smart ctrl

control design for power electronics

Design of control loops for a Permanent Magnet Synchronous Machine

Tutorial – October 2025 -

How to Contact:

sales@powersmartcontrol.com

www.powersmartcontrol.com

Notice

Power Smart Control tutorials or other design advice, services or information, including, but not limited to, reference designs, are intended to assist designers who are developing applications that use SmartCtrl; by downloading, accessing or using any particular Power Smart Control resource in any way, you (individually or, if you are acting on behalf of a company, your company) agree to use it solely for this purpose and subject to the terms of this notice.

Power Smart Control reserves the right to make corrections, enhancements, improvements and other changes to its resources.

You understand and agree that you remain responsible for using your independent analysis, evaluation and judgment in designing your applications and that you have full and exclusive responsibility to assure the safety of your applications and compliance of your applications with all applicable regulations, laws and other applicable requirements.

Disclaimer

Power Smart Control S.L. (PSC) makes no representation or warranty with respect to the adequacy or accuracy of this documentation or the software which it describes. In no event will PSC or its direct or indirect suppliers be liable for any damages whatsoever including, but not limited to, direct, indirect, incidental, or consequential damages of any character including, without limitation, loss of business profits, data, business information, or any and all other commercial damages or losses, or for any damages in excess of the list price for the license to the software and documentation.

Power Smart Control SL

General index

Figure index

1. Introduction

Motor control is essential in a wide variety of applications, covering sectors such as automotive, industrial, aerospace, and renewable energy, among many others.

In general, the structure of this type of control is based on a double-loop scheme in the dq reference frame: the inner loop regulates the current or torque, while the outer loop controls the speed.

This tutorial is intended to guide you step by step, to design the outer and inner control loop in dq axis of a Three Phase Inverter that feeds a Permanent Magnet Synchronous Machine.

Figure 1 shows the power stage of the converter.

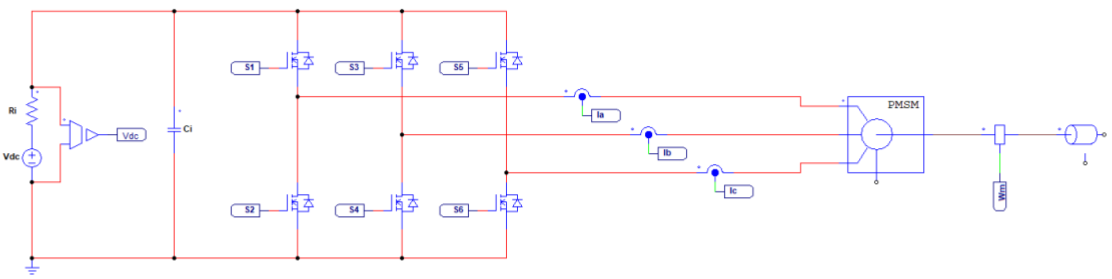

Figure 1: Power Stage

The SmartCtrl equation editor will be used to tune the PI controllers of the control loops. For this purpose, the dq model of the Permanent Magnet Synchronous Machine shown in Figure 2 will be used.

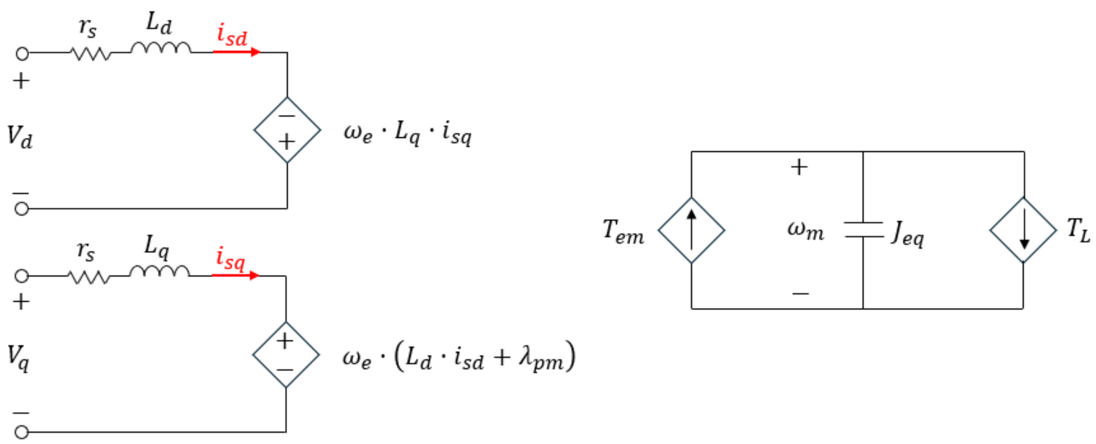

Figure 2: DQ model of the Permanent Magnet Synchronous Machine

The term ω_m represents the mechanical speed of the rotor and the term ω_e represents the electrical speed. The electrical speed, ω_e, can be estimated using the expression shown in (1).

$$\omega_e = \frac{P}{2} \cdot \omega_m \qquad (1)$$

Where P represents the number of poles of the machine.

On the other hand, torque is defined by (2).

$$T_{em} = \frac{3}{2} \cdot \frac{P}{2} \cdot \left[(L_d - L_q) \cdot i_{sd} \cdot i_{sq} + \lambda_{pm} \cdot i_{sq} \right] \qquad (2)$$

Where the term λ_{pm} represents the back EMF constant. The terms i_{sd} and i_{sq} are the stator currents. The terms L_d and L_q are the dq inductances.

The control structure is shown in Figure 3.

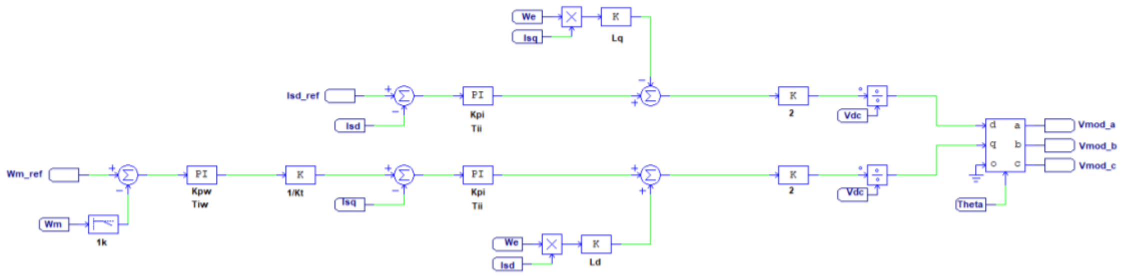

Figure 3: Control structure of the Permanent Magnet Synchronous Machine

In this control strategy, the current reference for the d-axis is set to 0A. This allows the torque to depend solely on the current of the q-axis. See expression (3).

$$T_{em} = \frac{3}{2} \cdot \frac{P}{2} \cdot \lambda_{pm} \cdot i_{sq} \qquad (3)$$

The torque constant, K_T, is calculated by (4)

$$K_T = \frac{3}{2} \cdot \frac{P}{2} \cdot \lambda_{pm} \qquad (4)$$

The angle Theta used in Park transforms is estimated using the scheme shown in Figure 4.

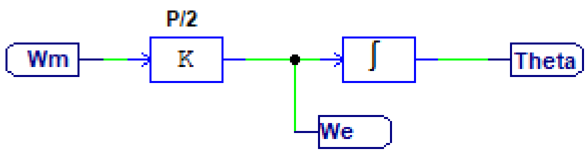

Figure 4: Estimation of the angle theta

In this tutorial, the modulation to be used is PWM, as shown in Figure 5.

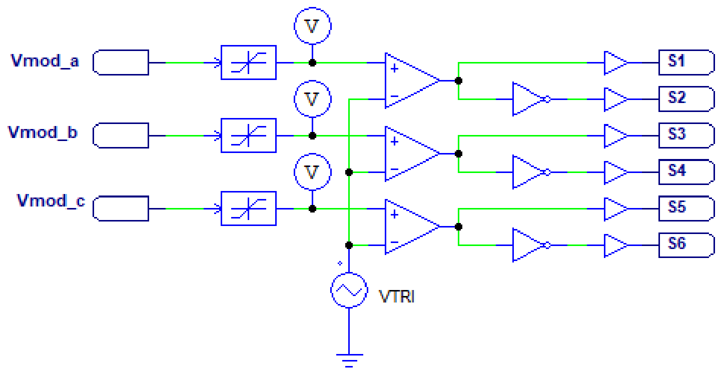

Figure 5: Pulse Width Modulation (PWM)

2. Current loop design

1. Open your SmartCtrl Software.
2. To begin the design, click on *Equation editor.* See Figure 6.

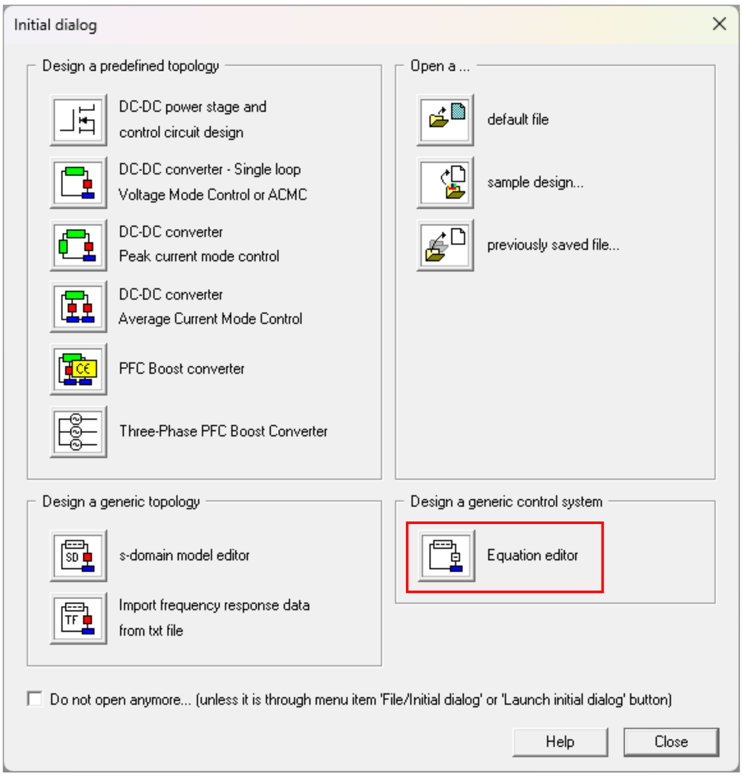

Figure 6: SmartCtrl initial window

3. The transfer function of the current plant is defined. See Figure 7.

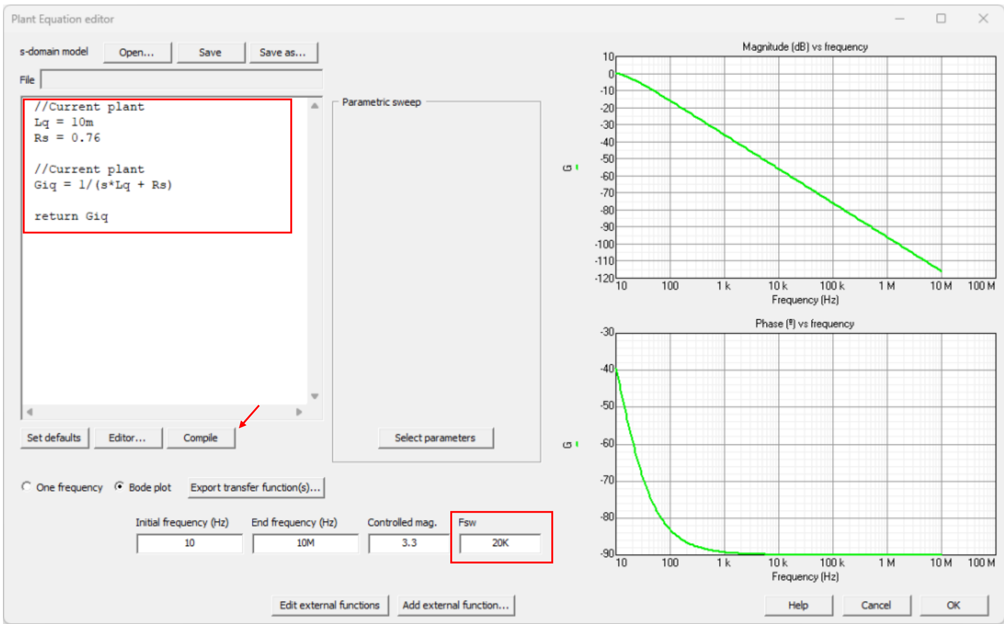

Figure 7: Equation editor for current plant

The current plant is defined by (5).

$$G_{iq} = \frac{1}{L_q \cdot s + R_s} \tag{5}$$

The main parameters of the Permanent Magnet Synchronous Machine to be used in this example are shown in Table 1.

Stator resistance (R_s)	$0.76\ \Omega$
d-axis inductance (L_d)	$8\ mH$
q-axis inductance (L_q)	$10\ mH$
Peak line-to-line back emf constant ($V_{pk}/krpm$)	$92.86\ V/krpm$
Number of poles (P)	4
Moment of Inertia (J)	$0.025\ kg \cdot m^2$

Table 1: Permanent Magnet Synchronous Machine parameters

Table 2 shows the parameters of the three-phase inverter.

DC voltage (V_{DC})	$450\ V$
Input resistance (R_i)	$10\ m\Omega$
Input capacitance (C_i)	$470\ uF$
Switching frequency	$20\ kHz$

Table 2: Three-phase inverter parameters

After defining the plant and clicking compile, click OK.

4. The current sensing stage plant is defined as shown in Figure 8.

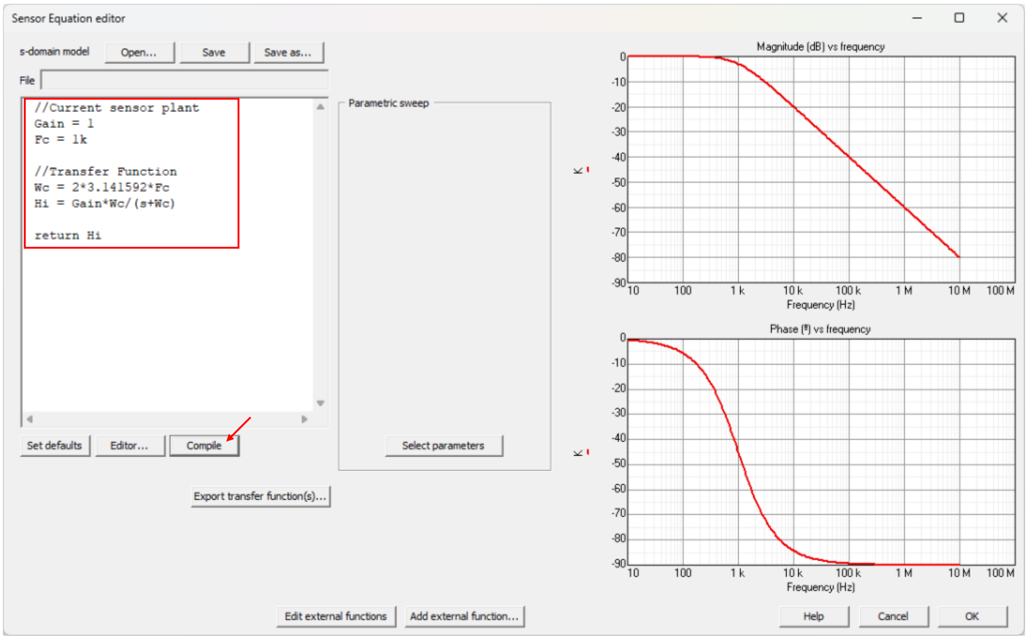

Figure 8: Equation editor for current sensor

The current sensor plant is defined by (6).

$$H_i = Gain \cdot \frac{W_c}{s + W_c} \qquad (6)$$

The cutoff frequency of the low-pass filter is equal to 1kHz.

After defining the plant and clicking compile, click OK.

5. The PI compensator is selected, as shown in Figure 9.

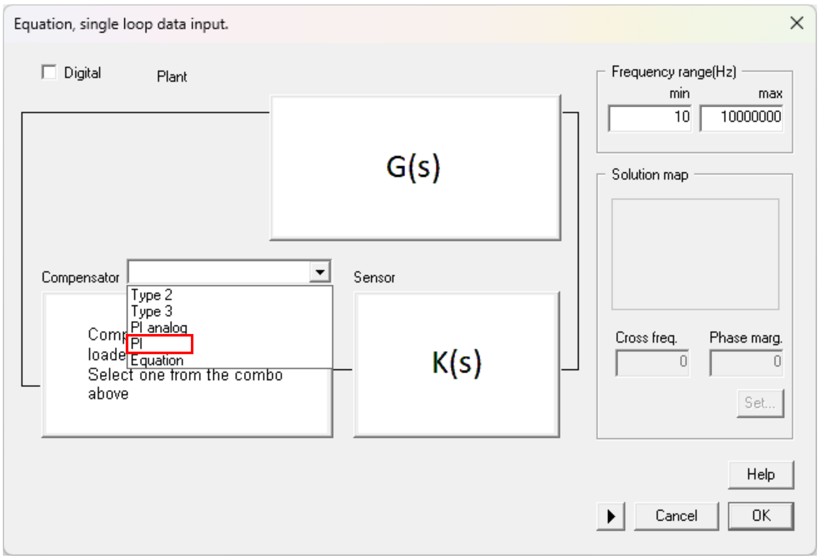

Figure 9: Compensator selection for current loop

The modulator gain is set to 1. See Figure 10.

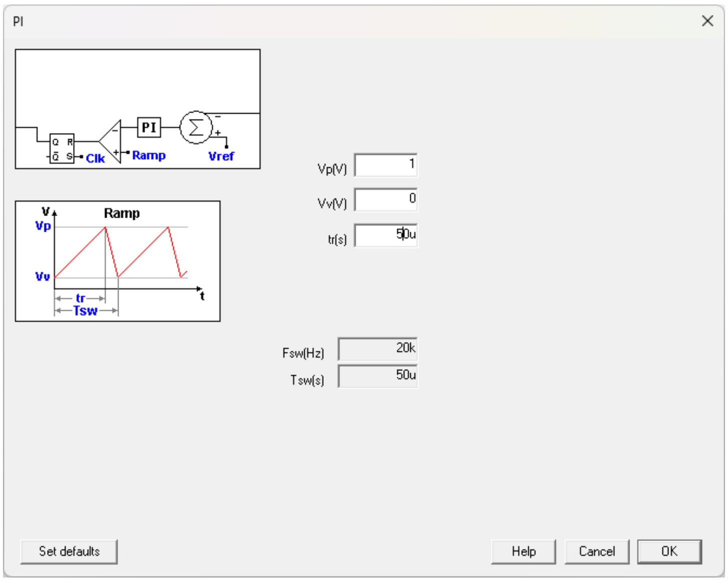

Figure 10: Modulator gain

In SmartCtrl, the modulator gain is calculated using expression (7)

$$G_{mod} = \frac{1}{V_p - V_v} \cdot \frac{t_r}{T_{sw}} \qquad (7)$$

Where V_p represents the peak value of the carrier signal, V_v is the valley value of the carrier signal, and t_r is the rise time of the carrier signal.

6. A crossover frequency and phase margin are selected from the solution map. See Figure 11. In this example, an Fc of 300Hz and a PM of 60º are considered.

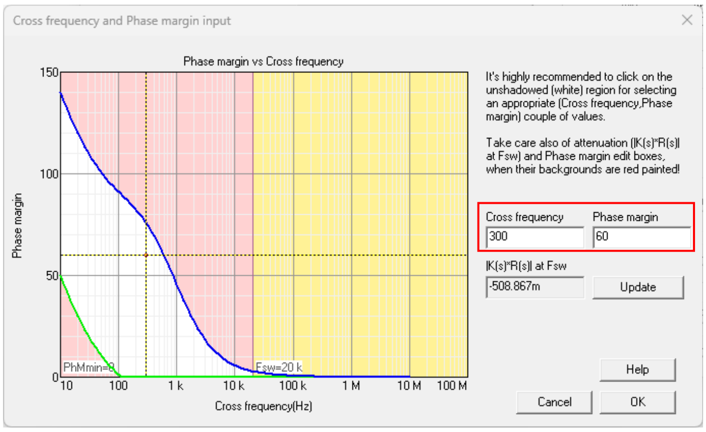

Figure 11: Solution map of current loop

7. Finally, the Kp and Ti values of the PI controller will be copied to PSIM simulation. In this example, the value of Kp is equal to 18.966 and Ti is equal to 1.896m.

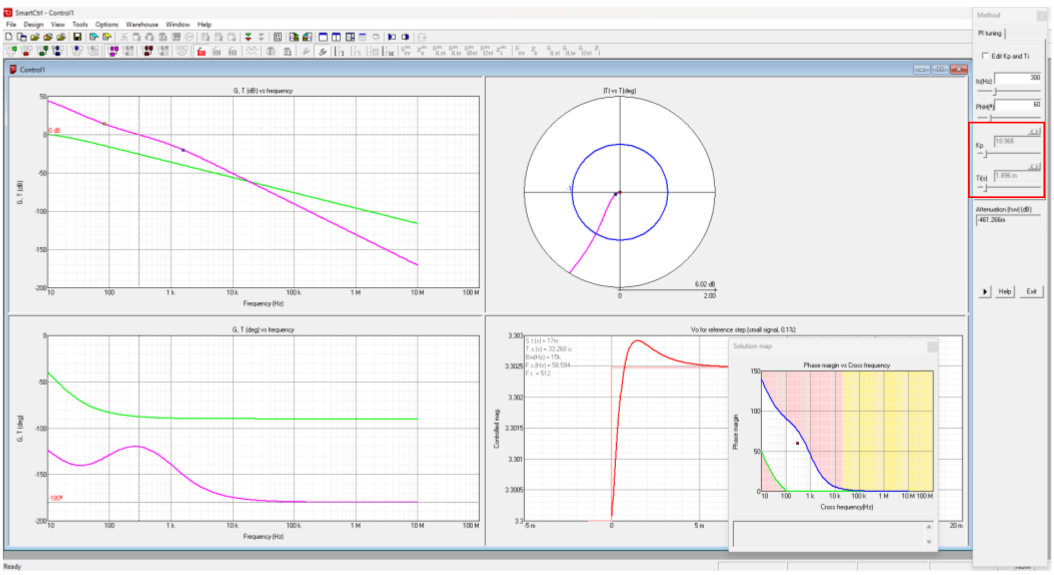

Figure 12: PI parameters of the current loop

3. Speed loop design

1. To design the speed loop, open a new project in SmartCtrl. To do this, go to the initial dialog option in the File menu. See Figure 13.

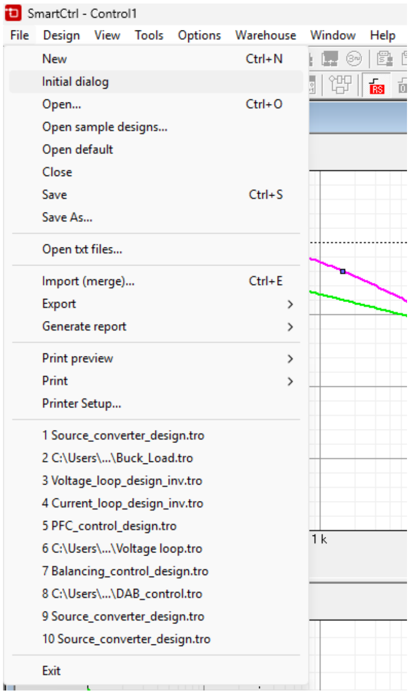

Figure 13: Access to the initial dialog

2. Select the equation editor option again. See Figure 6.
3. The speed plant is defined in the equation editor. See Figure 14.

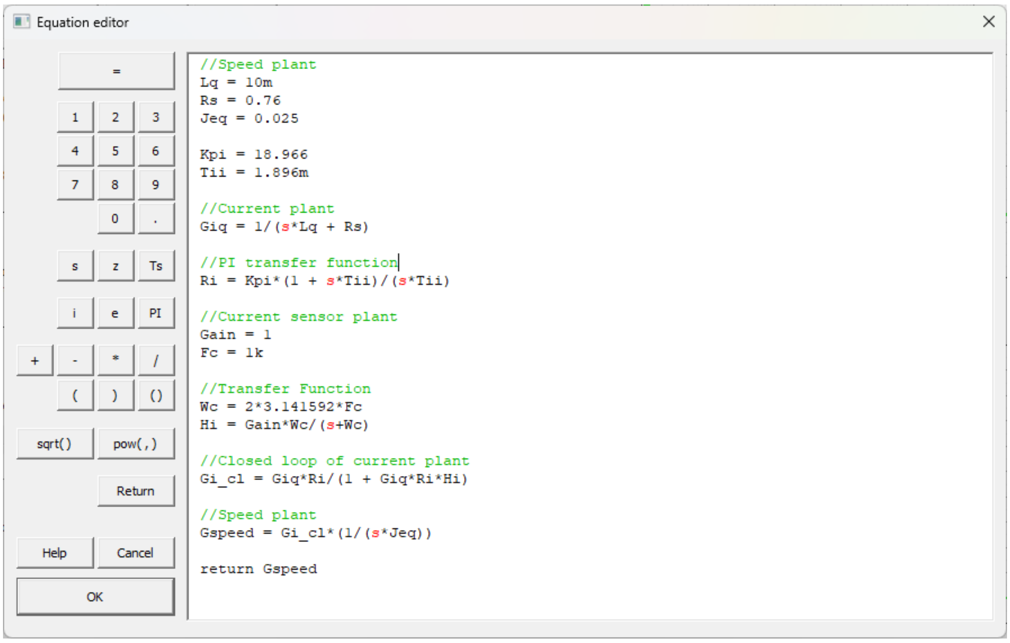

Figure 14: Speed plant transfer function

As can be seen in Figure 14, the speed plant depends on the internal closed-loop current.

Figure 15 shows the equivalent block diagram of the control structure.

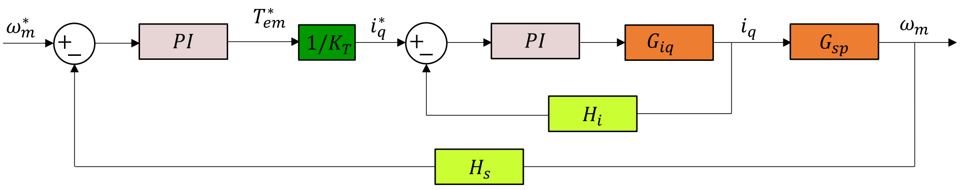

Figure 15: Block diagram of the control structure

After defining the plant and clicking compile, click OK.

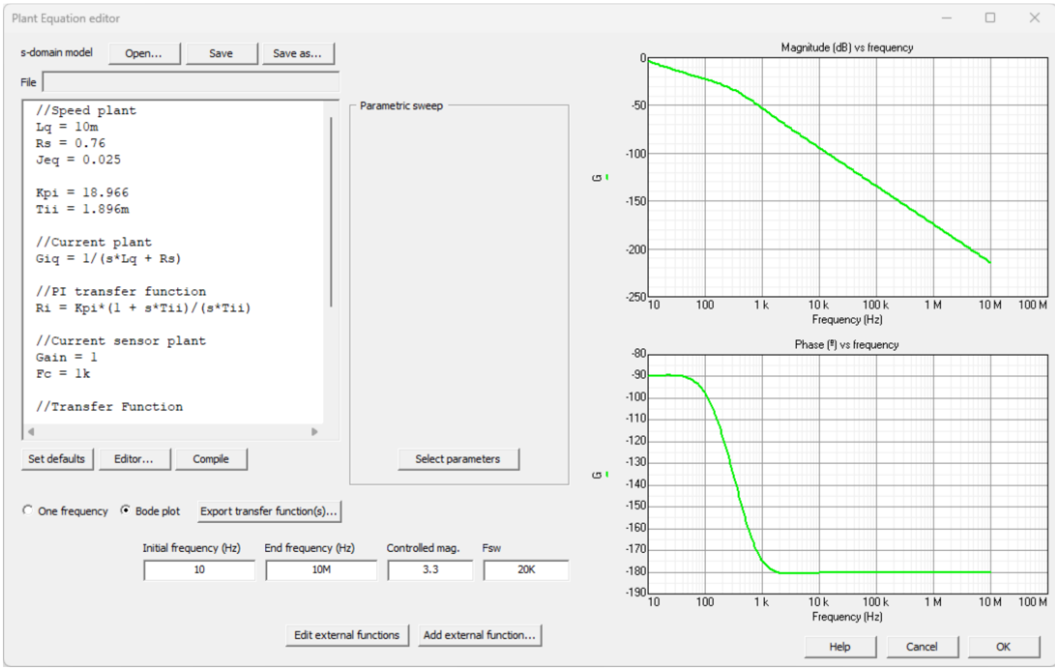

Figure 16: Speed plant in equation editor

4. The speed sensing stage plant is defined as shown in Figure 17.

The speed sensor plant is defined by (8).

$$H_{sp} = Gain \cdot \frac{W_c}{s + W_c} \tag{8}$$

The cutoff frequency of the low-pass filter is equal to 1kHz.

After defining the plant and clicking compile, click OK.

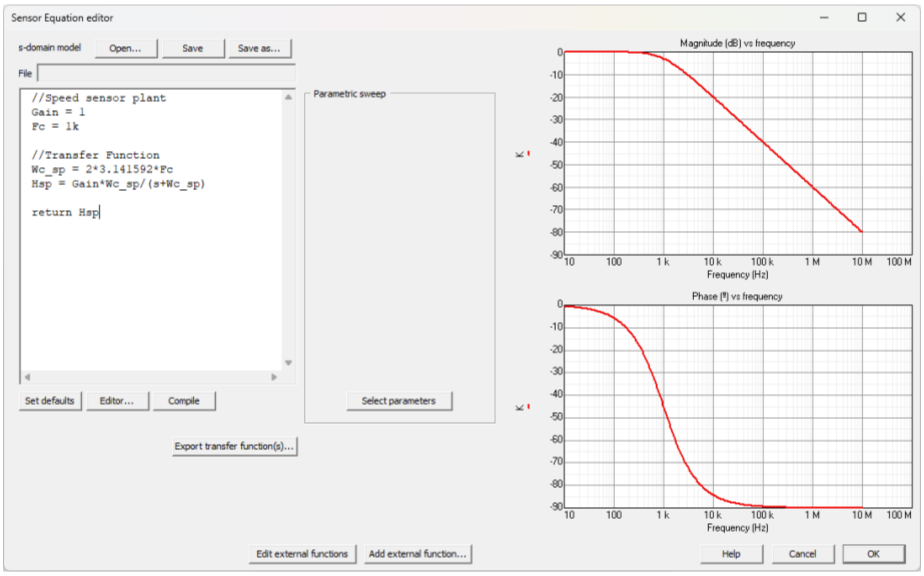

Figure 17: Speed sensor equation editor

5. The PI compensator is selected, and the modulator gain is set to 1. See Figure 18.
   To set the modulator gain equal to 1, see Figure 10.

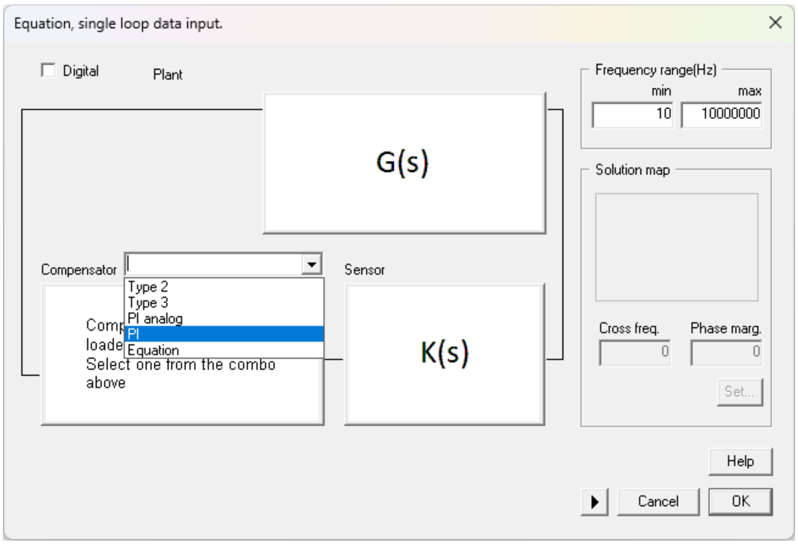

Figure 18: PI compensator for speed loop

6. A crossover frequency and phase margin are selected from the solution map. See Figure 19. In this example, an Fc of 40Hz and a PM of 50º are considered.

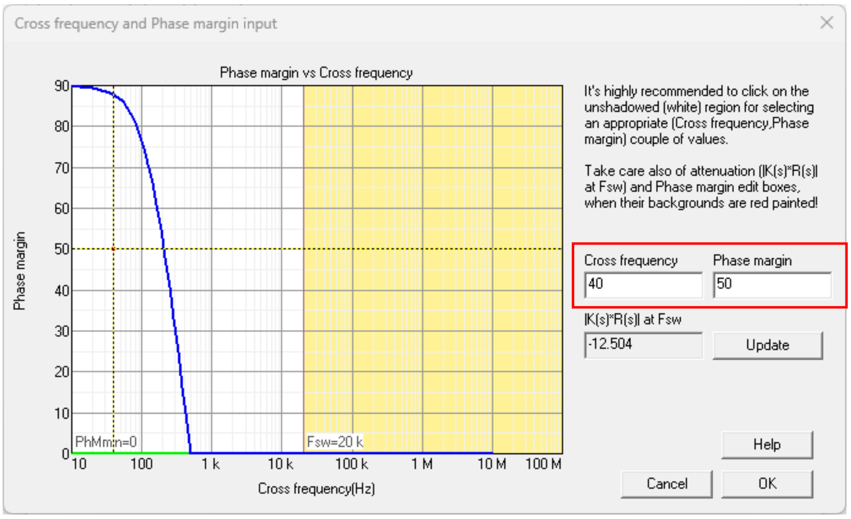

Figure 19: Solution maps for speed loop

7. Finally, the Kp and Ti values of the PI controller will be copied to PSIM simulation. In this example, the value of Kp is equal to 4.748 and Ti is equal to 5.149m.

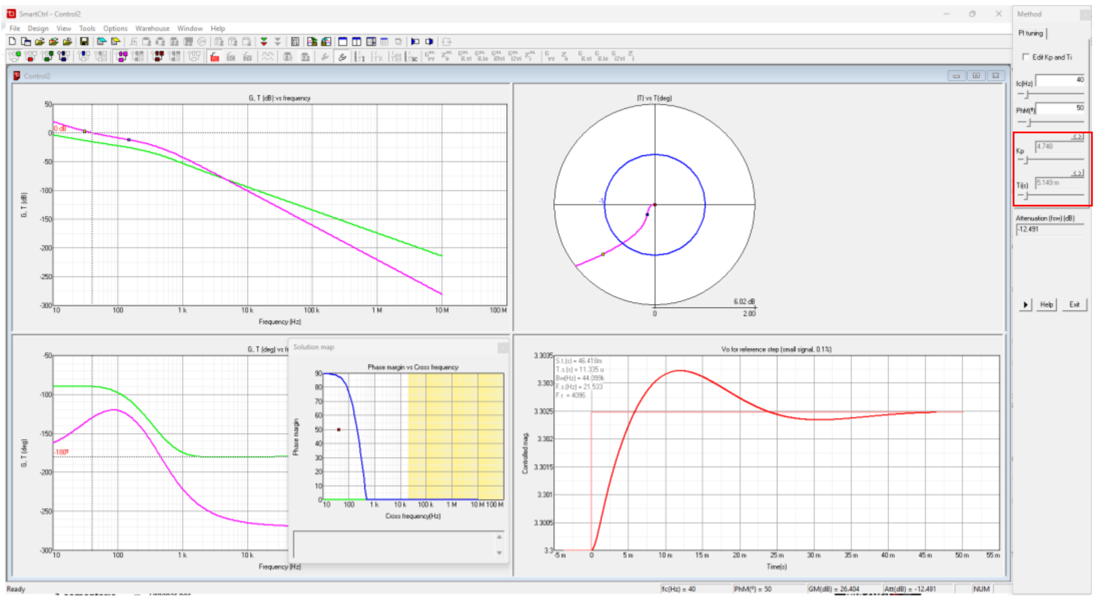

Figure 20: PI parameters of the speed loop

4. PSIM Validation

Figure 21 shows a schematic diagram of the PSIM simulation. In this simulation, the stability of the loop and the transient response are checked when a step is injected into the rotor speed reference and the load torque.

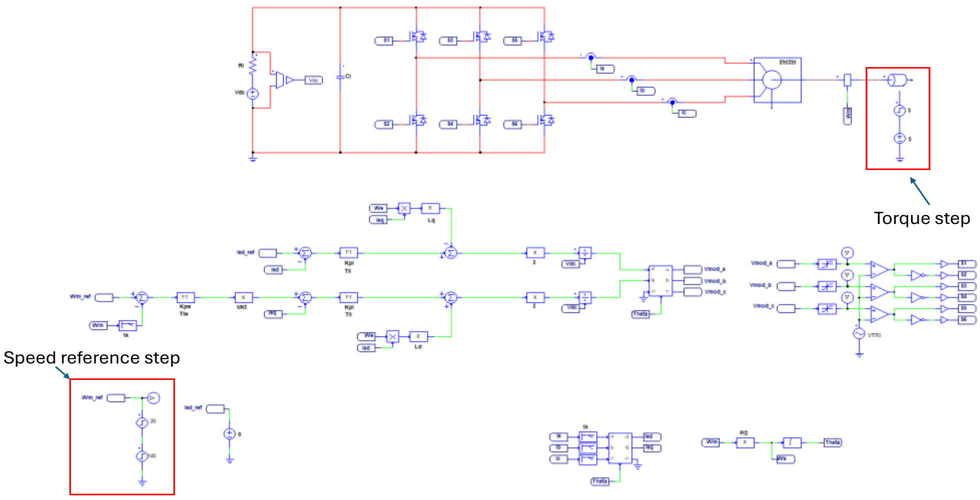

Figure 21: PSIM simulation

The simulation parameters are shown in Figure 22.

```
Vdc = 450
Ri = 10m
Ci = 470u
Fsw = 20k

Rs = 0.76
Ld = 8m
Lq = 10m
Vpk = 92.86
P = 4
J = 0.025

Lambda = 60*Vpk/(sqrt(3)*3.141592*P*1000)

Kt =(3/2)*(P/2)*Lambda

Kpi = 18.966
Tii = 1.896m

Kpw = 4.748
Tiw = 5.149m
```

Figure 22: PSIM parameter file

As can be seen in Figure 23, the control loops designed with SmartCtrl are stable.

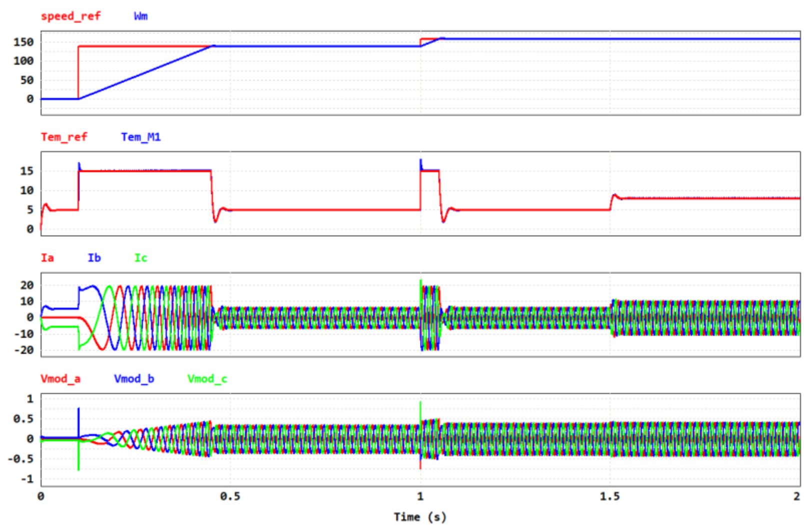

Figure 23: Waveforms of the transient response of the loops

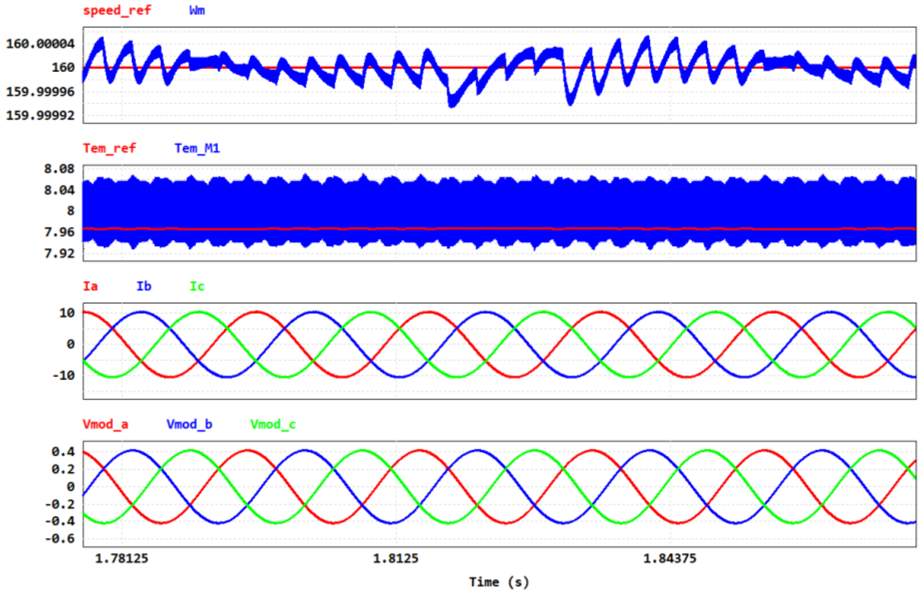

Figure 24: Main Waveforms of PSIM simulation