

RT Box

DEMO MODEL

Sensorless Vector Control for Permanent Magnet Synchronous Machine

Cascaded speed and current control with an observer based position and speed estimation

Last updated in RT Box Target Support Package 3.0.3

www.plexim.com

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

1 Overview

This RT Box demo model features a drive system with a permanent magnet synchronous machine (PMSM). The following sections describe in detail the implementation of the power stage and controls using the PLECS Electrical and Control domains. This demo model has the following features:

- The drive outputs a nominal torque of $0.11\,\mathrm{Nm}$ and is fed by a $24\,\mathrm{V}$ DC voltage source.
- Explanation how to convert data sheet parameters of a PMSM to use in the PLECS PMSM model.
- The implementation of a rotor position and speed observer for sensorless field oriented control strategy.
- A closed-loop cascaded controller with inner d- and q- axis current controller and outer speed controller.
- The model includes both the plant and controller parts inside one subsystem. This allows the same model to be used in an offline simulation in PLECS and for real-time simulations on the PLECS RT Box[1].

The following sections provide a detailed description of the model and instructions on how to simulate it.

The execution time represents the actual time it takes to execute one calculation step of the PLECS model on the RT Box hardware. The chosen discretization step sizes and average execution times for each core in this demo model are shown in Tab. 1.

Table 1: Discretization step size and average execution time of the demo on one RT Box

	Core 0: exec. time / step size	Core 1: exec. time / step size
RT Box 2, 3 or 4	$3.5\,\mu\mathrm{s}$ / $5\,\mu\mathrm{s}$	$1.3\,\mu\mathrm{s}$ / $50\,\mu\mathrm{s}$
RT Box 1	$5\,\mu\mathrm{s}$ / $5\,\mu\mathrm{s}$	N/A

1.1 Requirements

To run this demo model, the following items are needed (available at www.plexim.com[2]):

- One PLECS RT Box[3] and one PLECS[4] and PLECS Coder[5] license
- The RT Box Target Support Package[6]
- Follow the step-by-step instructions on configuring PLECS and the RT Box in the Quick Start guide of the RT Box User Manual[7].
- Two 37 pin Sub-D cables to connect the box in loop-back setup at the front panel.

Note that this demo model primarily showcases the RT Box `multi-tasking` mode. All RT Boxes support multi-tasking, but the implementation differs depending on the model.

- When the target is an RT Box 2, 3 or 4, the main CPU core (Core 0) runs the plant as “Base task” with a sample time of `Ts_plant`. Another core (Core 1) runs closed-loop controls in “Controller” task in parallel with a sample time of `Ts_controller`, which is much slower and usually equals the switching period of the converter. In this way, the multi-core feature of the RT Box 2, 3 or 4 is showcased by splitting the computational effort onto different cores. Besides, the setup can easily transition to a HIL or RCP test later on.
- However if the user has only a single RT Box 1 available, this model can also run with the multi-tasking feature onto the only CPU core of the RT Box 1, but in a pre-emptive multi-tasking fashion. In this case, the “Base task” is doing the plant calculation with the highest priority with a sample

[1] https://www.plexim.com/products/rt_box
[2] http://www.plexim.com
[3] https://www.plexim.com/products/rt_box
[4] https://www.plexim.com/products/plecs/plecs_standalone
[5] https://www.plexim.com/products/plecs/plecs_coder
[6] https://www.plexim.com/download/rt_box
[7] https://www.plexim.com/sites/default/files/rtboxmanual.pdf

time of `Ts_plant`. The "Controller" task is executed as a background task with lower priority at the sample time of `Ts_controller`.

Please check the setting under **Scheduling** tab of the **Coder options...** window.

Note

This model contains model initialization commands that are accessible from:
PLECS Standalone: the menu **Simulation > Simulation Parameters... > Initializations**
PLECS Blockset: right click in the **Simulink model window > Model Properties > Callbacks > InitFcn***

2 Model

The top-level schematic of the demo model is depicted in Fig. 1. The model contains one subsystem: "Plant + Controller". The subsystem is enabled for code generation by right-clicking on the component and selecting **Enable code generation** check box from the **Subsystem + Execution settings...** menu. This allows the model to be simulated and the controls to be verified offline in PLECS before a real time simulation on the RT Box target.

The inherit delays of the closed-loop control are modelled in a simplified way in the offline simulation using a Delay block (z^{-1}) with sample time `Ts_control`.

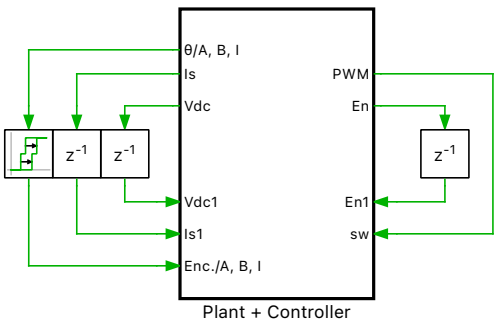


Fig. 1: Top level schematic of the demo model sensorless speed controlled PMSM

2.1 Plant

The power circuit includes an PMSM and a three-phase full bridge voltage source inverter (VSI). The DC voltage source V_i with $V_\mathrm{dc} = 24\,\mathrm{V}$ supplies the VSI, which is represented by three MOSFET Half Bridge power modules.

The six switching signals are fed into the subsystem by a PWM Capture block from the PLECS RT Box component library. The measurements of the DC voltage and the AC current output by Analog Output blocks. The rotor angular position is converted by the Incremental Encoder block into digital orthogonal pulses, which can be measured outside of the "Plant + Controller" subsystem.

The rotor inertia absorbs energy when the motor is accelerating and thus provides a load when starting up or changing speed. For this purpose the value from data sheet of Tab. 2 was used in the PMSM block, as internal mechanical load. For closed loop startup, the load cannot be higher than 40 %, otherwise the observer will not be able to track the speed in time and the control will become unstable. To demonstrate a load step, a load profile alternates between 20 % and 40 % of the nominal motor torque. The load is not applied until the controller is switched on.

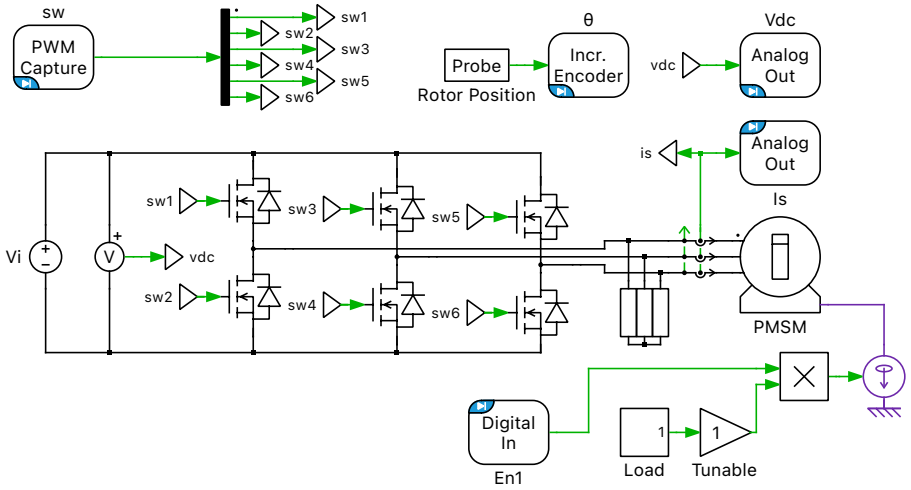


Fig. 2: Power circuit of the PMSM drive system

Machine Parameters

The machine parameters are listed in Tab. 2 which were taken from the data sheet. The following parameters must be converted in order to use them in the PLECS model:

- Line to line quantities must be converted in line to neutral quantities such as stator resistance R and stator inductance Ld and Lq. The conversion factor is $\sqrt{3}$.
- The permanent magnet flux linkage ψ can be calculate from line to neutral back EMF voltage e_{ln}. Which is the peak voltage induced by the permanent magnets in each phase per unit rotational speed. The relation can be expressed as follows:

$$e_{\mathrm{ln}} = p\,\psi\,\omega \tag{1}$$

In Tab. 2, the back EMF voltage e_{ln} was measured at a speed of 1000 revolution per minute (rpm). However, the PLECS model parameters are in radiants per second (rad/s) which requires a conversion.

Note

The machine model is chosen as `Rotor reference frame`. This configuration allows the generation of code for this model, however, it does not allow the simulation of the machine in generator mode together with dead-times on the inverter half-bridges.

Table 2: Specifications of applied PMSM BLWS232D-36V-4000

Number of pole pairs	2
Rated Power (W)	40
Rated Torque (Nm)	0.1
Rated Speed (rpm)	4000
Line to Line Resistance (Ω)	2.4
Line to Line Inductance (mH)	4.39
Rotor Inertia (Nms^2)	0.0000074852
Back EMF Voltage (V/krpm)	4.5

2.2 Controls

In the Controller task, the DC-link voltage and the stator currents are measured by Analog In blocks. The mechanical angular speed of the rotor is obtained from the Quadrature Encoder Counter block, which converts the orthogonal digital pulses.

The initial mechanical speed reference value is set at $2000\,\mathrm{rpm}$. In order to guarantee a robust start-up process of the observer, it is always started at a speed of $2000\,\mathrm{rpm}$ and regulated to the desired speed after the start-up. The speed of $500\,\mathrm{rpm}$ cannot be undercut in connection with the attached load in order to guarantee stable operation of the observer. Otherwise the observer will not be able to track the speed in time and the control will become unstable.

For the startup scenario a Finite State Machine is used. The system is first started with a current ramp in current control mode in order to align the position of the machine. After the current ramp and a short settling time, the speed control is started. If the drive system is operated with encoder, this startup scenario is omitted. The observer is already set to an initial speed of $500\,\mathrm{rpm}$ for a better transient response during startup. For robust system behavior, an abrupt speed change is limited with a rate limiter.

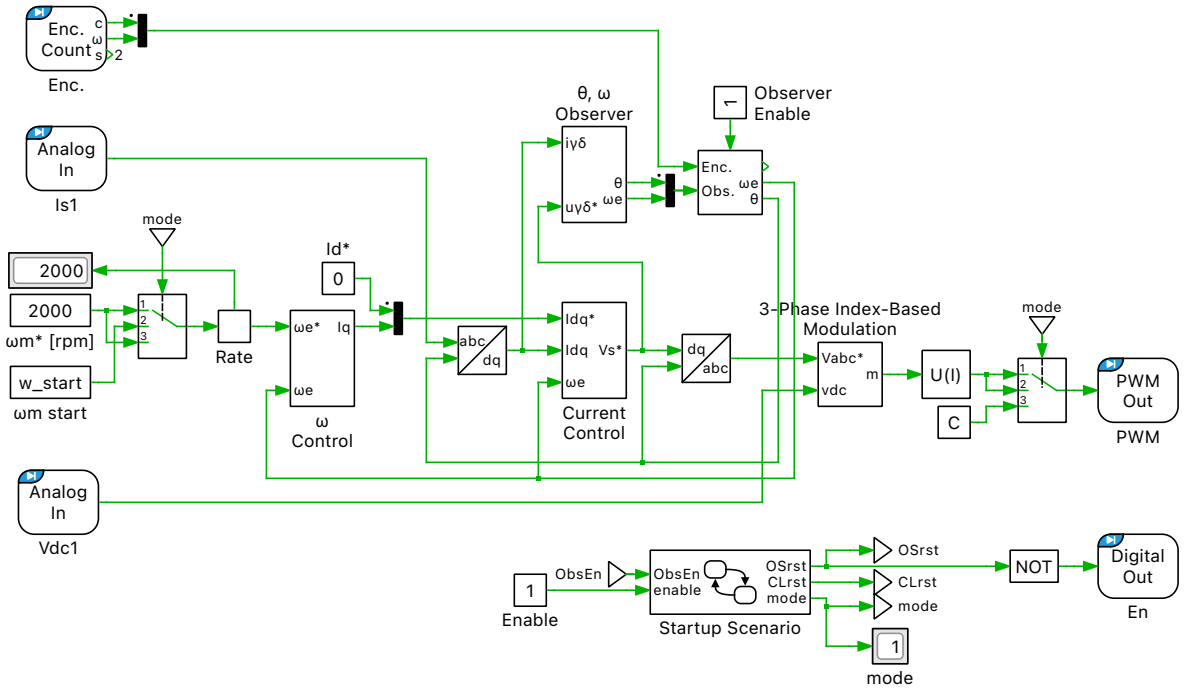


Fig. 3: Controller model of the sensorless rotor-field oriented control

Rotor-field oriented control is applied to the drive system and the basic structure is shown in Fig. 3, where the stator current is regulated in the dq-frame. The speed and rotor position can be calculated by either using the Observer subsystem or Encoder Counter target block. This selection is made by the "Selector" block, where the observer can be enabled or disabled. A higher-level controller is implemented for speed control, whereby a speed reference in rpm can be set. The complete controller can be enabled or reset via "Enable". In this case, the model continues to run in open loop with a fixed modulation index. In further investigations where the plant part is replaced by real hardware, this Digital Out is needed for enabling and disabling.

The voltages are the control variables and are represented in the dq-frame which can be derived from Eq. (2) while $L_{\mathrm{sd}} = L_{\mathrm{sq}}$. The currents are controlled with a PI controller, which is explained in the following.

$$\begin{bmatrix} u_{\mathrm{d}} \\ u_{\mathrm{q}} \end{bmatrix} = \begin{bmatrix} R_{\mathrm{s}} & -\omega L_{\mathrm{sq}} \\ \omega L_{\mathrm{sd}} & R_{\mathrm{s}} \end{bmatrix} \cdot \begin{bmatrix} i_{\mathrm{d}} \\ i_{\mathrm{q}} \end{bmatrix} + \begin{bmatrix} L_{\mathrm{sd}} & 0 \\ 0 & L_{\mathrm{sq}} \end{bmatrix} \cdot \begin{bmatrix} \frac{d}{dt} i_{\mathrm{d}} \\ \frac{d}{dt} i_{\mathrm{q}} \end{bmatrix} + \begin{bmatrix} 0 \\ \omega \psi_{\mathrm{f}} \end{bmatrix} \tag{2}$$

The voltage reference generated in dq-frame is converted to abc-frame, and then passes through a 3-Phase Index-Based Modulation block to generate three-phase modulation indices. Inside the 3-Phase Index-Based Modulation block mask dialog, one can choose the simple `Sinusoidal PWM` or other Space Vector PWM modulation methods.

PI current controller

The d and q axis currents which are controlled by a PI controller are included in subsystem "Current Control", which is shown in Fig. 4.

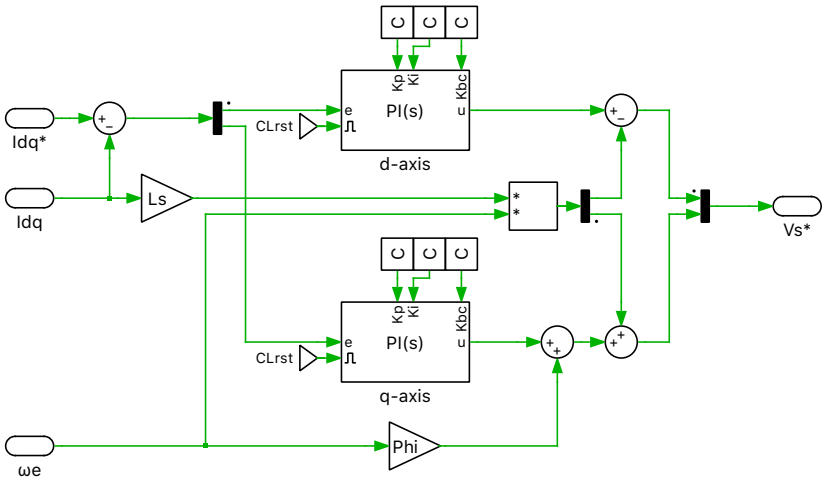


Fig. 4: PI controller in dq-frame

The proportional and integral gains are designed following the "Optimum Magnitude" method, which is described in more detail in the "Boost Converter". For the PI controller a PLECS library block was used, further descriptions can be found under help button. As the parameter source is set to external, the control parameters `Kp` and `Ki` can be manipulated during a real time simulation on the RT Box. Additionally the integrators can be reset via the "CLrst" signal.

Rotor position and speed observer

The "Observer" subsystem shown in Fig. 5 was developed according to [1] and is explained conceptually in the following. The observer can be utilized by any type of synchronous motor such as a surface PMSM (non-salient $L_\mathrm{d} = L_\mathrm{q}$), interior PMSM (with saliency $L_\mathrm{d} < L_\mathrm{q}$) and synchronous reluctance motor.

In the observer model the estimated rotating γ-δ frame is used, which differs from the d-q reference frame by the position error θ_e.

The position error θ_e is obtained from the extended electromotive force (EMF), which is estimated by the observer. The position error θ_e is used to estimate rotor position and speed. Fig. 6 shows the conceptional block diagram of the least-order observer for estimating the e_γ where g is the observer gain, which is set to $g = 600$. The δ-axis component of e_δ is estimated in the same way as e_γ. Comparing Fig. 5 with Fig. 6, the blue dotted frame outlines how the observer is embedded in the overall model.

The sign block evaluates the reference speed and gives a negative "sign" to position error θ_e, if the reference speed is negative. This is necessary during start-up of the drive to ensure correct tracking of the observer. All integrators of the observer can be reset by the "OSrst" signal.

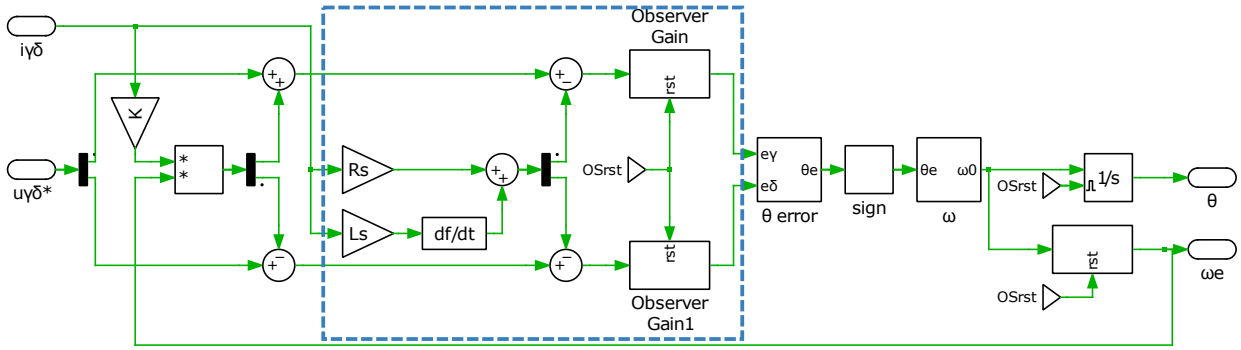


Fig. 5: Subsystem of rotor position and speed observer

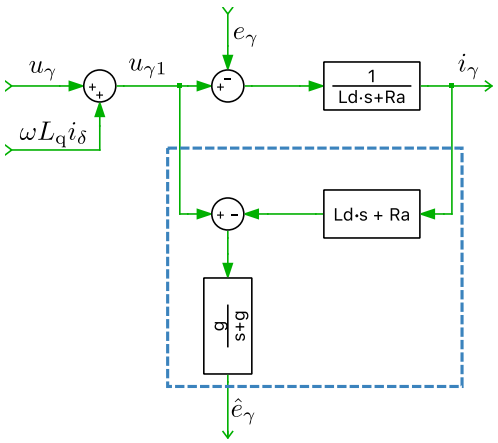


Fig. 6: Equivalent block diagram of least-order observer

Algorithm

The machine model used for the sensorless control algorithm is given in $\gamma - \delta$ coordinates including the extended EMF [1]:

$$\begin{bmatrix} u_\gamma \\ u_\delta \end{bmatrix} = \begin{bmatrix} R_\mathrm{s} & -\omega L_\mathrm{sq} \\ \omega L_\mathrm{sd} & R_\mathrm{s} \end{bmatrix} \cdot \begin{bmatrix} i_\gamma \\ i_\delta \end{bmatrix} + \begin{bmatrix} L_\mathrm{sd} & 0 \\ 0 & L_\mathrm{sq} \end{bmatrix} \cdot \begin{bmatrix} \frac{d}{dt} i_\gamma \\ \frac{d}{dt} i_\delta \end{bmatrix} + \begin{bmatrix} e_\delta \\ e_\gamma \end{bmatrix} \tag{3}$$

where

$$\begin{bmatrix} e_\delta \\ e_\gamma \end{bmatrix} = E_\mathrm{ex} \begin{bmatrix} -\sin\ \theta_\mathrm{e} \\ \cos\ \theta_\mathrm{e} \end{bmatrix} + (\hat{\omega} - \omega) L_\mathrm{s} \begin{bmatrix} -i_\delta \\ i_\gamma \end{bmatrix}. \tag{4}$$

Assuming the error between estimated speed $\hat{\omega}$ and actual speed ω is sufficiently small, Eq. (4) can be simplified and the second therm is eliminated, resulting in Eq. (5).

$$\begin{bmatrix} \hat{e}_\delta \\ \hat{e}_\gamma \end{bmatrix} = E_\mathrm{ex} \begin{bmatrix} -\sin\ \hat{\theta}_\mathrm{e} \\ \cos\ \hat{\theta}_\mathrm{e} \end{bmatrix} \tag{5}$$

By transforming Eq. (5) position error $\hat{\theta}_\mathrm{e}$ can be estimated in two ways, both schemes are implemented in the configurable subsystem "θ_e error":

Scheme A:

$$\hat{\theta}_\mathrm{e} = \tan^{-1}\left(-\frac{\hat{e}_\gamma}{\hat{e}_\delta}\right)$$

Scheme B:

$$\hat{\theta}_{\mathrm{e}} \approx \left(-\frac{\hat{e}_{\gamma}}{\hat{E}_{\mathrm{ex}}} \right)$$

A PI compensator $G_{\mathrm{e}}(s)$ is used to compensate the estimated speed $\hat{\omega}$ and estimated position $\hat{\theta}$, so that the position error $\hat{\theta}_{\mathrm{e}}$ becomes zero. During acceleration the estimated position error can be reduced using a 3. order instead of a 2. order compensator. Both are implemented in a configurable subsystem.

$$G_{\mathrm{e}}(s) = K_1 - \frac{K_2}{s} + \frac{K_3}{s^2}$$

The coefficients are designed with a natural frequency of $\omega_n = 70\,\mathrm{rad/s}$ and a damping factor of $\zeta_n = 1.5$ for the 2. order compensator according to the following rule: $K_1 = 2\zeta_n\omega_n, K_2 = \zeta_n\omega_n^2$. The 3. order compensator is designed according to the following rule: $s^3 + K_1 s^2 + K_2 s + K_3 = (s+\omega_n)(s^2 + 2\zeta_n\omega_n s + \omega_n^2)$. The cutoff frequency of the low-pass filter for the speed estimator is selected as $1000\,\mathrm{rad/s}$.

Speed Controls

To control the drive speed an *outer* speed control loop is provided. The time constants of the *outer* control loop is dictated by mechanical time constants, which is larger on comparison to the *inner* current control loop. The proportional-integral speed controller can be described as follows

$$\frac{T_{\mathrm{e}}^*}{\Delta\omega_{\mathrm{m}}} = K_{\mathrm{p}} \left(1 + \frac{1}{\tau_{\mathrm{i}} s} \right)$$

$T_{\mathrm{e}*}$ defines the reference torque and $\Delta\omega_{\mathrm{m}}$ is defined as $\Delta\omega_{\mathrm{m}} = \omega_{\mathrm{m}}^* - \omega_{\mathrm{m}}$. The controller is represented in terms of a proportional gain K_{p} and integral time constant τ_{i}. According to [2] page 117 ff. for the closed loop transfer function of the speed controller the bandwidth is effectively defined by the bandwidth frequency ω_{B}^{sp}, while the proportional gain is given as

$$K_{\mathrm{p}} = \omega_{\mathrm{B}}^{sp} J$$

The inertia J of the total mechanical load must be taken into account additionally for designing the control parameters. For this purpose, the total combined inertia must be estimated.

According to [2] $\omega_{\mathrm{B}}^{sp} = 100\,\mathrm{rad/s}$ is a typical value for the effective speed control bandwidth in high-performance drives. In this demo model, however, a lower value of $\omega_{\mathrm{B}}^{sp} = 70\,\mathrm{rad/s}$ was assumed due to the additional delay of the observer. The value ω_{B}^{sp}, together with the inertia J, fully defines the proportional gain of the controller.

For a damping factor of $\zeta = 1$ and the already defined K_{p}, the integral time constant τ_{i} is determined as follows:

$$\tau_{\mathrm{i}} = \frac{4}{\omega_{\mathrm{B}}^{sp}}\,.$$

The manipulated variable of the speed controller is the reference torque T_{e}^*, which in turn is the setpoint of the inner current controller loop. Therefore a conversion from mechanical torque T_{e} to i_{q} current has to be done regarding Eq. (6). The values for flux linkage Ψ_{m} of the permanent magnets and number of rotor permanent magnet pole pairs p are needed additionally.

$$T_{\mathrm{e}} = \frac{3}{2}\Psi_{\mathrm{m}} i_{\mathrm{q}} p \tag{6}$$

To perform this conversion the gain block with the coefficient K is used.

$$i_{\mathrm{q}}^* = \frac{T_{\mathrm{e}}^*}{K}$$

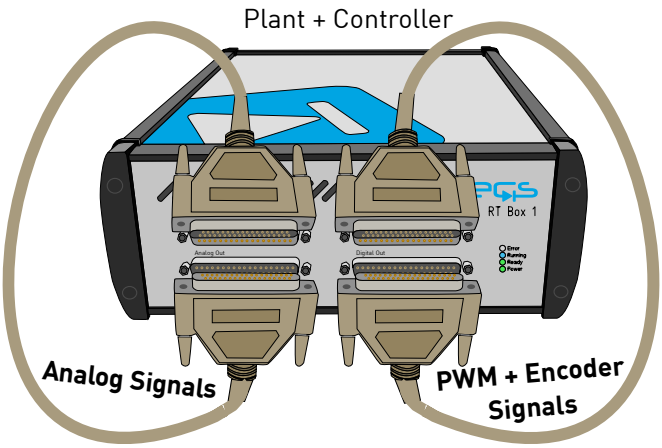


Fig. 7: Hardware configuration for the real-time operation of the demo model

3 Simulation

This model can run both, in offline mode on a computer or in real-time mode on the PLECS RT Box. For the real-time operation, one RT Box (referred to as "Plant + Controller") needs to be set up as demonstrated in Fig. 7.

Please follow the instructions below to run a real-time model on a single RT Box:

1. Connect the Analog Out interface to the Analog In interface with one DB37 cable, and the Digital Out interface to the Digital In interface with another DB37 cable (as shown in Fig. 7).
2. From the **System** tab of the **Coder options...** window, select the "Plant + Controller" subsystem and **Build** it onto the RT Box.
3. Once the model is uploaded, from the **External Mode** tab of the **Coder options...** window, **Connect** to the RT Box and **Activate autotriggering**.

3.1 Transient Scenarios

The load profile alternates between $0.04\,\mathrm{Nm}$ and $0.06\,\mathrm{Nm}$ with a nominal torque of $0.1\,\mathrm{Nm}$, which is shown in Fig. 8. The required torque can be compensated through the current controller in approx. $0.05\,\mathrm{s}$. The drop in speed, which is approx. $200\,\mathrm{rpm}$, can be compensated from the speed controller within $0.5\,\mathrm{s}$.

Step change of speed reference

The stator current, rotational angle and speed are shown in the scope of the "Controller" task. To observe a transient behavior of the system, e.g. a step change of the speed reference from $1000\,\mathrm{rpm}$ to $2000\,\mathrm{rpm}$, please further follow the scenario below:

- Make sure that the External Mode and **Activate autotriggering** of the RT Box is enabled.
- Switch the **Trigger channel** parameter to {tts[Observer]} in the **External Mode** tab of the subsystem's **Coder options...** window.
- Setup the **Trigger level** parameter to be 1500 and **Trigger delay [steps]** to be -4000.
- Change the Constant block "ω_m^*" from the default value of 1000 to 2000.

The step change will be captured by the scope in the "Controller" task, as shown in Fig. 9 on p. 11.

A comparison between the measured angle and speed from the encoder and the estimated angle and speed from the observer is shown. Nevertheless, the measured and estimated speed match each other very well. When the speed reference changes rapidly, the position estimation error and the speed estimation error are caused. However, this error is very small and has very little effect on the motor control.

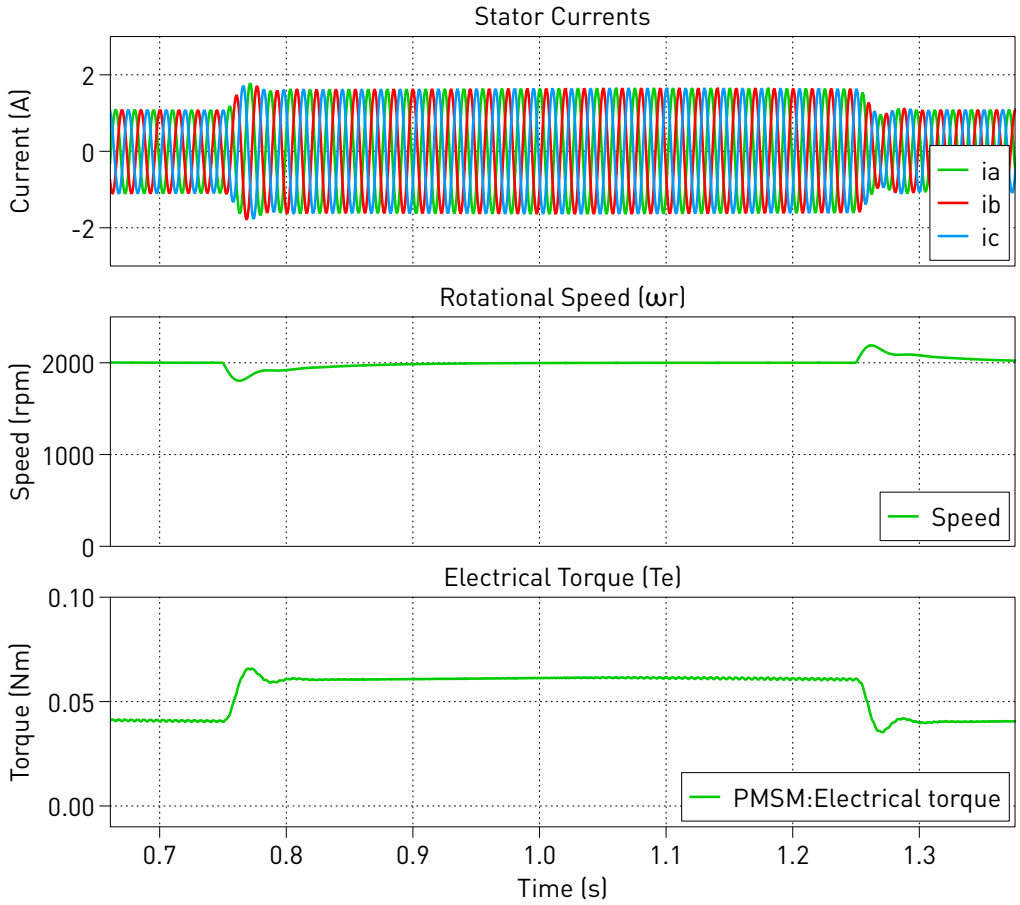


Fig. 8: Transient response of the speed controller for a load torque step from 0.01 Nm to 0.02 Nm (offline simulation results).

At speeds below $1000\,\mathrm{rpm}$, a larger speed ripple becomes visible. In general, the observer works better for higher speeds, but this effect is additionally amplified in lower speed ranges by a blanking time.

Note

The literature reference [1] does not describe a specific startup procedure for this sensorless control algorithm. However, the system start-up procedure is especially delicate when using sensorless control schemes. This demo model uses a simple open-loop start-up procedure to ramp up the controls and system operation but does not go into further details. This means depending on the exact timing of events in a real-time simulation the control may fail to start up properly. In such a case the model needs to be rebuilt.

3.2 Conclusions

This model demonstrates a PMSM drive system which can run in both offline simulation and real-time operation for Hardware-in-the-loop testing and rapid control prototyping. Rotor-field oriented control is applied to the drive system where the stator current is regulated in the dq-frame. A higher-level controller is implemented for speed control, whereby an observer is used to track the electrical angular velocity and rotor position.

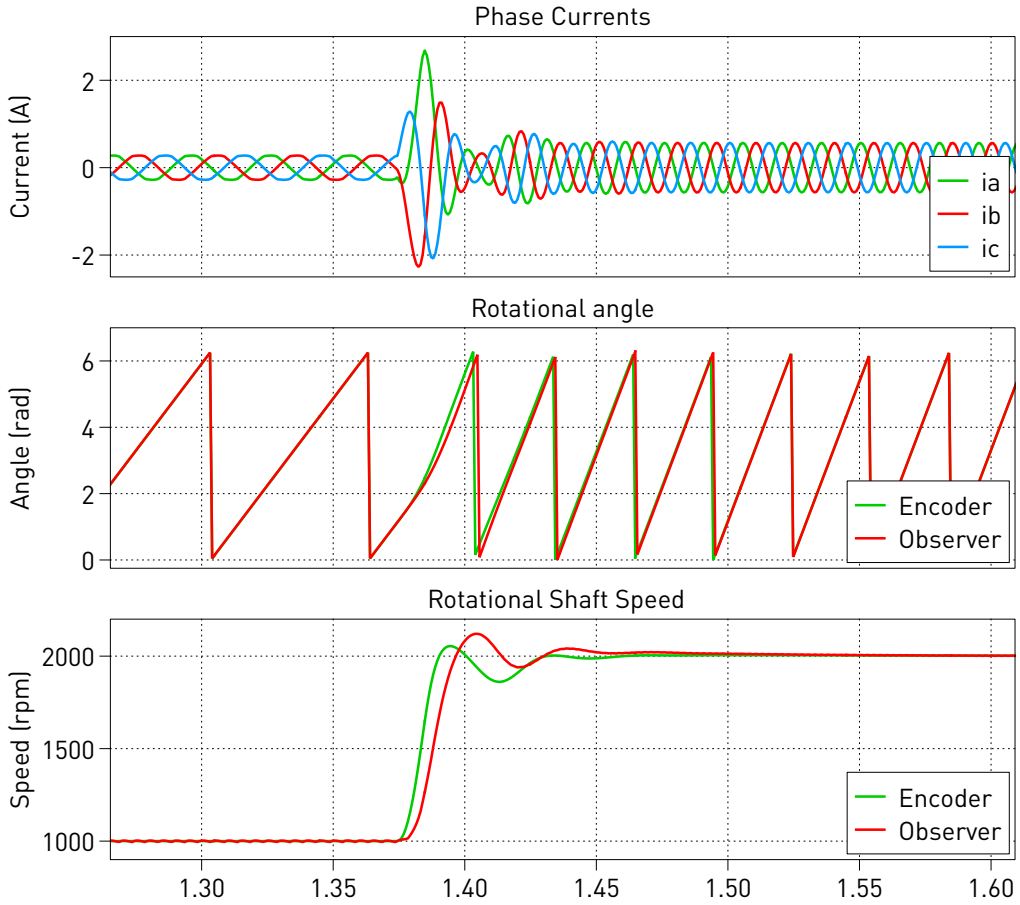


Fig. 9: Transient response of speed controller for a change in speed reference from 1000 RPM to 2000 RPM (offline simulation results).

4 Bibliography

[1] S. Morimoto, K. Kawamoto, M. Sanada and Y. Takeda, “Sensorless control strategy for salient-pole PMSM based on extended EMF in rotating reference frame”, IEEE Trans. Ind. Appl., vol. 38, no. 4, pp. 1054-1061, Jul./Aug. 2002.

[2] R. De Doncker, D. Pulle and A. Veltman, “Advanced electrical drives”, Springer, 2011.

Revision History:

RT Box TSP 2.1.5	First release
RT Box TSP 3.0.1	Use the 3-Phase Index-Based Modulation block to provide more three-phase inverter modulation strategies, also update the single box model to use the multi-tasking feature on RT Box 1
RT Box TSP 3.0.3	Remove the model with two separate RT Boxes and keep only the single RT Box model using multi-tasking feature

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
`https://www.plexim.com`	Web

RT Box Demo Model

© 2002–2026 by Plexim GmbH

The software PLECS described in this document is furnished under a license agreement. The software may be used or copied only under the terms of the license agreement. No part of this manual may be photocopied or reproduced in any form without prior written consent from Plexim GmbH.

PLECS is a registered trademark of Plexim GmbH. MATLAB, Simulink and Simulink Coder are registered trademarks of The MathWorks, Inc. Other product or brand names are trademarks or registered trademarks of their respective holders.