

PLECS

DEMO MODEL

Basic Current Controller Design for Motor Drives

Using the PLECS Analysis Tools

Last updated in PLECS 4.6.1

www.plexim.com

- ▶ Request a PLECS trial license
- ▶ Check the PLECS documentation

1 Overview

This demonstration shows how to obtain the loop-gain $L(s)$ of a current-controlled, three-phase drive system using the PLECS Multitone AnalysisTool. Furthermore, calculations of the PI current controller parameters based on specifications of the crossover frequency and required phase margin are shown in detail.

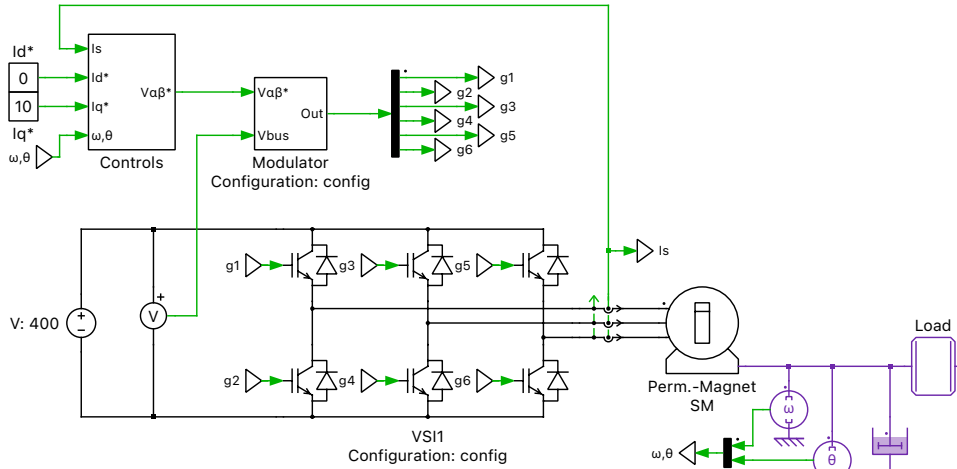


Figure 1: System overview

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

2.1 Plant

The inverter circuit in Fig. 1 is modeled as an ideal three-legged bridge with a stiff 400 V_{dc} supply. The six IGBTs are switched by means of a space-vector modulator to realize the phase voltages commanded by the vector current control. The three-phase currents are measured for closed-loop current control, where a traditional PI controller is used for current regulation in the d- and q-axes. The mechanical speed and angle are measured and scaled by the number of pole-pairs to determine the electrical speed and angle.

Machine Model The AC motor is a synchronous machine with surface-mounted magnets and no saliency ($L_{sd} = L_{sq}$). The phase currents are transformed from the three-phase abc-frame to the two-dimensional rotating reference dq-frame aligned to the rotor position. The transformation into the dq-frame converts an AC system into an equivalent DC system. This enables the use of traditional PI controllers with zero steady-state error. However, the transformation into the dq-frame introduces cross-coupling between the q- and d-axes, which have to be considered in the controller structure.

$$u_{sd} = R_s i_{sd} - \omega_e L_{sd} i_{sq} + L_{sd} \frac{di_{sd}}{dt}$$

$$u_{sq} = R_s i_{sq} + \omega_e L_{sq} i_{sd} + L_{sq} \frac{di_{sq}}{dt} + \omega_e \psi_f$$

The dq-frame model can be described with the equations shown above, where ω_e is the synchronous frequency, and ψ_m is the amplitude of the stator flux linkages due to the permanent magnet. For simplicity it is assumed that the magnet strength is constant and the machine doesn't experience saturation.

2.2 Controller Design

Plant transfer function

Rotor-field oriented control is applied to the drive system, where the stator current is regulated in the rotating reference frame. The d- and q-axis currents are controlled with PI controllers tuned in order to respect a specified crossover frequency and phase margin, which is explained hereafter. By using decoupling and feed-forward terms in the control structure the plant for the current controller operating in the rotating reference frame design is simplified significantly and can be written as:

$$P(s) = \frac{i_{sd}}{u_d^*} = \frac{1}{R_s + sL_{sd}}$$

The transfer function for the q-axis is identical.

Time delays

Digital control always introduces time delays, mainly due to PWM generation, ADC sampling and the calculation time of the control algorithm on the microcontroller. The total time delay will reduce the achievable control bandwidth and also the closed-loop system stability. Expected time delays should be respected during the controller design process, especially if the tuning-method relies on a mathematical model of the plant to derive the controller parameters. For this demo, two different sources of time delay are taken into account:

- The finite computation time of the control algorithm on the microcontroller: $T_{\text{calc}} = T_{\text{sw}}/2$
- The average time delay introduced by the pulse-width modulator operating in single-update mode: $T_{\text{PWM}} = T_{\text{sw}}/2$

The equivalent total delay T_d therefore sums up to one control calculation time step:

$$T_d = T_{\text{calc}} + T_{\text{PWM}} = T_{\text{sw}}$$

Approximating this time delay as a rational transfer function simplifies the controller design. For this, the Padé approximation can be used. In this example, the time delay is approximated by a second-order Padé approximation. The transfer function has the following form, where the coefficients are calculated in Octave or MATLAB by using the function `padecoef(T_d , 2)`:

$$D_\Sigma(s) = \frac{1 - \frac{T_d}{2}s + \frac{T_d^2}{12}s^2}{1 + \frac{T_d}{2}s + \frac{T_d^2}{12}s^2}$$

Controller design

The continuous time transfer function of a PI controller in parallel form can be written as:

$$C(s) = k_p + \frac{k_i}{s}$$

The open-loop transfer function $L(s)$ is a multiplication of the plant transfer function $P(s)$, the PI controller transfer function $C(s)$ and the time delay approximation $D_\Sigma(s)$:

$$L(s) = P(s) \cdot D_\Sigma(s) \cdot C(s)$$

Hereafter, the plant and the equivalent time delay transfer functions are summarized as a single transfer function:

$$P_D(s) = P(s) \cdot D_\Sigma(s).$$

The main idea of the “Closed-Form Crossover Specification” method is to calculate the PI controller parameters k_p , k_i such that the loop gain has the specified crossover frequency ω_c and phase margin φ_m . In order to obtain meaningful controller parameters the specifications must be compatible with the constraints of the control system and the plant. If this is not the case, the closed-form calculation can even lead to an unstable system behavior.

At the desired crossover frequency ω_c , the magnitude r_P and the phase φ_P of the plant are defined as:

$$P_D(j\omega_c) = r_P \cdot e^{j\varphi_P},$$

and can be calculated by using the Octave function $[r_P, \varphi_P] = \text{bode}(P_D, \omega_c)$.

Using the calculated parameters r_P , φ_P from the plant transfer function and the two design specifications ω_c , φ_m , the PI controller parameters k_p and k_i can be calculated in a closed form by using the following two equations:

$$k_p = \frac{-1}{r_P} \cdot \cos(\varphi_m - \varphi_P)$$

$$k_i = k_p \cdot (-\tan(\varphi_m - \varphi_P) \cdot \omega_c)$$

Fig. 2 shows the loop-gain $L(s)$ with the designed control parameters. The controller bandwidth and phase-margin requirements are achieved.

3 Simulation

PLECS allows one to measure the frequency response of a physical system to a small perturbation signal. A small-signal analysis of three-phase systems can be performed by first converting the three-phase AC values into the rotating reference frame. This transformation allows the configuration of a small-signal analysis around the steady-state operating point.

Note In this example it is assumed that the controller decoupling is achieved and the system is decoupled in the dq-frame. If this is not the case the plant behaves as a Multiple-Input/Multiple-Output (MIMO) system. Such a system requires the use of more sophisticated multivariable controllers. The frequency analysis is limited to the q-axis but to obtain a complete picture it is important to observe the effect of perturbation of the q-axis system on both the q- and d-axis systems (and vice versa).

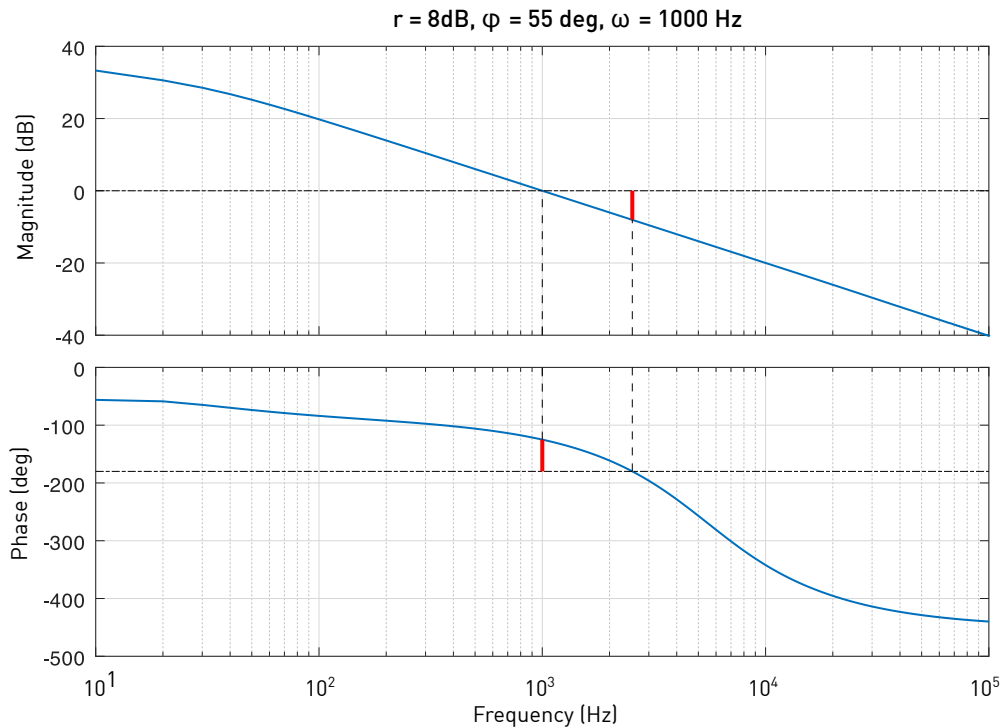


Figure 2: Loop-Gain transfer function analytically calculated for $\omega_c = 1000\text{Hz}$ and $\varphi_m = 55^\circ$

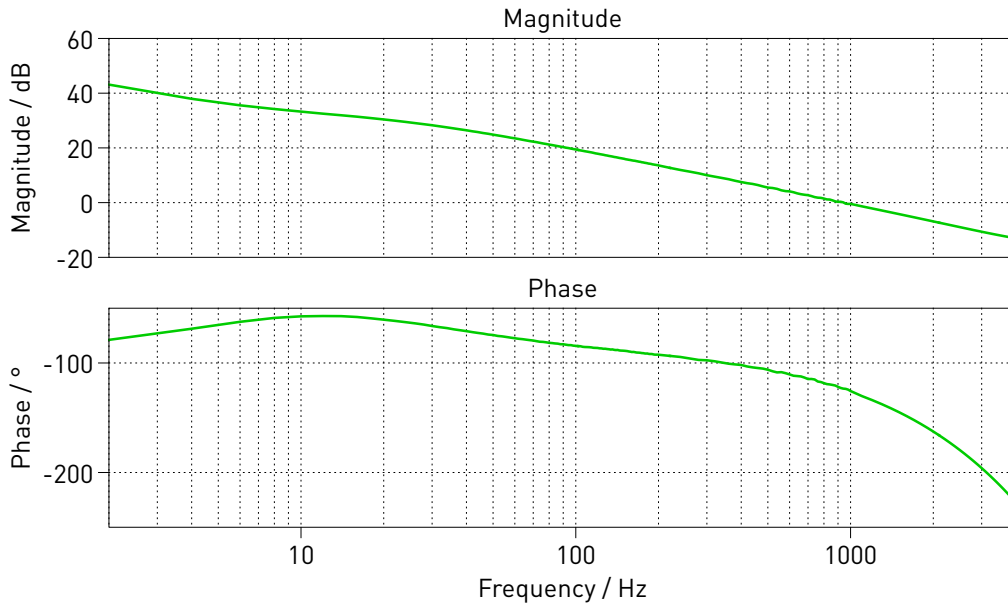


Figure 3: Loop-Gain transfer function measured on the physical system model in PLECS

3.1 Loop Gain

To run an analysis, choose **Analysis tools...** from the **Simulation** menu, then select the “Loop Gain” analysis from the list and click **Start analysis**.

In this simulation, a loop-gain meter is used on the q-axis current to obtain the open-loop transfer function of the controlled motor drive system. From the graph it can be seen that the system exhibits the characteristics of an integrator near DC, thus eliminating any steady-state error between the reference and the controlled value. It can be verified that the design specifications for controller band-

width ($\omega_c = 1000$ Hz) and phase-margin ($\varphi_m = 55^\circ$) are met. The slope of the system magnitude is -20 dB/decade around the cross-over frequency and the gain-margin is 8 dB. Thus, the system is stable. The frequency response measured in PLECS shows a good agreement with the frequency response of the analytical calculation shown in Fig. 2.

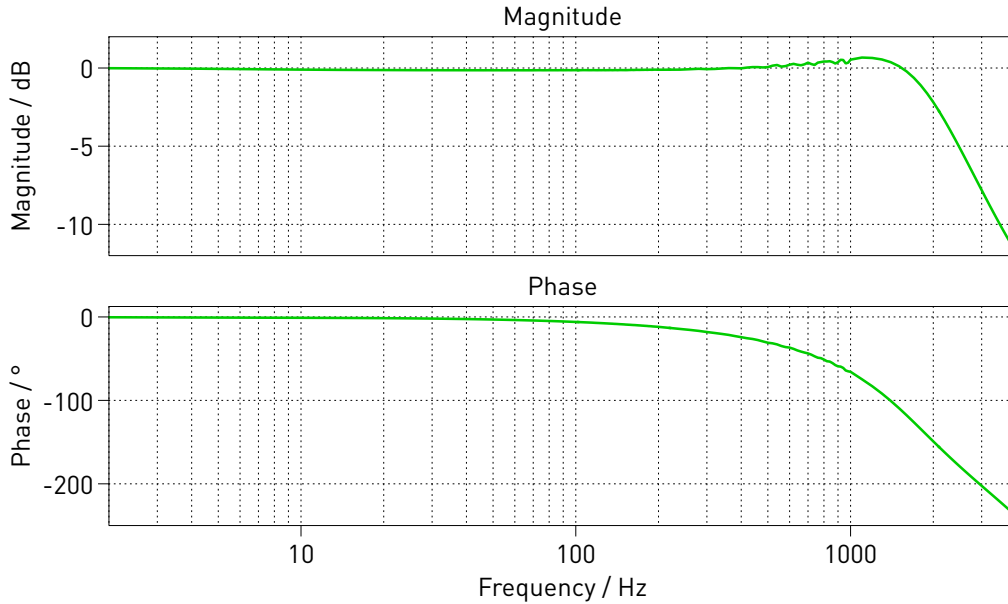


Figure 4: Reference-to-output transfer function measured on the physical system model in PLECS

3.2 Reference-to-Output Transfer Function

The frequency analysis for a reference-to-output transfer function, where the plant is the motor drive system, can be obtained by perturbing the d- or q-axis setpoints and observing the effects on the corresponding d- or q-axis quantity. In this simulation, the q-axis current setpoint is perturbed and the effects of the perturbation are observed in the q-axis current. The frequency analysis shows the characteristics of a low pass filter as is expected based on the open-loop transfer function.

3.3 Limitations

The transfer functions in the simulations above only show the behavior up to half of the switching frequency. This is because the Delay block representing the controller calculation time, the modulator and the switching network are non-linear elements. The modulator becomes non linear for perturbation frequencies above $f_{sw}/2$. In order to get corresponding results from the analytical calculation and the PLECS Analysis Tool at frequencies higher than $f_{sw}/2$ these elements need to be linearized manually.

This linearization is implemented in this demo model. The initialization script features a parameter config that can take the value 'switched' or 'averaged'. Using the average configuration the results shown in Fig. 5 can be obtained.

4 Conclusion

This example shows how the PLECS Analysis Tools can be used to measure a transfer function directly in a model of a physical system. In addition, an analytical controller design based on specifications in the frequency domain is presented.

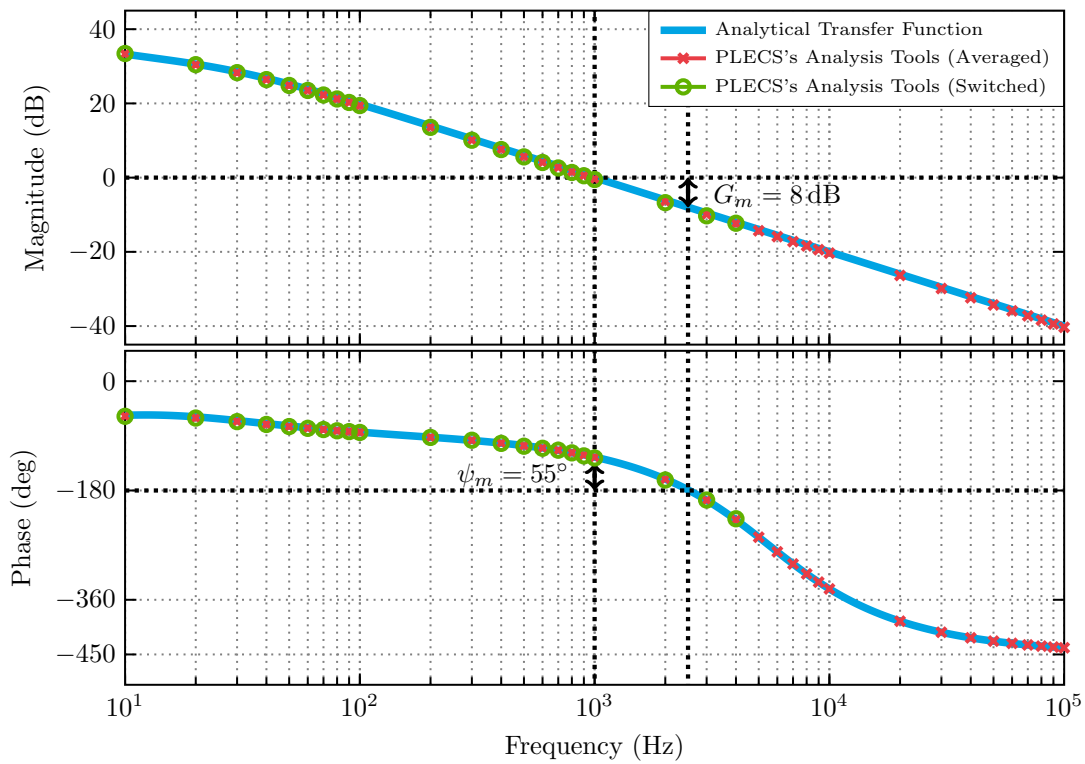


Figure 5: Loop gain calculation using a completely linearized system

References

- [1] Guzzella, L.. “Analysis and Synthesis of Single-Input/Single-Output Control Systems.” (2019).

Revision History:

PLECS 4.6.1 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

PLECS Demo Model

© 2002–2021 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.