Motor Control

Sensorless 3-Phase PMSM Motor Control

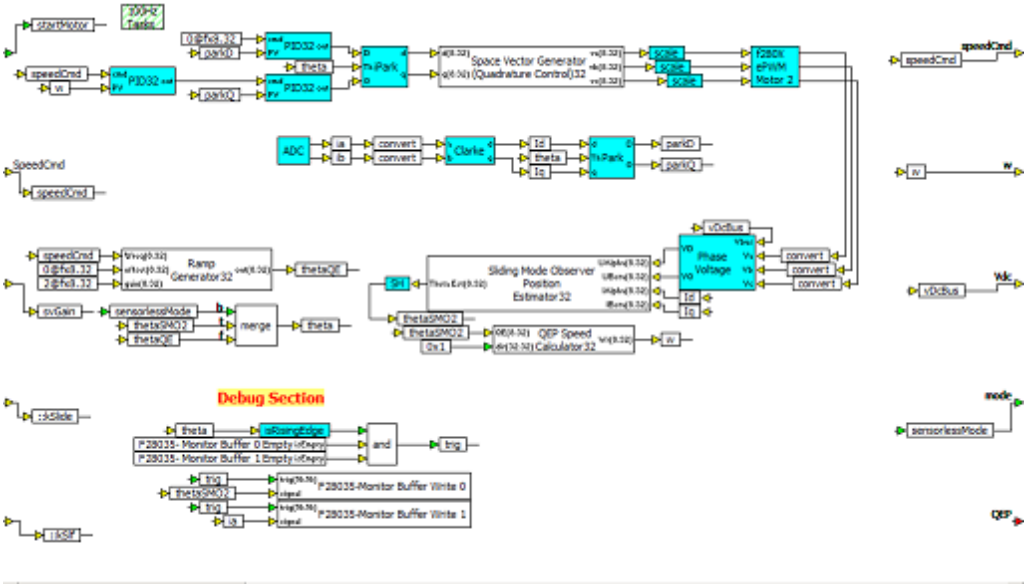
TI DRV8312EVM Sensorless Field-Oriented Control Motor Kit

Running the Embed Motor Control Models

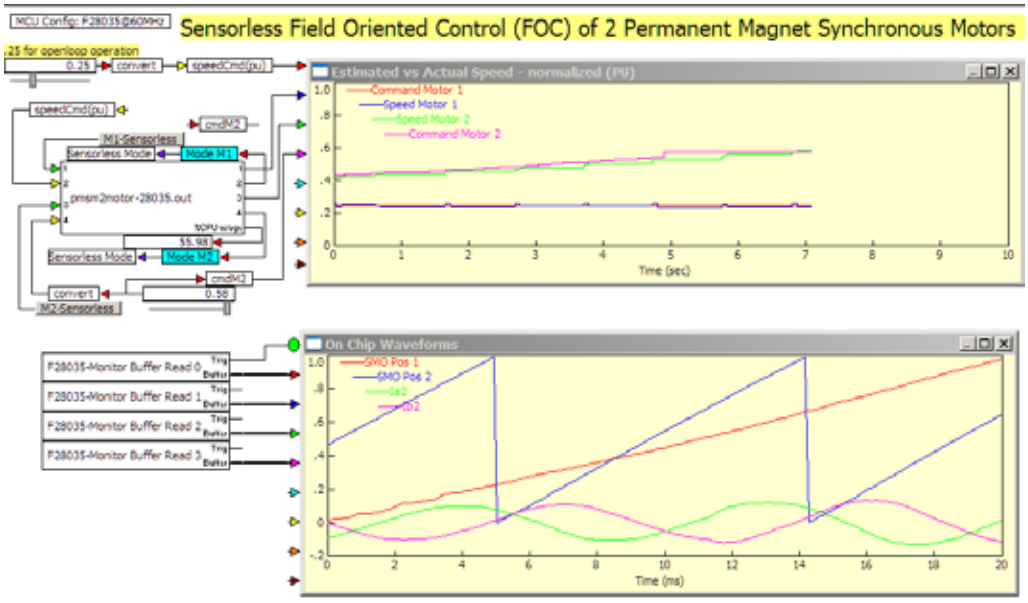
Location: Examples > Embedded > Digital Motor Control > PMSM
Name: DRV8312EVM-pmsm-28069

These models — discussed below — generate C-code to control both the DRV8312EVM and tmds2mtrpfckit sensorless PMSM motor kits from Texas Instruments.

The diagrams control a 3-phase brushless permanent magnet synchronous motor (PMSM). The DRV8312EVM board consists of a logic section with the F28035 Piccolo controlCARD (the socket will fit any of TI's C2000 controlCARD series) with an on-board USB to XDS100 JTAG, and a DRV8312 MOSFET. The TI MOSFET chip has 3 1/2 H-Bridges, is rated up to 325W and operates at 97% efficiency. In addition, there are high precision op-amps tied to current sensors on the ground leg of each output phase, as well as the DC bus. Each op-amp is wired to ADC channels on the C2000 controlCARD to allow accurate phase current measurements.



Embed source diagram to control TI Motor Control Kit



Embed diagram to test and debug FOC controller on F28035-based TI Motor Control Kit

Embed Controls the Hardware

The Altair Technical Services group has created diagrams to drive the Texas Instruments Sensorless Motor Control Kit based on the F28035 Piccolo low-cost controlCARD. The controller is called sensorless because there is no direct rotor position sensor. Instead, rotor position is inferred from the back EMF on the stator windings using current sensors on the inverter board. The current sensor readings are used by a sliding mode observer rotor position estimator block. The diagram was assembled using standard Embed blocks such as configurable PWM blocks wired to pre-built subsystems, like fixed-point Park and Clarke transforms, PI controllers, ADC inputs, GPIOs and logic to control the operating mode of the inverter (open-loop ramp driven vs closed-loop sensorless). An intermediate debug cycle ensured correct settings and configuration, but it was done at high level, so bugs due to C syntax and missing libraries were not a factor.

Simple Creation of Low-Speed Background Task

Click on a subsystem and you can configure the subsystem to execute at any rate. You can also set it to run as a background task at that rate. The background task will run at the specified rate but is preempted by the main control task. Any blocks placed in the subsystem will run at the specified rate. Typically, supervisory commands on a CAN or UART bus are transmitted and received here. In this case, the motor control method selection logic (open-loop ramp speed vs closed-loop sliding mode observer regulated speed) are placed in the subsystem. In the motor control diagram, the main control sample rate is 10kHz and one background task is set to run at 100Hz.

From Diagram to Code with the Click of a Mouse

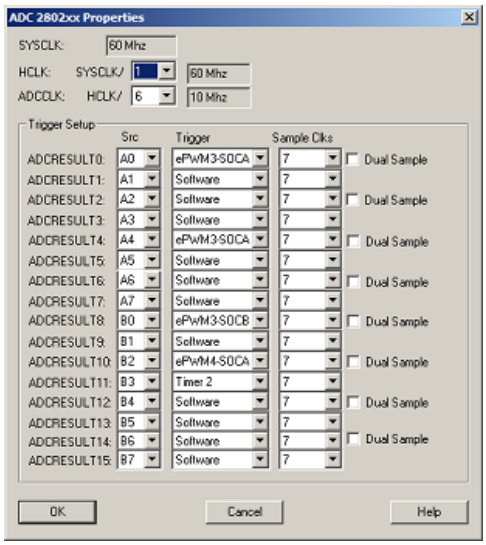
In Embed, the classic PMSM motor control loop is constructed from standard fixed-point blocks coupled with blocks in the Digital Motor Control block set. The phase currents feed a Clarke and Park transform, with PID control on the Park D commanded to zero, and Park Q output (torque) commanded by the outer velocity PID loop output. This is seen in the Embed diagram to the right. These blocks are placed inside a compound block with edge connectors for the values supplied from Embed running on the PC while the compiled control runs on the Piccolo target. In this case, a button input is supplied to select between three modes of operation: motor off, motor in open-loop mode, and motor in sensorless closed-velocity-loop mode. There are also input pins to vary the gain on the second harmonic space vector waveform generators feeding the PWMs and to vary settings of SMO gains.

To generate code and compile, click to the top level, select the encapsulated compound block, and select **Tools > Codegen**. Clicking the **Compile** button generates the code and runs the linker to create an OUT file for the F28035 target.

Debug with Interactive Parameter Change from the PC

Debugging takes place in the debug diagram on the PC. This diagram complements the source code diagram from above. In the debug diagram the compound block containing the motor control is replaced with a targetInterface block. The targetInterface block is initialized with the path to the OUT file generated by Embed. When **System > Go** is selected, the targetInterface block will automatically download the OUT file and begin

communication with the target. This diagram is run in real time (using the **System > System Properties** command) at around 100Hz since the JTAG cannot support a communication rate much faster. By convention, the debug diagram shares the same name as the source code diagram, but has a "-d" suffix.



Embed Piccolo ADC Config Dialog

Synchronize ADC with PWM Waveform

Another important consideration is to synchronize the ADC unit with the PWM waveform on the power device being measured. In this kit, the phase current sensors for Motor 1 are wired to ADCA0 and ADCA1. Motor 2 Ia is wired to ADCA4 and Ib to ADCB0.

The F28035 Piccolo processor has an updated ADC architecture from the F280x series that allows each ADC channel to select any of the individual PWM events as a start of conversion signal. This is set up in the **Embedded > F280X > ADC Config** dialog. PWM units driving the DRV8402 are set to send a start-of-conversion signal (SOC) when TBCTR=PRD (period match). Since the PWM is set for up/down count, the measurement will be made in the center of the **off** portion of the PWM duty cycle. This works because the PWM is used to drive the high side of each phase leg, the current sensor is on the low side, and the DRV8402 automatically activates the low side when the PWM driving the high side is off. So we have set ADCA0 to trigger from PWM1-SOCA, and ADCA1 from PWM1-SOCB. For Motor 2, we trigger ADCA4 from PWM3-SOCA, and ADCB0 from PWM3-SOCB.

Waveform Capture for Debugging

For debugging, the estimated rotor position — given by the Sliding Mode Observer Position Estimator and phase currents *ia* and *ib* — is written to a 200-element buffer at the 10kHz control sample rate, giving a 50ms window. The buffer acquisition is triggered by a positive cross at .01 on the estimated rotor position waveform. The waveform data is sent to Embed via the JTAG link and is updated in a digital scope in Embed on the PC at about 20Hz.