <http://www.ti.com/tool/motorkitscncd69miso>) you may simply copy the updated folder "InstaSPIN_FOC_F2806xM" to C:\ti\guicomposer\webapps

TI Spins Motors



Hardware Set-up

1. All new kits should arrive with proper settings for all jumpers and switches
2. Please review the QSGs and hardware documentation – available through [MotorWare](#) - for details
3. Install F28069M controlCARD into controlCARD slot
4. Check of Critical Board Settings
 - a. TMDSCNCD28069MISO
 - i. SW1: UP-UP
 - ii. SW2: DOWN-DOWN
 - iii. SW3: UP-UP
 - b. DRV8312_REVD
 - i. JP1: VR1 - Middle
 - ii. JP2: Populated
 - iii. M1: High-Middle
 - iv. All three toggle switches in the middle (MCU) position
 - v. Motor Phase Wires to MOA, MOB, MOC
 - vi. DC Voltage to J9 or PVDDIN
 - c. DRV8301_REVD
 - i. JP2: Populated
 - ii. Motor Phase Wires to OUTA, OUTB, OUTC
 - iii. DC Voltage to PVDD and GND
 - d. HVMTR_r1p1
 - i. Populate: J3, J4, J5, J8,
 - ii. Populate: M3-J5
 1. (turns OFF on-kit emulation so you can use controlCARD emulation)
 - iii. J7: Right-Middle (towards capacitors)
 - iv. Motor Phase Wires to U, V, W
 - v. J2
 1. For DC bus input (recommended): Do NOT Populate J2 and
 - a. connect 50-350V DC supply from BS6 to BS5
 2. For AC input: Populate J2 Bridge-Middle

TI Spins Motors



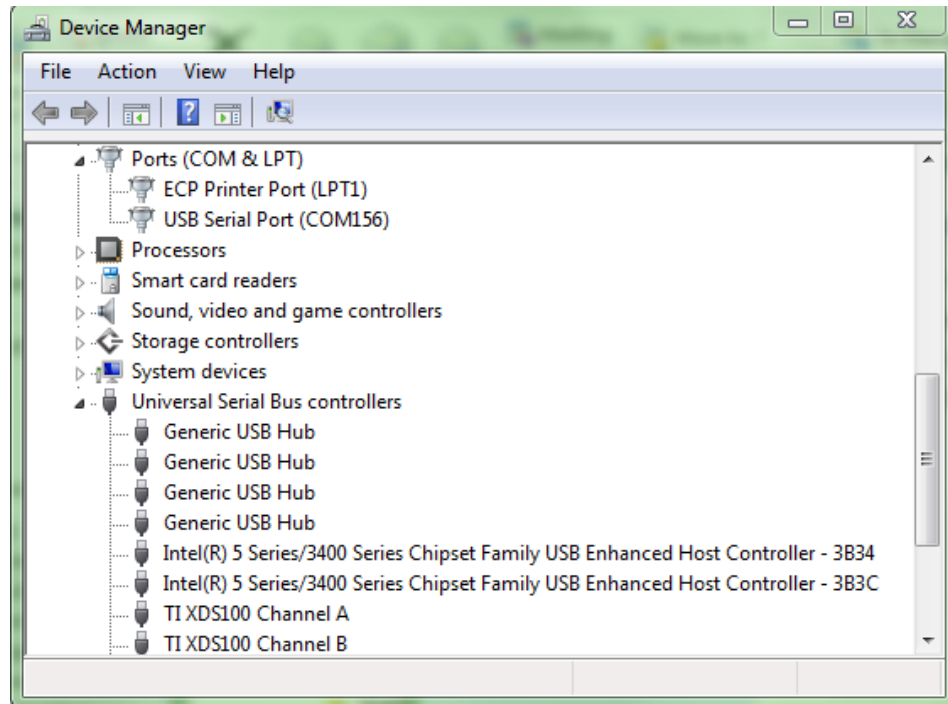
- a. connect banana cable from BS5 to BS1
- b. connect AC power cord to P1
- 3. DO NOT APPLY HIGH VOLTAGE AC OR DC UNTIL ALL CONNECTIONS ARE CHECKED, VERIFIED, AND YOU HAVE LAUNCHED THE GUI
- vi. 15V DC power supply to JP1 and SW1 ON

TI Spins Motors



Launch GUI

1. With low voltage DC powered, connect USB from your PC to J1 on controlCARD
 - a. You can verify that you connected to the FTDI XDS100v2 emulator by checking your Windows Device Manager for Ports: USB Serial Port (COMxxx) and TI XDS100 Channel A and B



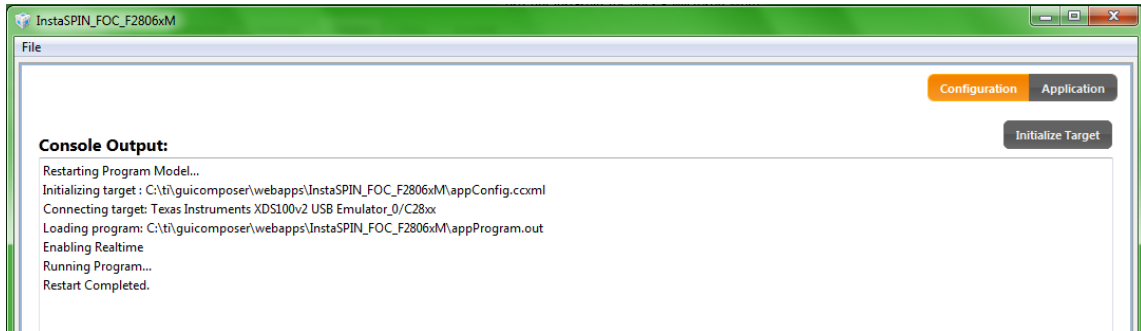
2. Run:
C:\ti\guicomposer\webapps\InstaSPIN_FOC_F2806xM\InstaSPIN_FOC_F2806xM.exe

- a. The GUI Composer application will start

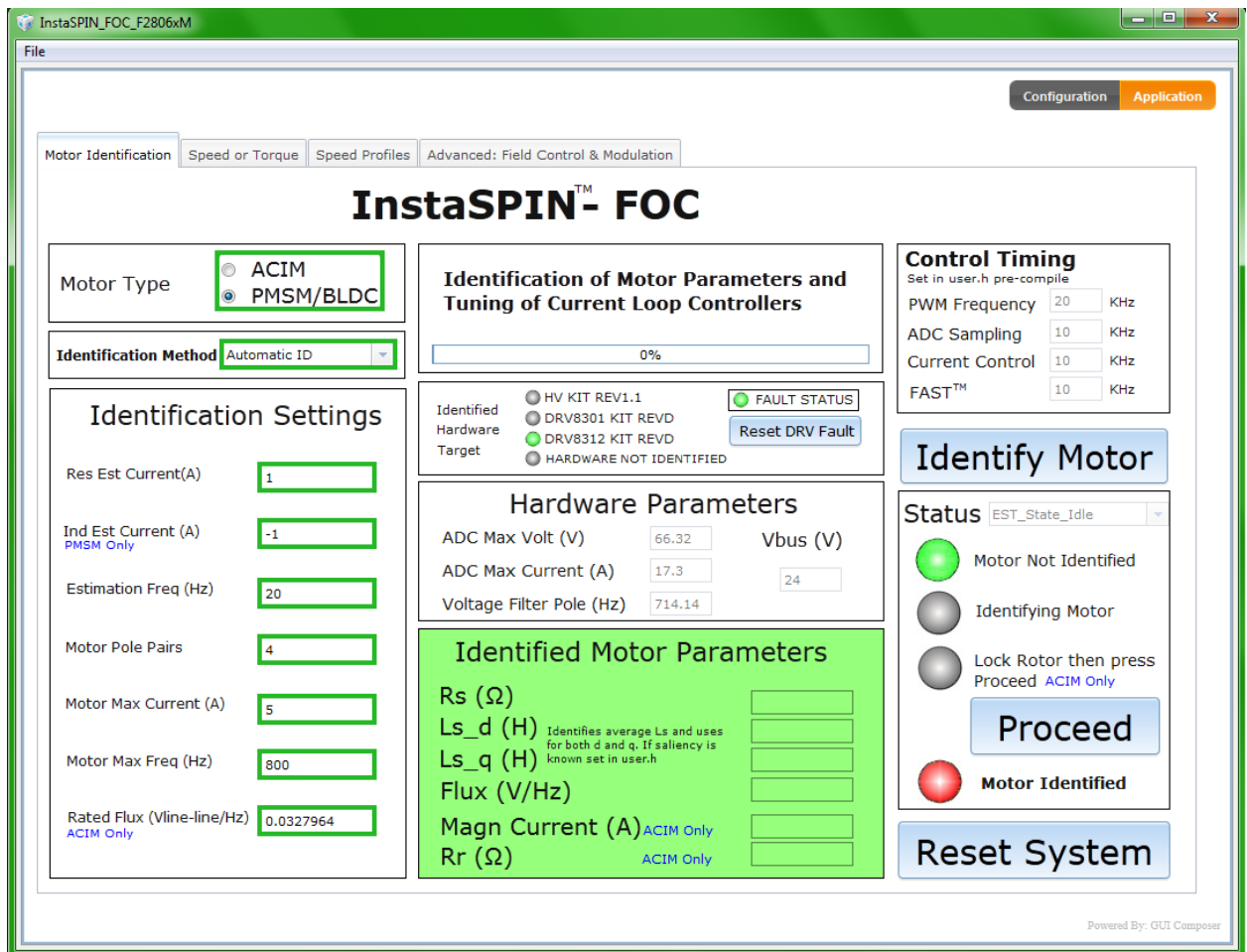


- b. GUI Composer will initialize, connect, then load \appProgram.out into the memory of the Piccolo F28069M.

TI Spins Motors



Normal Console Output of a successful GUI Launch



Start-Up Tab: Motor Identification

- c. If using the high voltage kit you can now energize the high voltage AC input (110-220Vac) or high voltage DC Bus (50-350Vdc)

TI Spins Motors



Tab1: Motor Identification

The screenshot shows the InstaSPIN-FOC GUI with the Motor Identification tab selected. The interface is divided into several sections:

- Motor Type:** Radio buttons for ACIM, PMSM/BLDC, and BLDC.
- Identification Method:** A dropdown menu set to Automatic ID.
- Identification Settings:** Input fields for Res Est Current(A) (1), Ind Est Current (A) (-1), Estimation Freq (Hz) (20), Motor Pole Pairs (4), Motor Max Current (A) (5), Motor Max Freq (Hz) (800), and Rated Flux (Vline-line/Hz) (0.0327964).
- Identification of Motor Parameters and Tuning of Current Loop Controllers:** A progress bar showing 0% completion.
- Hardware Parameters:** Input fields for ADC Max Volt (V) (66.32), ADC Max Current (A) (17.3), Voltage Filter Pole (Hz) (714.14), and Vbus (V) (24).
- Identified Motor Parameters:** Input fields for Rs (Ω), Ls_d (H), Ls_q (H), Flux (V/Hz), Magn Current (A), and Rr (Ω).
- Control Timing:** Input fields for PWM Frequency (20 KHz), ADC Sampling (10 KHz), Current Control (10 KHz), and FAST (10 KHz).
- Status:** A dropdown menu set to EST_State_Idle, with indicators for Motor Not Identified, Identifying Motor, Lock Rotor then press Proceed ACIM Only, and Motor Identified.
- Buttons:** Identify Motor, Proceed, and Reset System.

Identified Hardware Target

This section shows the hardware target selection interface. It includes radio buttons for different hardware kits and a 'Reset DRV Fault' button.

- Only proceed if the GUI has properly identified the target hardware you are using
- Fault Status
 - The fault status light will be GREEN if NO FAULTS are present only on the DRV8312 or DRV8301
 - The fault status light will only turn RED if a FAULT is present on the DRV8301
 - Reset DRV Fault works only for DRV8301

TI Spins Motors



- DRV8312 includes a fault LED (/FAULT) on the hardware, and must be cleared using the toggle switches (move RSTA from MCU to RESET to MCU).

Hardware Parameters

- Based on the identified hardware this displays the hardware settings in the user.h file used in the compiled program
 - ADC Max Voltage, scaled to be seen as 3.3V input of the ADC
 - Note that FAST observer performance – especially low speed – is dependent on resolution of the measured phase voltage signals. In your own design you will find better performance by designing your hardware to exactly match the expected range of voltage measurements (based on the larger of the bus voltage or EMF of the motor at maximum expected frequency).
 - ADC Max Current, scaled to be seen as 3.3V input of the ADC
 - Note that InstaSPIN-FOC current control performance – especially low load – is dependent on resolution of the measured shunt current signals. In your own design you will find better performance by designing your hardware to exactly match the expected range of current measurements (based on the current/torque of the motor at maximum expected load).
 - Voltage Filter Pole, hardware RC filter on phase voltage readings, matched in user.h for corresponding software filter. Critical to performance of the FAST observer.
 - Vbus: Real voltage reading of the DC bus of the inverter
 - Insure that you are providing enough bus voltage for proper performance evaluation.
 - Especially with the High Voltage kit, most applicable motors are specified for 280-350Vdc. If AC entry is used the rectification from a 110Vac line is too low (ex: 165Vdc) to meet speed and torque expectations.

Control Timing

Control Timing		
Set in user.h pre-compile		
PWM Frequency	<input type="text" value="20"/>	KHz
ADC Sampling	<input type="text" value="10"/>	KHz
Current Control	<input type="text" value="10"/>	KHz
FAST™	<input type="text" value="10"/>	KHz

- The control timing is fixed in the user.h file used to compile this FROZEN project
 - **For some motors the PWM, current, or FAST estimation rates are too low for proper control. You MUST use the MotorWare projects and update the user.h file accordingly.**

TI Spins Motors



- For the GUI these settings are chosen to work across all hardware platforms and most of the target applications for this technology. **It is not optimized for each application and there will be motor types which require changes.**
 - PWM Frequency: 20 KHz default
 - Typically want to use the lowest possible to reduce switching losses but 8 – 20 KHz is the most typical range
 - **Lower inductance (often higher current) motors typically require higher PWM frequencies (30 – 100 KHz) to effectively control switching**
 - ADC Sampling: 10 KHz default
 - ADC sampling should be synchronized to the higher of the Current Control (typical) or FAST rate.
 - In higher PWM applications where the current control rate will run slower anyways (to allow the control system to run on < 100 MHz MCUs) the ADC can be decimated by using the Piccolo ePWM ADC SOC hardware (recommended) through the user.h software decimation.
 - The ADC conversions done signal starts the main control interrupt to run the sensorless FOC control system
 - Current Control: 10 KHz default
 - Synchronized to the ADC in this example
 - In extremely high dynamics (speed or load changes), extremely high speeds, or when controlling low inductance motors with large current ripple you want to run as often as possible (15-20 KHz) for maximum performance
 - In lower dynamic applications 5 KHz is often acceptable
 - FAST: 10 KHz default
 - Synchronized to the Current frequency in this example,
 - For extremely high dynamics (speed or load changes) or extremely high speeds you would want to run as often as possible (15-20 KHz) for maximum performance
 - In lower dynamic applications 2.5 KHz or 5 KHz is often acceptable
 - The Speed and Trajectory Controller is set to be run at 1 KHz, but can be modified as desired in user.h
 - Excellent results have been seen by running the speed controller at the FAST rate due to the accuracy of the speed estimation from FAST

TI Spins Motors



Motor Type

Motor Type

☐ ACIM

☒ PMSM/BLDC

- Select the three phase motor you wish to identify
 - Asynchronous (AC Induction)
 - Synchronous (permanent magnet or brushless DC)
 - Interior Permanent Magnet (IPM) salient motors are also supported by the InstaSPIN-FOC solution, but with a caveat on the motor identification
 - During identification the motor is treated as a non-salient motor and an average stator inductance is identified.
 - To maximize efficiency of salient (IPM) motors
 - For most accurate operation of the FAST observer
 - Ls-d and Ls-q must be user provided in user.h
 - Ls-d and Ls-q will change across motor loading, so in a full system a look-up table technique to load new Ls values to FAST should be considered
 - For most efficient FOC operation
 - A special maximum torque per amp algorithm may need to be considered
 - This will adjust the torque command output of the speed controller to the Iq and Id reference inputs of the InstaSPIN-FOC torque controller
 - An example will be provided in the future

Identification Method

Identification Method

Automatic ID

Automatic ID

User.h Settings

- Automatic ID
 - Select "Automatic ID" to use the Motor Identification portion of InstaSPIN-FOC to identify the required motor parameter of your motor
- User.h Settings

TI Spins Motors



- Select “User.h Settings” to bypass the Identification process and load the values and Identification Settings stored under USER_MOTOR in the user.h file that was used during project compile.

- **Re-compiling of this project will NOT be offered**
- **You must move to the MotorWare projects to save your USER_MOTOR settings in user.h**

- The values stored are for the DRV8312 EVM and the included 24V Anaheim/Telco BLDC motor only
- Default settings from user.h (for Anaheim / Telco NEMA17 24V BLDC)

```
#define Anaheim_BLY172S      102

#define USER_MOTOR Anaheim_BLY172S

#elif (USER_MOTOR == Anaheim_BLY172S)

#define USER_MOTOR_TYPE      MOTOR_Type_Pm

#define USER_MOTOR_NUM_POLE_PAIRS    (4)

#define USER_MOTOR_Rr          (NULL)

#define USER_MOTOR_Rs          (0.4110000000)

#define USER_MOTOR_Ls_d        (0.0007092811)

#define USER_MOTOR_Ls_q        (0.0007092811)

#define USER_MOTOR_RATED_FLUX    (0.0327964)

#define USER_MOTOR_MAGNETIZING_CURRENT (NULL)

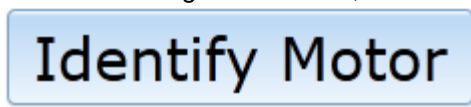
#define USER_MOTOR_RES_EST_CURRENT    (1.0)

#define USER_MOTOR_IND_EST_CURRENT    (-1.0)

#define USER_MOTOR_MAX_CURRENT        (5.0)

#define USER_MOTOR_FLUX_EST_FREQ_Hz    (20.0)
```

- The **BLACK** settings from user.h are displayed in the GUI under Identification Settings
 - The **RED** settings from user.h are what are identified (for synchronous motors)
- If “User.h Settings” is selected, after clicking



the **RED** values from user.h will be pulled and displayed as the Identified Motor Parameters

TI Spins Motors

Identification Settings

Res Est Current(A)	<input type="text" value="1"/>
Ind Est Current (A) <small>PMSM Only</small>	<input type="text" value="-1"/>
Estimation Freq (Hz)	<input type="text" value="20"/>
Motor Pole Pairs	<input type="text" value="4"/>
Motor Max Current (A)	<input type="text" value="5"/>
Motor Max Freq (Hz)	<input type="text" value="800"/>
Rated Flux (Vline-line/Hz) <small>ACIM Only</small>	<input type="text" value="0.0327964"/>

Identified Motor Parameters

Rs (Ω)	<input type="text" value="0.411"/>
Ls_d (H)	<input type="text" value="0.000709281"/>
Ls_q (H)	<input type="text" value="0.000709281"/>
Flux (V/Hz)	<input type="text" value="0.0327964"/>
Magn Current (A) <small>ACIM Only</small>	<input type="text"/>
Rr (Ω) <small>ACIM Only</small>	<input type="text"/>

- And Motor Identified will turn green



Motor Identified

- Note that when loading from user.h it will also load the hardware current and voltage offsets. Ex:

```
#define USER_DRV8312_KIT_REVD_I_A_offset (0.7386139631)
```

```
#define USER_DRV8312_KIT_REVD_I_B_offset (0.7363774185)
```

TI Spins Motors



```
#define USER_DRV8312_KIT_REVD_I_C_offset    (0.7402219176)
#define USER_DRV8312_KIT_REVD_V_A_offset    (0.1301432252)
#define USER_DRV8312_KIT_REVD_V_B_offset    (0.1301434636)
#define USER_DRV8312_KIT_REVD_V_C_offset    (0.1303436756)
```

Identification Settings

With this FROZEN GUI project:

Most Induction Motors will NOT identify correctly, use MotorWare proj_lab02a or 2b for Motor ID.

Most high speed, low Ls, or low Flux motors will NOT identify correctly, use MotorWare proj_lab02c for Motor ID.

The **IQ_FULL_SCALE_VOLTAGE** and **CURRENT** variables are not made accessible through the GUI - only adjustable through the user.h file and recompilation - but it is important to understand their impact on the system.

	IQ_FULL_SCALE _VOLTAGE_V (0-MAX)	ADC_FULL_SCALE _VOLTAGE_V (V) (peak-peak)	IQ_FULL_SCALE _CURRENT_A (0-MAX)	ADC_FULL_SCALE _CURRENT_A (peak-peak)
DRV8312	48.0	66.32	10.0	17.3
DRV8301	48.0	66.32	40.0	82.5
HVMTR	450.0	409.9	10.0	19.89

Unlike the ADC_FULL_SCALE values, which are defined by the specific hardware sensing circuit, the IQ_FULL_SCALE values are adjustable on any given project and are used to normalize all of the voltage/current terms to a per unit value (pu).

- For the current this is easy to define as you will never have a current variable value higher than what the hardware can measure.
 - Note however that the resolution is impacted as you attempt to cover a broader range of current. This lower resolution for the DRV8301 will affect Motor ID and run-time performance when used with a low current motor.
- For voltage you must also consider the B_{emf} of the motor and the calculations occurring inside the algorithms. These voltages inside the motor can be greater than the input voltage itself in cases where the motor is operated in field weakening or where the motor is operated beyond its rated speed.
 - When selecting this value to be used for Motor ID you would like to maximize your resolution, so choosing a low enough value is critical.

TI Spins Motors



- When selecting this value for full operational run-time you MUST insure that the IQ_VOLTAGE value is greater than any internal voltages of the motor (back EMF). If not, a variable overflow will occur, leading to unexpected results from the estimator.
- For this GUI project values were selected for good resolution during Motor ID and normal operation and evaluation of most motors.
 - **Due to this, large Bemf motors may NOT perform as expected with this GUI project at highest speeds and field weakening regions. This value must be adjusted in those cases. See the User's Guide for more information and use MotorWare projects.**

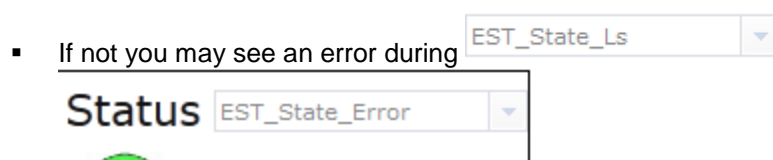
Res Est Current: Constant current used during resistance testing (positive number). This value should typically be ~10% maximum rated current – or current needed to produced rated torque - of the motor. It must be large enough (may be >10%) to insure that the motor starts spinning during the Estimation State RampUp though. Avoid too high of a current which will cause overheating of the motor during identification.

Ind Est Current: Constant current (negative) used during inductance testing.

- **PMSM:** This value should be ~10% (negative) of the rated current. In general this value is the negative value of the current used for resistance estimation.
- **ACIM:** When identifying ACIM motors, this inductance estimation current is ignored and not used. Set to 0.

Estimation Freq: Maximum target frequency used during estimation, ~10% of rated speed.

- **PMSM:** 20Hz typical works for most motors. Exceptions
 - if the motor's inductance is known to be in the tens of micro Henry (μH), this frequency should be increased to 40 Hz, 60 Hz, or even as high as 150Hz until a reliable value is obtained.
 - When using the DRV8301 kit with any lower voltage/current motors, 20 Hz may not provide an accurate value for the inductance.



or values that are obviously too low

2.1e-9

- This is due to the resolution of the feedback as scaled by the board's hardware ADC scaling and the software IQ scaling.
- Use as much voltage as possible and increase the Estimation Frequency, starting at 40 Hz until reliable results are obtained

TI Spins Motors



- **Use MotorWare proj_lab02c for high speed, low Ls,, or Low flux motors with attention paid to IQ_VOLTAGE scaling**
- **ACIM:** Estimation frequency should always be 5Hz for 50/60 Hz motors.
 - **It is recommended to use MotorWare for all induction motors as Motor ID was significantly improved starting with MotorWare 1_01_00_12**

Motor Pole Pairs: Used to calculate

- User friendly RPM from Electrical Frequency ($\text{RPM} = \text{Hz} * 120 / \text{poles}$)
- Accurate estimation of the shaft torque in N.m.

Motor Max Current: This is the maximum value allowed as the output of the Speed Controller (input to the Iq PI controller); it is NOT a safety setting for total current through the motor and should be set no higher than the maximum current rating of the motor.

- This value is also directly used for calculation of the initial Speed PI control Kp and Ki gain values as a “rule of thumb” that generally provides a stable – though not tuned – speed control output to a torque control system. Double the Motor Max Current and you will notice that the speed Kp and Ki also double.

Motor Max Freq: Used for per unit scaling of all frequencies in the system, this should be slightly higher than the highest electrical frequency the motor will target. This is in electrical frequency, so if the absolute maximum speed of a motor is 10,000 RPM on a 2 pole pair motor, this frequency should be $(10,000/120)*4 = 333.33\text{Hz}$. Some headroom is recommended, so in this case a Motor Max Frequency of 500Hz could be used

- **Even for low frequency motors it is recommended to keep the max frequency at a minimum of 500 Hz. Larger values do not negatively affect the control system.**
- Motor Max Freq must be $< 4 * \text{the Hardware Voltage Filter Pole}$ (with some buffer)

DRV8301_revD	335.648 Hz	Motor Max Freq Max = 1200 Hz
DRV8312_revD	714.14 Hz	Motor Max Freq Max = 2500 Hz
HVMTR_rev1p1	372.5 Hz	Motor Max Freq Max = 1300 Hz

- Note that the Motor Max Frequency also subsequently affects another portion of the system, the speed at which the ForceAngle feature switches off and FAST estimations are used.
- The frequency of this switching point is set by Motor Max Frequency multiplied by USER_ZEROSPEEDLIMIT (from user.h). In this GUI project USER_ZEROSPEEDLIMIT is set to 0.002 and cannot be changed unless you re-compile. Note how this effects the lowest operating speed at which FAST estimations will be used as you change the Max Frequency

Motor Max Freq (Hz)	Switch from ForceAngle to. FAST (Hz)
200 Hz	0.4 Hz
500 Hz	1.0 Hz

TI Spins Motors



2000 Hz	4.0 Hz
---------	--------

- For the default settings that load from user.h, the ForceAngle to FAST switch frequency = $800 * 0.002 = 1.6 \text{ Hz}$; $1.6 \text{ Hz} * 120 / 8 \text{ poles} = 24 \text{ RPM}$. FAST will be active over +/- 24 RPM, and inside this range the ForceAngle will be active unless disabled.
- **Note that the ForceAngle feature is used during motor identification, so this effective frequency of switching from ForceAngle to FAST must < than Estimation Freq or FAST will not be activated, hence it will not provide the necessary feedback for proper ID.**
- While FAST is capable of tracking at <1 Hz speeds, the lowest frequency for any system is completely dependent on the hardware and software scaling, quality of sensing, and $B_{emf} (\text{flux} * \text{speed_Hz})$ of the motor. You would also need to insure that the ForceAngle was set below the speed you want to run or is disabled during run-time. See the User's Guide further details.
 - For demonstrations on the TI motor kits at normal operating voltages 1-5 Hz for FAST is a reasonable expectation.
 - While you can't directly change this ForceAngle setting for a given Max Frequency through this GUI interface, keep this in mind as you are evaluating low speed performance
- It should be noted that motors that are designed to run at over 1.5 KHz may not be fully supported by InstaSPIN-FOC at this time. Please see the User's Guide for discussion on this topic and use MotorWare proj_lab02c for proper identification.

Rated Flux:

- **ACIM:** This value should be: $\text{SQRT}(2/3) * \text{Rated Voltage from line to line} / \text{Rated Frequency}$. For example, a 220VAC/60Hz ACIM motor is to be identified, the value to be entered for Rated Flux is: 2.99.
- **PMSM:** This setting is ignored for PMSM motors as it is a value identified.


TI Spins Motors


Motor Identification


Identify Motor


Status

EST_State_Idle

 Motor Not Identified

 Identifying Motor

 Lock Rotor then press
Proceed **ACIM Only**

 **Motor Identified**

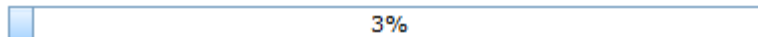
Proceed

Reset System





- The motor identification process, state machine, hardware scaling, choosing of PI control gains, and details of the tests being performed are documented in the User's Guide
 - While many synchronous motors will identify perfectly, there are known corner cases - especially for particular motor types and conditions -where the motor ID settings accessible through the GUI project will not allow for successful identification. Use MotorWare to adjust the scaling and use the updated proj_lab## for best ID.
 - It should be noted that a successful Motor ID is not necessary to use FAST or InstaSPIN-FOC. Motor parameters can be identified by hand measurement or through a motor datasheet and updated in the user.h file for compile. Use MotorWare projects.
- Click Identify Motor to begin the process
 - You will note that the Identifying Motor lamp illuminates, the Status moves to a new state, and the Percentage bar begins to move

TI Spins Motors

Identification of Motor Parameters and Tuning of Current Loop Controllers



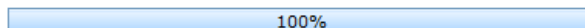
Status

-  Motor Not Identified
-  Identifying Motor
-  Lock Rotor then press
Proceed [ACIM Only](#)
-  **Motor Identified**

Proceed

- Status States
 - Defined in sw\modules\est\src\32b\est.h and thoroughly explained in the User's Guide
 - For ACI, the ID will pause at EST_State_LockRotor
 - At this point the Lock Rotor lamp will illuminate
 - Mechanically lock the rotor (using a wrench for small motors) for the duration of the testing and press Proceed
- Identification is completed when the display reads 100%, the Motor Identified lamp lights, and the parameters are displayed

Identification of Motor Parameters and Tuning of Current Loop Controllers



TI Spins Motors

Status

☐ Motor Not Identified

☐ Identifying Motor

☐ Lock Rotor then press
Proceed [ACIM Only](#)

☒ **Motor Identified**

- Example for Anaheim/Telco NEMA17 supplied with DRV8312 Kit
 - Note that your Anaheim/Telco motor may have different values, although the L_s and Flux values tend to be pretty consistent based on our testing. The R_s is typically 0.38 – 0.45 and is temperature dependent.

Identified Motor Parameters

R_s (Ω)		<input type="text" value="0.4228207"/>
L_{s_d} (H)	Identifies average L_s and uses for both d and q. If saliency is known set in user.h	<input type="text" value="0.0006741598"/>
L_{s_q} (H)		<input type="text" value="0.0006741598"/>
Flux (V/Hz)		<input type="text" value="0.0394098"/>
Magn Current (A)	ACIM Only	<input type="text"/>
R_r (Ω)	ACIM Only	<input type="text"/>

- Identification Issues
 - EST_State_Error
 - If this is displayed an error has occurred
 - Possible causes
 - Choosing ACIM with a PMSM/BLDC Motor or choosing PMSM/BLDC with an ACIM
 - Incorrect settings, particularly
 - Res Est Current (need large enough to spin during ramp-up)
 - Estimation Frequency (especially for high speed motors or when the scale of the motor signals are much lower than the scale of the hardware)

TI Spins Motors



- PWM Frequency (may need faster switching for very low inductance motors)
- Incorrect Rated Flux (for ACIM)
- To return to initial conditions click

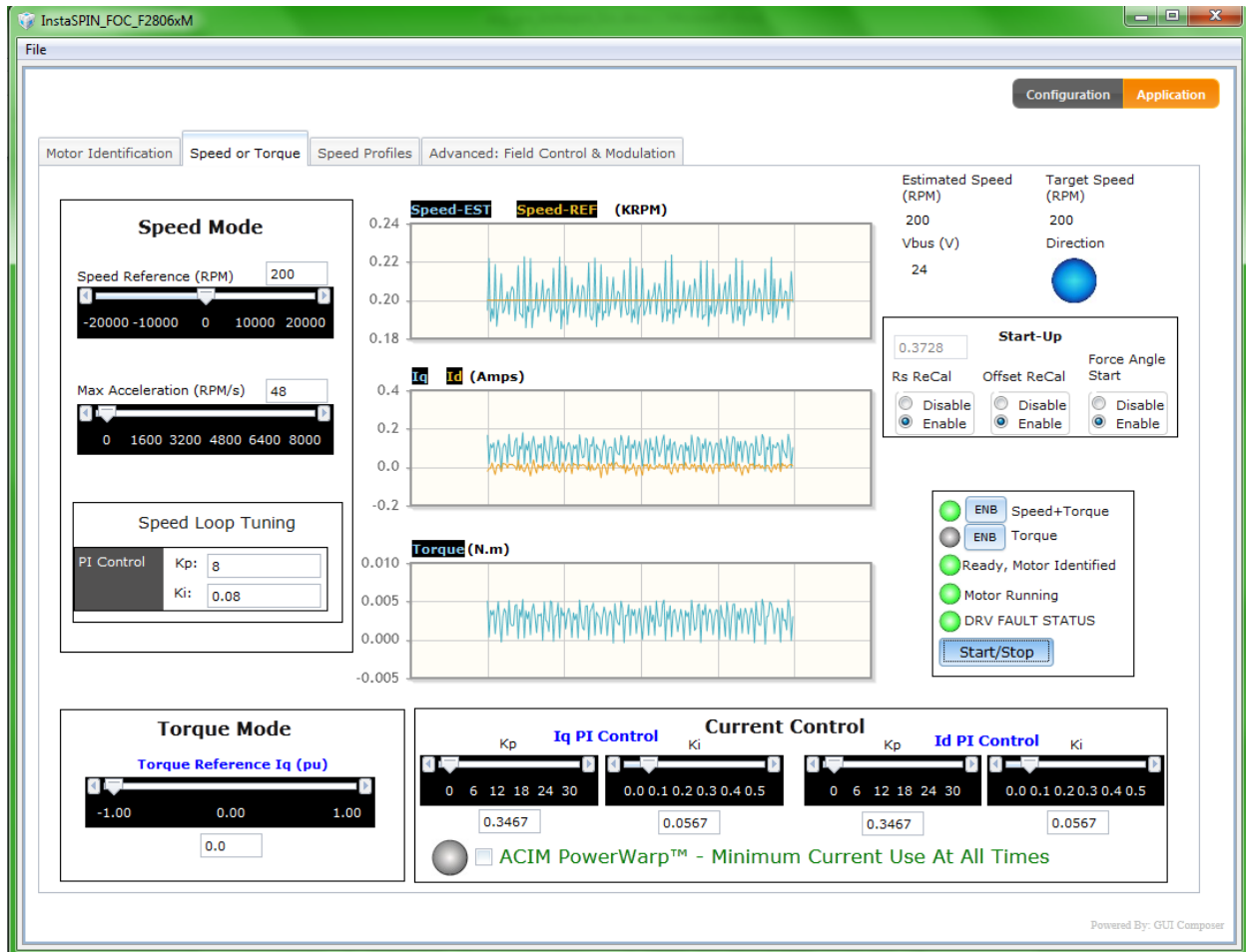
A blue rectangular button with rounded corners and a thin border, containing the text 'Reset System' in a bold, black, sans-serif font.
- If you reset the system during Identification or after an identification error the Est_State Status and the lamps will NOT reset until you next click Identify Motor.
- It is also possible that the motor identification completes without an Error but that the parameters identified are incorrect.
 - This occurs most frequently with very low inductance, high speed PMSM/BLDC motors (such as hobby motors used for helicopters and remote control cars)
 - During identification be sure to watch the motor. From EST_State_RampUp the motor should **begin spinning** up to the Estimation Freq (Hz) and should **continue until the identification process completes**.
 - If it does not spin, then the parameters identified are NOT correct.
 - If motor did not spin through completion, possible causes include
 - Res Est Current needs to be set higher to allow the motor to start-up
 - Estimation Frequency too low, increase in 20 KHz increments
 - PWM frequency too low to properly control current ripple of ultra low inductance motors (typically require 30-100 KHz)
 - Scaling of IQ vs. ADC vs. Motor values are incompatible
 - Will need to move from GUI project to MotorWare project and consult the User's Guide
 - For very low inductance motors the identified inductance L_s may not be precise due to the identification at too low of an Estimation Frequency. Try raising the Estimation Frequency to 40 Hz and 60 Hz and note if it changes significantly.
 - The smallest valid L_s is $\sim 1\text{e-}6$ H (1 μ H). Any lower should be considered an incorrect motor ID.

TI Spins Motors



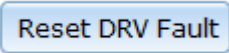

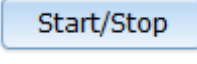


Tab2: Speed or Torque

The Speed or Torque tab is used for initial evaluation and tuning of the InstaSPIN solutions. Motors can be run in torque only mode (user sets or varies a torque input, controller maintains that constant torque) or speed + torque mode (user varies speed input, speed controller commands necessary torque across acceleration and load to keep speed)



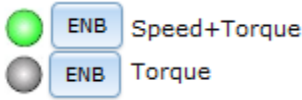
Status Lamps

- Ensure that  Ready, Motor Identified is GREEN, else return to Motor Identification Tab, RESET SYSTEM, and re-identify
- Ensure that  DRV FAULT STATUS is GREEN, else return to Motor Identification Tab to 
-  Motor Running will turn green once the drive is enabled using 

TI Spins Motors



Mode Options & Lamps



- Speed + Torque: Enables the ROM implementation of FAST used in an InstaSPIN-FOC torque controller with cascaded speed controller

- Speed Controller

Speed Loop Tuning	
PI Control	Kp: 8
	Ki: 0.04

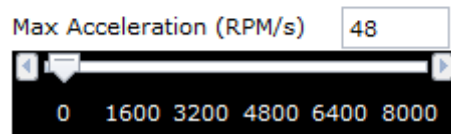
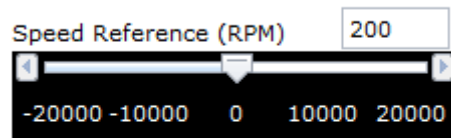
- Default Speed Controller is the PI included in ROM as part of InstaSPIN-FOC
 - The Kp and Ki gains are calculated from user.h settings
 - $Kp = 0.02 * \text{Motor Max Freq} * \text{Motor Max Current (pu)}$
 - $Ki = 2.00 * \text{Motor Max Freq} * \text{ctrlPeriod_sec} * \text{Motor Max Current (pu)}$
 - Do NOT consider the speed controller tuned. This is a simple “rule of thumb” calculation that generally allows stable operation over a wide range of motors
 - The default values provided by this “rule of thumb”
 - Are typically too low for larger inertia motors
 - Are typically too high for smaller inertia motors
 - User will need to tune the speed controller based on their system inertia and required dynamic performance, often across different speeds, motions, loads, etc.
 - As a general example, with the DRV8312 kit and motor
 - For very low speed operation with load, the gains need to be increased; try some values like 3x
 - For very high speed and into field weakening the gainst need to be decreased, try some values like /4
 - The User's Guide includes a speed control implementation section.
- The user input into the control system is a SpeedRef command in +/-RPM along with a maximum acceleration rate in RPM/s
 - max speed and acceleration (Hz, Hz/s) is limited in user.h and dependent on the motor, power supply, load, etc.
 - Higher speeds may be possible with your motor

TI Spins Motors

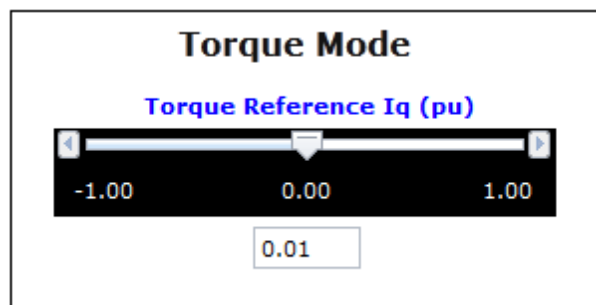


- The 20 KRPM and 8 KRPM/s noted in the GUI may NOT be possible with your motor

Speed Mode



- The output of the speed controller is an Iq torque command in per unit [-1.0 to 1.0], with 1.0 representing the Max Current setting on the first tab.
- Torque: Enables the ROM implementation of FAST used in an InstaSPIN-FOC Torque controller WITHOUT any speed controller capability
 - The input into the control system is a TorqueRef command in per unit [-1.0 to 1.0]

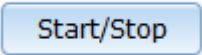
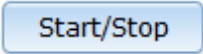


- **Caution: When in Torque mode you are commanding a constant torque. You should always apply a load to the motor or you risk even a small constant torque command creating a motor that quickly goes as fast as mechanically possible, which can create a fault and damage the inverter, motor, or user.**
- When in Torque Mode, even though the speed controller output is no longer an input to the control system a **non-zero, sign matching** target SpeedRef must still be entered for the direction and ForceAngle Start functionality. Ex:
 - Torque of + 0.05 pu, Speed Reference = +200
 - Torque of -0.05 pu, Speed Reference of -200

Start-Up Options





The image shows a software window titled "Start-Up". At the top left, there is a text box containing the value "0.3891". Below this, there are three columns of settings. The first column is labeled "Rs ReCal", the second "Offset ReCal", and the third "Force Angle Start". Each column has two radio buttons: "Disable" and "Enable". In all three columns, the "Enable" radio button is selected, indicated by a blue dot in the center of the button.

- Rs ReCal
 - Enables an ~8 second re-identification of the Stator Resistance (existing value shown directly above) each time the drive moves from disabled to enabled using 
 - Can be disabled if you just performed an ID
 - May be enabled periodically to view changes in Rs
 - Enable at least once if the motor parameters were loaded from the user.h file
 - See User's Guide for details of Rs On-Line (continuous run-time) re-estimation
- Offset ReCal
 - Enables an off-set calibration of the ADC channels when the drive is enabled using 
 - After the first time Offset ReCal runs (including during Motor ID), the values are stored for re-use, but it is recommended you enable when possible
 - If loading from user.h be sure your offset values are also stored. If not, enable Offset ReCal at least once
- Force Angle Start
 - Enables a full torque start-up from zero speed routine before the FAST based feedback is fed to the FOC torque system. The frequency at which FAST takes over is discussed in the Identification Settings – Motor Max Freq section.
 - Typically Enable when first starting from zero speed
 - Disable after start-up if you plan to move through zero speed
 - Details on the Force Angle feature and how to calculate and set acceleration and Force Angle to FAST transition points are covered in the User's Guide

TI Spins Motors

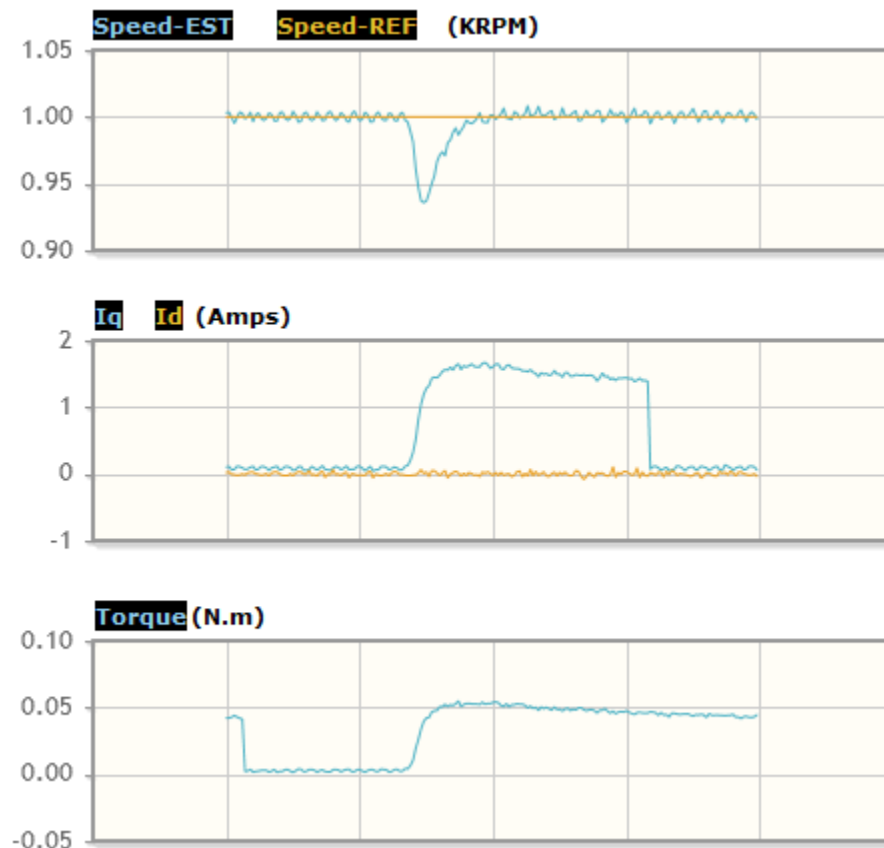


Display Panel

Estimated Speed (RPM)	Target Speed (RPM)	Estimated Speed (RPM)	Target Speed (RPM)
201	200	-199	-200
Vbus (V)	Direction	Vbus (V)	Direction
24		24	

- The Display Panel shows
 - Estimated Speed (RPM) from FAST
 - Target Speed (RPM) going to the speed controller (output of the motion profile)
 - Voltage Bus (V)
 - Direction: Blue (+) or Yellow (-)
 - Note, sometimes you will see the direction lamp erroneously change color. This is a bug in the GUI tool and not the direction variable that is coming from FAST.

Graphs

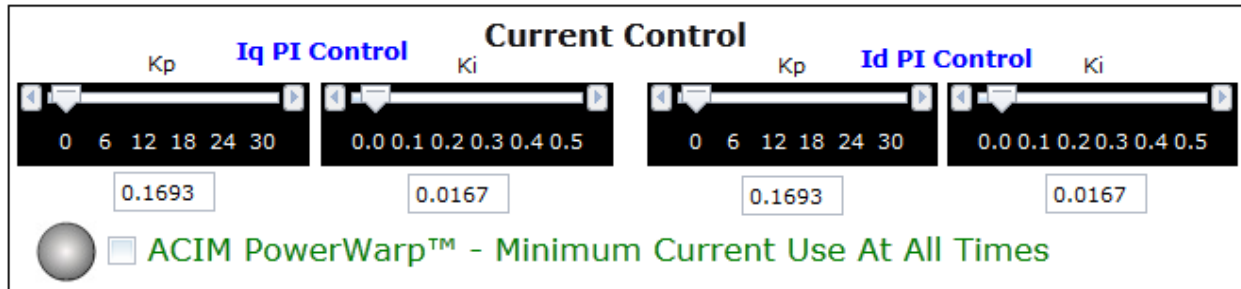


TI Spins Motors



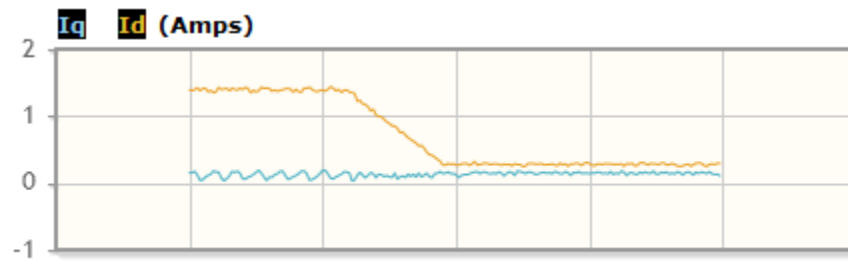
- The Graphs display real-time data on
 - Speed-EST vs. Speed-REF (KRPM)
 - Iq (Torque Component) vs. Id (normally 0) (Amps)
 - Shaft Torque (Newton Meters)
- The graphs automatically rescale to fit the dataset

Current Control



- The current (Iq and Id) PI controllers can be changed at any time
- The Kp and Ki settings are determined from the motor parameters and other scaling and settings in user.h. The formulas used are documented in the User's Guide.
- The values are set slightly soft to give good performance over a wide variety of hardware and motors. To stiffen the control, increase the values of Kp and Ki. At very high frequencies (speeds of 1 KHz+) stiffer control up to multiples of 4x the original values may be required.
 - This 4x is NOT required when using MotorWare projects that take advantage of the angle compensation algorithm.
- You may choose to run specific step response tests to ideally tune for the specific current dynamic performance you require
- ACIM [PowerWarp](#) is a mode for three phase induction (asynchronous) motors only. When enabled it will dynamically adjust the Id current to the lowest possible while still maintaining tracking of the rotor and good speed control, providing best efficiency for variable load applications. This will occur only as high torque is not required, in what is known as "low or partial load" operating conditions. It also is a way to automatically do Field Weakening on induction motors as the Id current will be reduced to meet speed target while still maintaining stability.

TI Spins Motors



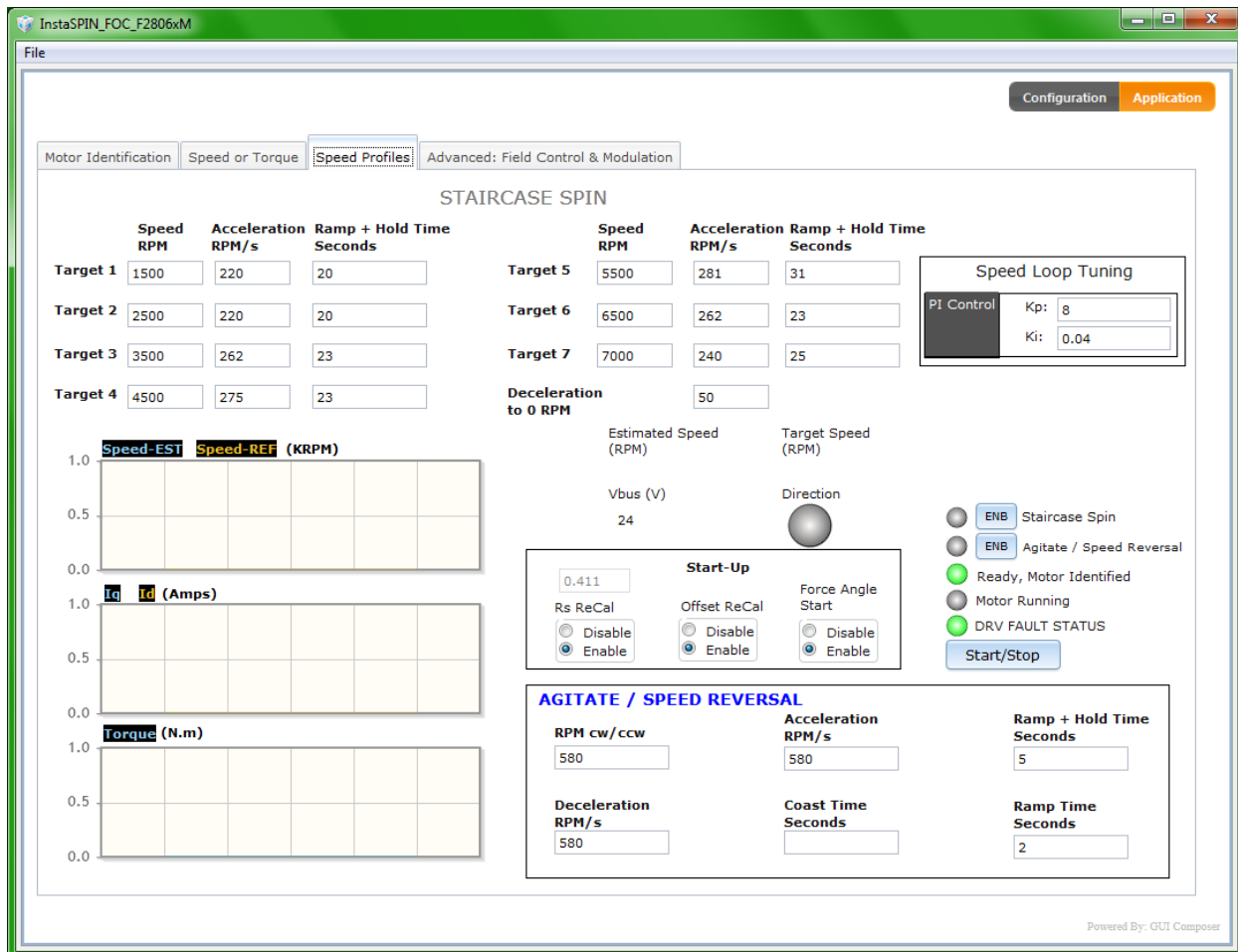
- When in PowerWarp mode - because the magnetic field is reduced - as torque is required (load increases) the dynamic response of the system will be lower (it will take longer to react and provide the needed torque vs. PowerWarp disabled). If the load disturbance is great enough it is possible that the motor will stall, but assuming it is not an overload condition FAST will continue to track the angle and the motor will recover and reach the speed reference once again.
- If Enabled for a synchronous (BLDC/PMSM) motor there will be no change to the control system or to performance

TI Spins Motors

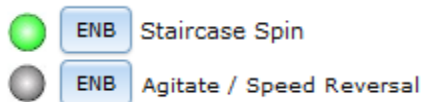


Tab3: Speed Profiles

The Speed Profiles tab is provided with two different state machine motion profile examples to enable **evaluation** of common variable speed applications.



Staircase Spin



- Enabled by selecting  Staircase Spin

TI Spins Motors



STAIRCASE SPIN

	Speed RPM	Acceleration RPM/s	Ramp + Hold Time Seconds		Speed RPM	Acceleration RPM/s	Ramp + Hold Time Seconds
Target 1	1500	220	20	Target 5	5500	281	31
Target 2	2500	220	20	Target 6	6500	262	23
Target 3	3500	262	23	Target 7	7000	240	25
Target 4	4500	275	23	Deceleration to 0 RPM		50	

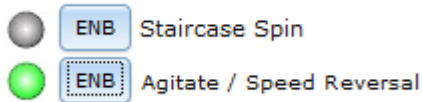
- This Motion Profile sets a SpeedRef (RPM) to the controller using the following logic
 - 0 RPM Start ramp to Target 1 Speed @ Target 1 Acceleration
 - At Target 1 Time, ramp to Target 2 Speed @ Target 2 Acceleration
 - After additional Target 2 Time, ramp to Target 3 Speed @ Target 3 Acceleration
 - After additional Target 3 Time, ramp to Target 4 Speed @ Target 4 Acceleration
 - ...
 - After additional Target 7 Time, ramp to 0 RPM at Deceleration
 - **Motors at high speed are creating a large B_{emf} voltage. Caution should be taken during Deceleration to do so slowly or an over voltage situation will occur resulting in a fault and possibly damaging the inverter or the motor.**
- Speed can be positive or negative
- Acceleration/Deceleration should always be entered as a positive number
- Capability is heavily dependent on the motor, mechanics, inertia, power supply, and load. Do not assume that a motor will track any motion profile that is entered.
- Disable RsReCal for proper operation of the counters during Speed Profiles
 - The timers start when the run-time controller is in on the on-line state. If Enabled, RsReCal is run during the on-line state and the first ramp + hold time will be offset by ~8 seconds
- Enable Force Angle Start every time you start the drive
 - After Start-Up
 - Enable/Disable does not matter for a profile that does not cross zero speed
 - Disable if your motion includes transitions straight through zero, or short transition durations at zero speed
 - Leave Enabled if your motion includes longer periods near and at zero speed. FAST continues to track well moving through zero, and with larger B_{emf} motors and good sensing it can hold zero for some amount



TI Spins Motors



of time. Evaluate if the performance Enabled / Disabled is better for your application.

Agitate / Speed Reversal Mode



- Enabled by selecting   Agitate / Speed Reversal

AGITATE / SPEED REVERSAL		
RPM cw/ccw	Acceleration RPM/s	Ramp + Hold Time Seconds
<input type="text" value="580"/>	<input type="text" value="580"/>	<input type="text" value="5"/>
Deceleration RPM/s	Coast Time Seconds	Ramp Time Seconds
<input type="text" value="580"/>	<input type="text"/>	<input type="text" value="2"/>

- This Motion Profile sets a SpeedRef (RPM) to the controller using the following logic
 - 0 RPM Start, ramp to RPM cw/ccw @ Acceleration
 - After Ramp + Hold Time
 - If desired a Coast Time (motor freewheels) can be entered
 - Typically 0
 - After Coast Time
 - If desired a Ramp Time can be entered
 - Ramp towards 0 RPM @ Deceleration
 - If time remains, Hold @ 0 RPM
 - If 0 is entered for Ramp Time, or a value lower than necessary to bring the motor to 0 speed due to inertia, load, or deceleration setting, the motion profile will immediately repeat in the opposite direction
 - Repeat in opposite direction (- RPM cw/ccw)
 - Repeat forever
- RPM can be positive or negative (depending on initial starting direction)
- Acceleration/Deceleration should always be entered as a positive number

TI Spins Motors



- Capability is heavily dependent on the motor, mechanics, inertia, power supply, and load. Do not assume that a motor will track any motion profile that is entered.
- Disable RsReCal for proper operation of the counters during Speed Profiles
 - The timers start when the run-time controller is in on the on-line state. If Enabled, RsReCal is run during the on-line state and the first ramp + hold time will be offset by ~8 seconds
- Enable Force Angle Start every time you start the drive
 - After Start-Up
 - Disable if your motion includes transitions straight through zero, or short durations at zero speed
 - Leave Enabled if your motion includes longer periods near and at zero speed
 - FAST continues to track well moving through zero, and with larger B_{emf} motors and good sensing it can hold zero for some amount of time. Evaluate if the performance Enabled / Disabled is better for your application.
 - Low speed agitation with load is one of the most challenging control scenarios. In a real application the currents are often quite large, causing R_s values to increase dramatically. Please see the User's Guide for discussion and implementation of on-line continuous R_s recalibration.

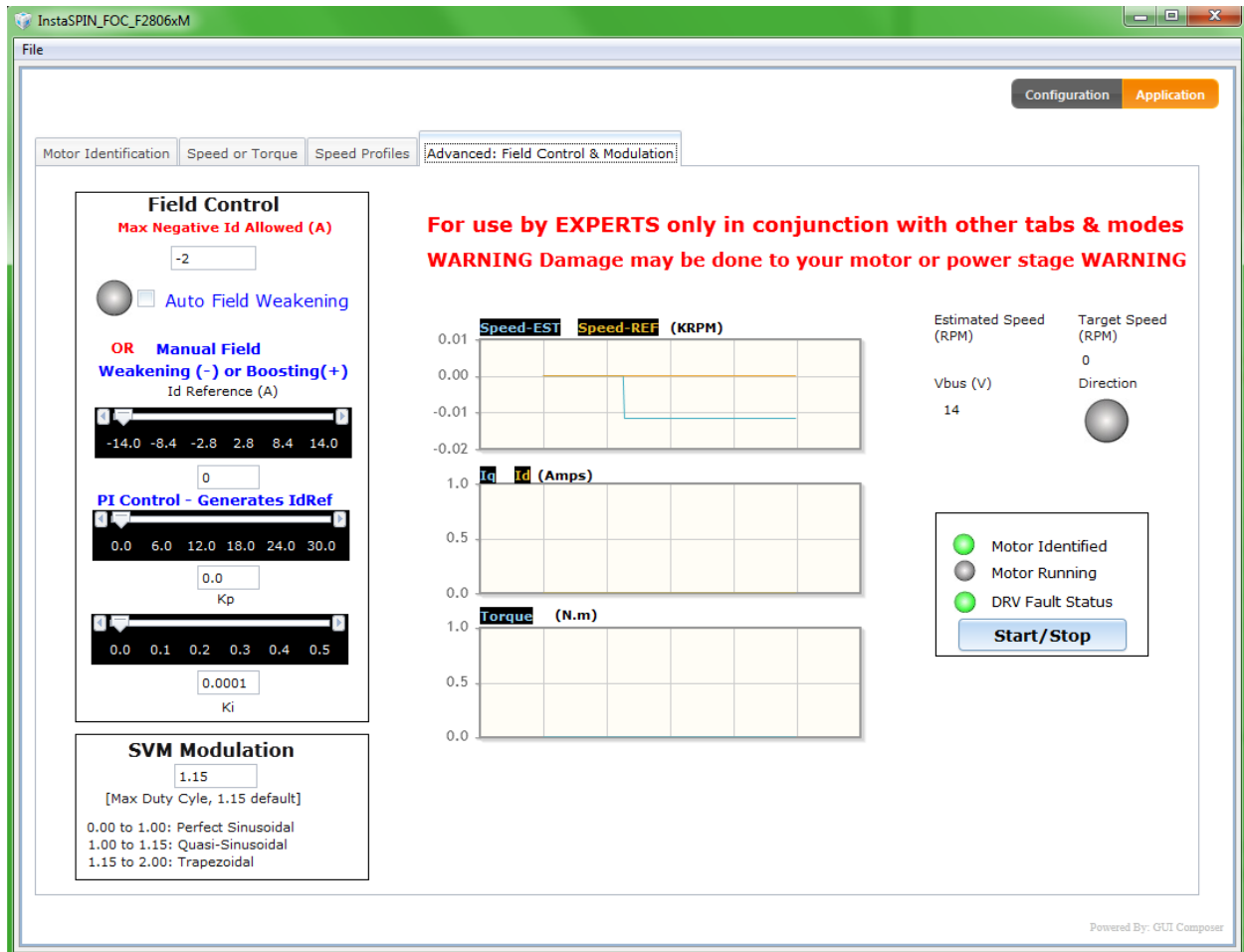
TI Spins Motors



Tab4: Advanced: Field Control & Modulation

The Advanced Field Control tab is meant to be used in conjunction with the other tabs, typically Speed+Torque, Staircase Spin (for field weakening), or Torque Mode (for field boosting). Once operating in those modes you can use this tab to further control the field.

This should only be used with those of knowledge of field weakening. Using field weakening improperly may permanently damage your motor and/or inverter.



Field Control

- Max Negative Id Allowed
 - Safety feature active ONLY for Automatic Field control
 - BE CAREFUL USING MANUAL CONTROL
 - Too large of $-Id$ can demagnetize a motor. Insure that $\sqrt{Id^2 + Iq^2} < I_{rated}$ or your permanent magnet motor will demagnetize.

TI Spins Motors



- Auto Field Weakening
 - Enabling will increase the $-I_d$ until Estimated speed reaches Target speed, or $-I_d$ reaches Max Negative I_d Allowed
 - In low voltage motors you may see the $-I_d$ go straight to the maximum allowed. If this is the case you may want to use manual field weakening or investigate alternative automatic field weakening algorithms.
 - Once Auto Field Weakening applies a $-I_d$ value,
 - As you reduce your speed below the field weakening RPM you will need to manually increase your $-I_d$ towards 0.
 - This should be done incrementally and slowly, not all at once or you risk an over voltage condition on the bus
 - This is just one implementation of automatic field weakening to be shown as an example. There are many additional ways to implement field weakening that may work better for your application.
- Manual Field
 - Always active/available if Auto Field Weakening is NOT selected
 - This is the standard way field weakening is usually performed, where a look-up table is created with a set of I_d values for a given speed and load.
 - I_d Reference manually sets the end Target I_d (typically zero for standard FOC) to be controlled
 - (-) makes for field weakening, allowing higher speeds but less torque
 - (+) makes for field boosting, allowing for higher torque for induction machines
- PI Control
 - Controls the gains of the controller whose output is the I_d reference
 - When an I_d is entered you do not want this value to be a step input to the control system or an instability may occur
 - The PI controller lets you tune the response of adding field control
- Note that during field weakening the speed loop often becomes a larger control challenge. You will need to re-tune (soften) the speed controllers to avoid against oscillation. Review the User's Guide for guidance on PI tuning.

SVM Over-Modulation

- Use caution changing this setting when the drive is enabled, especially if the duty cycle is operating near the modulation limit. Recommend to change when the drive is disabled to avoid faults.
- This setting changes the maximum duty cycle and hence maximum speed.
 - 0.00 to 2.00 valid range

TI Spins Motors



- Entries < 0.00 will clip to 0.0, >2.00 will clip to 2.00 internally (will still display the entered value)
- Typical values:
 - 0.8667 = Standard Space Vector with no over modulation, 87% duty cycle
 - 1.15 = Space Vector with slight over modulation to achieve 100% peak duty cycle
 - 2.00 = 100% duty cycle with trapezoidal waveform
- Note: in MotorWare a different scale range is used for modulation indices and an updated SVM module is available. This uses a scale of 1.0 for pure sinewave and 1.333 as the maximum trapezoidal modulation.
- The User's Guide includes detailed description of this topic.

Example performance

- DRV8312-69M-KIT with provided Telco NEMA17 BLDC

Modulation	Max RPM Typical (no load)
0.1	500
0.5	2570
0.85	4400
1.15	5700
1.50	6155
1.75	6285
2.00	6360

Field Weakening	Max RPM Typical (no load, 1.15 Modulation)
0	5700
-0.2	5850
-0.5	6090

TI Spins Motors



-1.0	6500
-1.50	6995
-2.0	7540

TI Spins Motors



Shutting Down

- Stop the Drive



- Close the GUI



- Remove bus power

WARNING



Do not close the GUI until the drive has been stopped. Failure to do so will leave the program running or put the processor into an unknown state, causing the system to continue to draw current, possibly damaging the controlCARD, board, host computer, motor and posing a fire hazard. After proper shut-down always disconnect the power supplies and remember that **capacitors are charged and will take time to dissipate!**