EMBEDDED CODE GENERATION DEMO MODEL

Vector Control of an Induction Machine

Control of an induction machine with embedded code generation for TI C2000 MCUs

Last updated in C2000 TSP 1.0

www.plexim.com

- Request a PLECS and PLECS Coder trial license
- Get the latest TI C2000 and RT Box Target Support Package
- Check the PLECS, RT Box and TI C2000 TSP documentation

1 Overview

This demo model shows the simulation of an induction motor drive system, and provides an explanation of the typical workflow of the PLECS Embedded Coder, using Texas Instruments (TI) C2000 MCUs. Combined with a PLECS RT Box, the performance of the MCU can be verified directly.

Note This model contains model initialization commands that are accessible from:

PLECS Standalone: The menu **Simulation + Simulation Parameters... + Initializations**

PLECS Blockset: The Simulink menu **File + Model Properties + Callbacks + InitFcn***

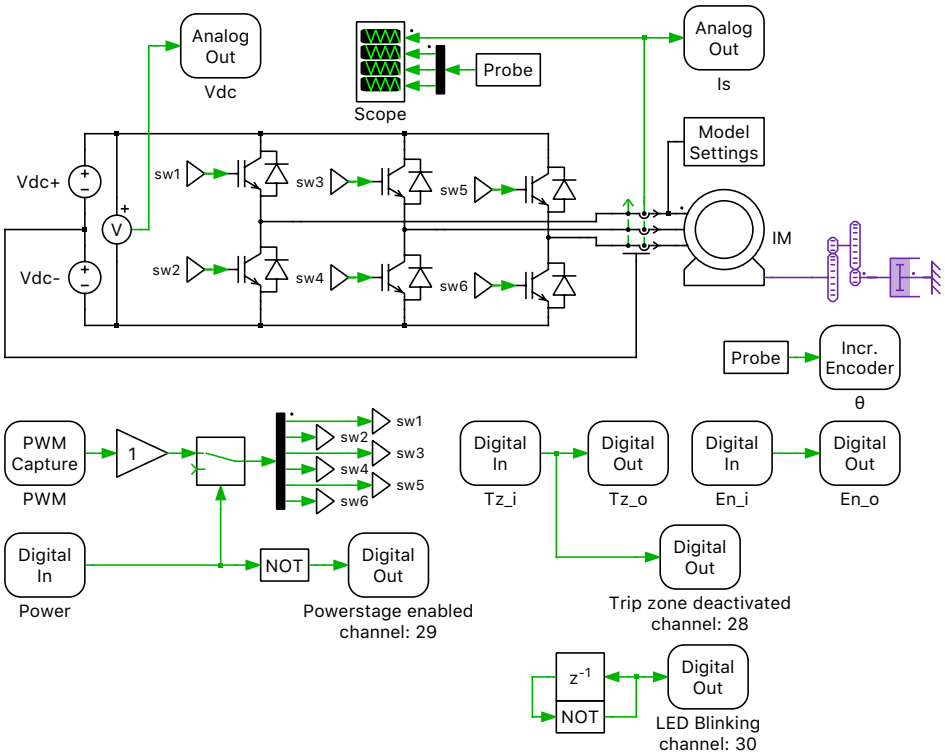

Figure 1: Power circuit of the induction machine drive system

2 Model

The demo model is composed of two subsystems: the “Plant” subsystem includes the power circuit and the “Controller” subsystem includes the control loop.

2.1 Power Circuit

The power circuit includes an induction machine (IM) and a three-phase full bridge voltage-source inverter (VSI). The mechanical interface of the IM is loaded by a rotational damper block via a gear box to emulate the linear friction. The VSI is supplied by a DC voltage source of $V_{\mathrm{dc}} = 400\,\mathrm{V}$, and is represented by three IGBT Half Bridge power modules.

The six pulse-width modulated (PWM) switching signals are brought into the subsystem using a PWM Capture block from the PLECS RT Box component library. The measurements of the DC voltage and

the AC current are exported out of the subsystem via Analog Output ports. The rotor angular position and rotational speed are converted into digital orthogonal pulses by an Incremental Encoder block. The encoder pulses can be accessed on the corresponding port of the "Plant" subsystem mask. Two Digital Output blocks forward the external digital input signals to two digital output pins of the RT Box. These pins should be connected with the GPIOs of the MCU to enable/disable the PWM outputs via software or trip zones. The mechanism of the MCU's PWM enable/disable will be elaborated in section 2.2.

2.2 Controls

In the "Controller" subsystem, rotor-field oriented control is applied to the drive system. The basic structure is shown in Fig. 2, where the stator current is regulated in the dq frame.

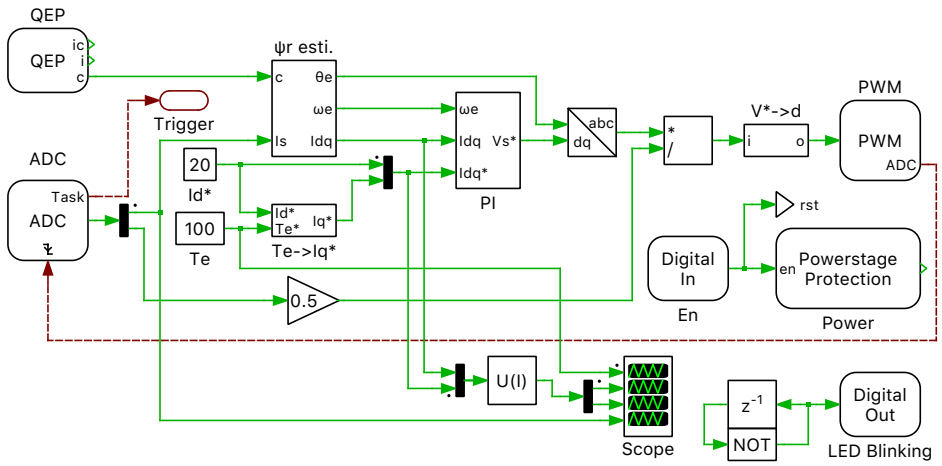

Figure 2: Controller model of the induction machine drive system

Fig. 3 shows the equivalent circuit of the induction machine in the dq frame, which rotates synchronously with the rotor flux. The values of L_{M}, $L_{\sigma\mathrm{S}}$ and R_{R} are calculated from the original machine parameters, which can be found in the initialization commands of the model.

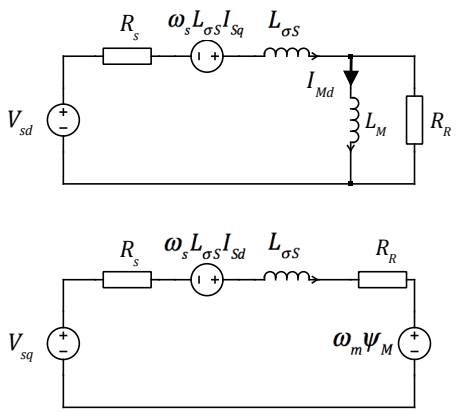

Figure 3: Equivalent circuit of the induction machine in the dq frame.

The PI controllers for the d and q axis current are included in the subsystem "PI", as shown in Fig. 4. The proportional and integral gains are designed following the "magnitude optimum" method [2][3].

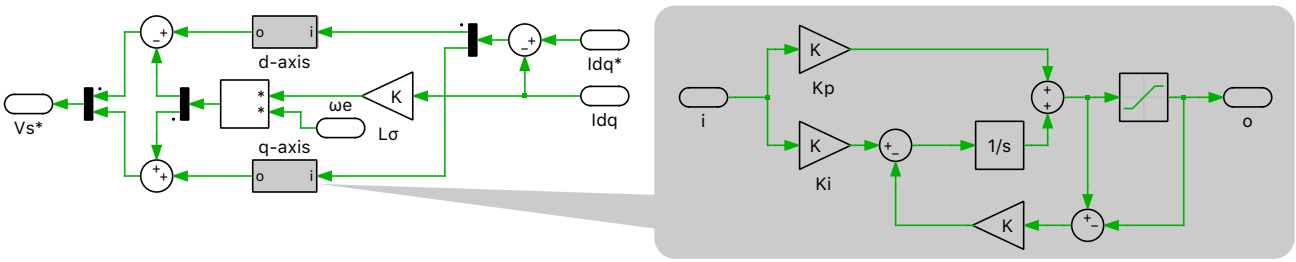

Figure 4: PI controller in the dq frame.

To avoid the use of a embedded flux sensor, a magnetic flux estimator introduced on page 322 of [1] is employed in the subsystem "Ψ_{r} esti.". Making use of the measured mechanical angular speed ω_{m}, the stator current is transformed into the rotor rotational frame ("xy") as $\vec{I}_{\mathrm{s,xy}}$. The rotor flux $d\vec{\Psi}_{\mathrm{r,xy}}$ in the rotor rotational frame is governed by the differential equation given below:

$$\frac{d\vec{\Psi}_{\mathrm{r,xy}}}{dt} = R_{\mathrm{R}}\left(\frac{-d\vec{\Psi}_{\mathrm{r,xy}}}{L_{\mathrm{M}}}\right) + \vec{I}_{\mathrm{s,xy}} \tag{1}$$

Following the differential equation, $d\vec{\Psi}_{\mathrm{r,xy}}$ can be calculated using $\vec{I}_{\mathrm{s,xy}}$ as an input. Processing the "x" and "y" components of the rotor flux, in the rotor rotational frame, using a "Rectangular to Polar" transformation block, yields the slip angular position. Summing up the slip angular position and the mechanical angular position, the electrical angular position θ is obtained. θ is then used to transform the stator current from the "abc" frame to "dq" frame. The structure of the rotor flux estimator is demonstrated in Fig. 5.

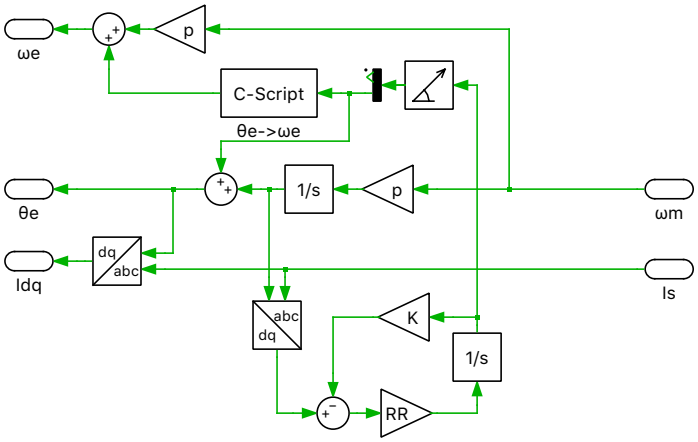

Figure 5: Structure of the rotor flux estimator.

The measurements of the DC-link voltage and the stator currents are introduced to the model environment from the ADC block of the TI C2000 Target component library. In order to convert the detected analog voltage into values with physical units to be used for the control algorithm, a scaling factor and an offset factor are provided to each channel, via the parameter window of the ADC block. The **ADC unit** and the **Analog input channel** parameters can be modified accordingly per available resources of different MCUs.

The mechanical angular speed of the rotor is obtained from the Quadrature Encoder Counter (QEP) block, which converts the orthogonal digital pulses into the rotational position-related counter value. The **Maximum counter value** must be provided according to the configuration of the real encoder hardware, along with the index of the **GPIO numbers**.

The gate signals are generated by the PWM block. The **Carrier type**, **Carrier frequency** and **Blanking time** parameters can be configured in an intuitive way from the **Main** tab of the PWM

block parameters window. The input to the PWM block is obtained from the controller in the form of a duty cycle, in the range of [0,1]. Please note that each duty cycle signal is fed to both A and B outputs of an ePWM module, but with opposite polarity. From the **Events** tab of the PWM block parameter window, the **ADC Trigger** parameter is configured as `Underflow`. This makes the first ePWM module configured in the **PWM generator** of the **Main** tab generate a "start of conversion" impulse for the ADCs, whenever the carrier value reaches its minimum. With this configuration, an additional output port called "ADC" appears on the mask of the PWM block. This is connected to the input port called "Trigger" of the ADC block, via the red dashed signal wire.

The control task is executed after each ADC conversion. This is configured by connecting the output port called "Task" of the ADC block to a "Control Task Trigger" port from the TI C2000 Target component library.

In order to quickly shut down the PWM gate signals in reaction to certain GPIO inputs under fault condition, three "Trip Zone" events (Tz1 - Tz3) can be configured to the PWM output channels. The behavior after each trip zone event can be chosen between "Disabled", "Cycle-by-cycle" and "One-shot" via the **TZx mode** parameter of the PWM block. The GPIOs to trigger the trip zone events Tz1 - Tz3 are defined by the **TZx GPIO number** parameter of the Powerstage Protection block. This block is also in charge of the software turn on/off of the PWM output and resetting the trip zone event, using its input port.

3 Simulation

In addition to running a simulation of this demo model in offline mode on a computer, the "Controller" subsystem can be directly converted into a target specific code for the TI 28069 LaunchPad [4], as configured.

Follow the instructions below to upload the "Controller" subsystem to a TI MCU.

- Connect the MCU to the host computer through a USB cable.
- From the **System** tab of the **Coder + Coder options...** window, select "Controller".
- Next, from the **Target** tab, select the appropriate target from the drop-down menu. Then under the **General** subtab, select the desired **Build type**.
- If "Generate code into CCS project" is selected, then locate the appropriate `cg` folder from the CCS project (refer to [8] for step-by-step instructions), into the **CCS project directory** field and click **Build**. The code of the "Controller" subsystem will be automatically generated. Then, proceed to build and debug the project as a normal CCS project.
- Alternatively, to "Build and program" the MCU directly from PLECS, enter the path of the appropriate `.ccxml` file into the **Uniflash target configuration** field (refer to [8] for step-by-step instructions) and click **Build**.

If programmed correctly, LED "D9" (or the LED corresponding to GPIO "DO_DSP_LED" mentioned under the model initilization commands) should blink.

Note If using the RT Box LaunchPad Interface board, make sure that the **RST** jumper is open throughout the simulation.

Prior to controlling a real hardware prototype of the power stage with the programmed MCU, it is highly recommended to first verify the behavior of the controller using a PLECS RT Box and perform a hardware-in-the-loop (HIL) test. A typical hardware configuration is shown in Fig. 6, where the evaluation kit, a TI 28069 LaunchPad (the red board), is connected to the RT Box via a RT Box LaunchPad Interface (the green board).

Follow the instructions below to run a real-time model on the RT Box.

Figure 6: Hardware setup of the HIL verification

- From the **System** tab of the **Coder + Coder options...** window, select “Plant” and **Build** it onto the RT Box.
- Once the model is uploaded, from the **External Mode** tab of the **Coder options...** window, **Connect** to the RT Box and **Activate autotriggering** to observe the test results in real-time.

If programmed correctly, the LED corresponding to “DO-30” of the RT Box LaunchPad Interface board should blink.

Within the “Plant” subsystem or the power circuit running on the RT Box, the simulated voltages and currents are proportionally converted into analog signals, and delivered through Analog Out connectors on the front panel of the RT Box. These analog signals are captured by the RT Box LaunchPad Interface board and routed to the ADC input pins of the TI LaunchPad. The MCU then processes these analog signals to generate PWM switching signals, which are delivered to the RT Box via the Digital In pins. Toggle the switch “DI-29” on the RT Box LaunchPad Interface board from “High” to “Low” and then back to “High” to reset the MCU and observe the real-time waveforms in the Scope of the “Plant” subsystem. When the power stage is enabled, the LED corresponding to “DO-29” of the launchpad interface board should turn on.

If switch “DI-28” on the RT Box LaunchPad Interface board is toggled to “Low”, a trip-zone event is activated and disables the output of all the PWMs. When the trip-zone event is activated, the LED corresponding to “DO-28” of the launchpad interface board is turned off. In order to resume the system, toggle “DI-28” back to “High”, turning the LED corresponding to “DO-28” on, then toggle switch “DI-29" from “High” to “Low”, and then back to “High” to reset the MCU. The LED corresponding to “DO-29” should now turn on.

Note At this stage, verify that the LEDs corresponding to “DO-28” and “DO-29” on the RT Box LaunchPad Interface board are turned on.

In order to tune the parameters of the control program in the MCU and observe any intermediate values, follow the instructions below to connect to the external mode of the TI MCU.

- First, **Disconnect** the “Plant” subsystem from the **External Mode** of the PLECS RT Box, if connected.
- Then, from the **System** tab of the **Coder + Coder options...** window, select “Controller”.
- Next, from the **External Mode** tab, select the appropriate **Target device** and click **Connect**.
- Then, **Activate autotriggering** to observe the test results in the “Controller” subsystem Scope.

In this demo model, a step change to the reference torque can be configured by changing the value of the Constant block “Te”, on the fly, in real-time, since it has been added to the **Exceptions** list in the

Parameter Inlining tab of the **Coder options...** window. The step response can be observed in real-time, as shown in Fig. 7, by setting up an appropriate **Trigger control** from the **External Mode** tab.

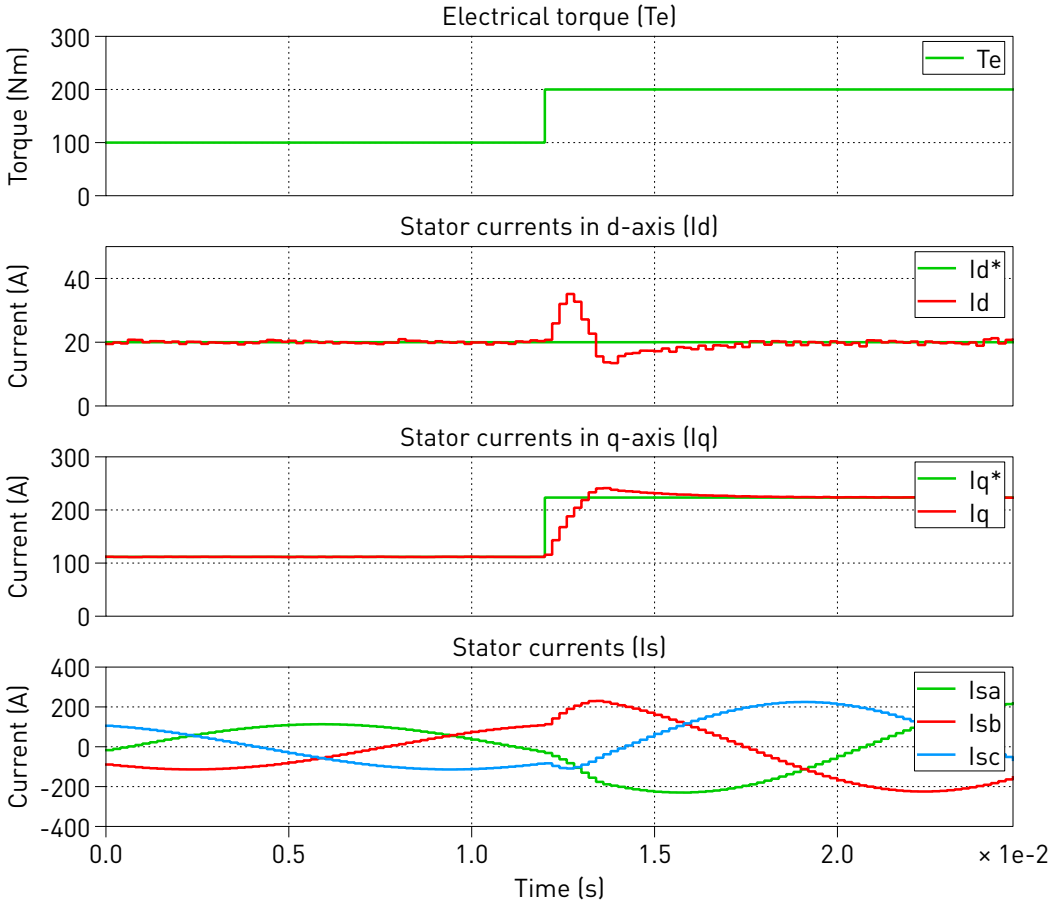

Figure 7: Transient response to a step change of the torque reference in the controller in real-time

Please note that the IO configuration of all the peripheral blocks (ADC, PWM) are configured by mapping to the TI 28069 LaunchPad [4]. For a TI MCU other than the TI 28069 LaunchPad, the IO configuration has to be adapted. In addition to the TI 28069 LaunchPad, this demo model also supports code generation for other TI C2000 MCUs, such as the TI 28377S [5], 28379D [6] and 280049C LaunchPads [7]. From the **Model initialization commands** window of **Simulation Parameters... + Initializations** tab from the **Simulation** menu, change the value of `type_evm`, to choose the desired target. Also, remember to configure the corresponding **Target** in the **Coder Options** window accordingly.

4 Conclusion

This model demonstrates an induction machine drive system that supports embedded code generation for TI C2000 MCUs.

References

[1] R. De Doncker, D. Pulle and A. Veltman, “Advanced Electrical Drives”, Springer, 2011.

[2] Konstantinos G. Papadopoulos, “PID Controller Tuning Using the Magnitude Optimum Criterion”, Springer, 2015.

[3] Damir Vrancic, “Magnitude Optimum Techniques for PID Controllers”, 2012.

[4] TI C2000 Piccolo MCU F28069M LaunchPad Development Kit,
URL: http://www.ti.com/tool/LAUNCHXL-F28069M.

[5] TI C2000 Delfino MCUs F28377S LaunchPad Development Kit,
URL: http://www.ti.com/tool/LAUNCHXL-F28377S.

[6] TI C2000 Delfino MCUs F28379D LaunchPad Development Kit,
URL: http://www.ti.com/tool/LAUNCHXL-F28379D.

[7] TI C2000 Piccolo MCU F280049C LaunchPad Development Kit
URL: http://www.ti.com/tool/LAUNCHXL-F280049C.

[8] PLECS TI C2000 Target Support User Manual,
URL: https://www.plexim.com/download/documentation.

Revision History:

C2000 TSP 1.0 First release

How to Contact Plexim:

☎	+41 44 533 51 00	Phone
	+41 44 533 51 01	Fax
✉	Plexim GmbH Technoparkstrasse 1 8005 Zurich Switzerland	Mail
@	`info@plexim.com`	Email
	`http://www.plexim.com`	Web

Embedded Code Generation Demo Model