Designing a Digital Control in z-domain

Tutorial –December 2018-

How to Contact:

info@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 licence to the software and documentation.

The software SmartCtrl© used in this tutorial is furnished under a license agreement. The software may be used only under the terms of the license agreement.

Power Smart Control SL

General index

Figure index

1. Introduction

This tutorial is intended to show how SmartCtrl can be applied to design a digital control using the equation editor function. In this case, a typical buck converter in voltage mode will serve as an example to demonstrate the SmartCtrl capabilities to design a digital control considering the sampling, modulator effects and delays.

Along the tutorial, several aspects will be highlighted:

- A brief review of the main considerations to take into account when designing the digital control. A small signal discrete-time model of buck converter is shown.
- How to use the new features that the "Equation Editor" module offers to enter the transfer functions such as the plant, sensor and compensator in discrete-time.
- The procedure to achieve a correct design of the compensator.

The tutorial is structured as follows: Firstly, a brief theoretical introduction, regarding a small signal discrete-time model of buck converter in voltage mode, is provided.

Then the design flow is made for the case in which it starts from a discrete plant.

Finally, a PSIM simulation is provided to verify the response of the control loop.

2. Small signal discrete-time of buck converter in voltage-mode

The proposed control structure for digital control of buck converter is shown in Figure 1.

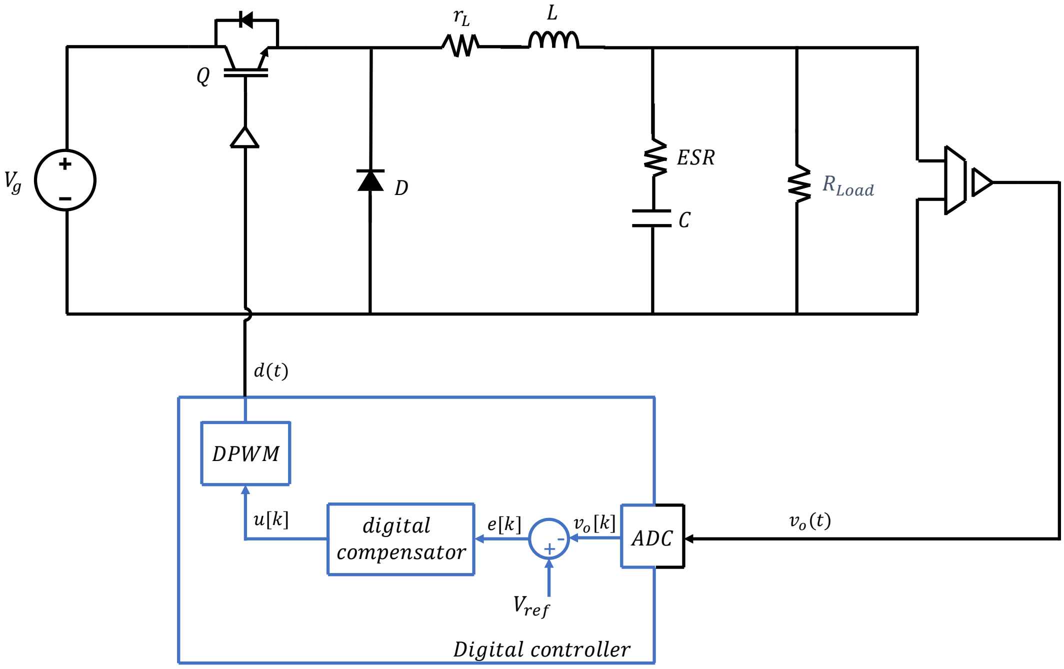

Figure 1: Buck converter: digital voltage-mode control

The main specifications are the following:

$V_g = 12\ V$, DC input voltage.

$V_o = 3.3\ V$, output voltage.

$F_{sw} = 250\ kHz$, switching frequency.

$I_o = 4.125\ A$, output current.

The output LC filter components are:

$r_L = 100\ \mu\Omega$. (Inductor equivalent series resistance)

$L = 30\ \mu H$.

$ESR = 30\ m\Omega$. (Capacitor equivalent series resistance)

$C = 160\ \mu F$.

The voltage sensor parameters are:

$K_s = 1$, gain.

$HF_{pole} = 25\ kHz$, pole frequency.

The first step is to define the state-space representation.

$$\begin{bmatrix} \frac{di_L}{dt} \\ \frac{dv_c}{dt} \end{bmatrix} = A_x \cdot \begin{bmatrix} i_L(t) \\ v_c(t) \end{bmatrix} + B_x \begin{bmatrix} V_g \\ I_o \end{bmatrix}$$

$$\begin{bmatrix} i_L(t) \\ v_o(t) \end{bmatrix} = C_x \cdot \begin{bmatrix} i_L(t) \\ v_c(t) \end{bmatrix} + E_x \begin{bmatrix} V_g \\ I_o \end{bmatrix}$$

When the subscript "x" is equal to 1, it represents the state "on" and 0 represents the state "off", therefore you get:

$$A_1 = A_0 = \begin{bmatrix} -\frac{r_L + ESR}{L} & -\frac{1}{L} \\ \frac{1}{C} & 0 \end{bmatrix}$$

$$B_1 = \begin{bmatrix} \frac{1}{L} & \frac{ESR}{L} \\ 0 & -\frac{1}{C} \end{bmatrix}$$

$$B_0 = \begin{bmatrix} 0 & \frac{ESR}{L} \\ 0 & -\frac{1}{C} \end{bmatrix}$$

$$C_1 = C_0 = \begin{bmatrix} 1 & 0 \\ ESR & 1 \end{bmatrix}$$

$$E_1 = E_0 = \begin{bmatrix} 0 & 0 \\ 0 & -ESR \end{bmatrix}$$

The following discrete model considers the digital delay and type of modulation. In this example the sampling frequency F_{samp} is equal to switching frequency F_{sw}.

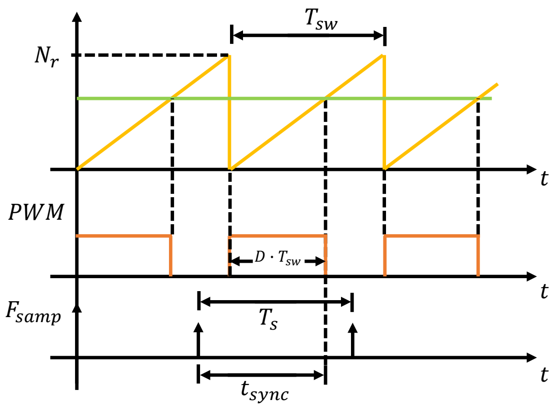

Figure 2: Trailing-Edge Modulation

Where:

t_{sync}: It accounts for the time difference between the moment when a signal is sampled and when it is used to update the compensator output.

N_r: represent the number of steps.

The following expressions let us to construct the discrete small-signal matrices:

$$\Phi = e^{A_o \cdot (T_s - t_{sync})} \cdot e^{A_1 \cdot D \cdot T_{sw}} \cdot e^{A_0 (t_{sync} - D \cdot T_{sw})}$$

$$\Upsilon = \frac{T_s}{N_r} \cdot e^{A_o \cdot (T_s - t_{sync})} \cdot F_\downarrow$$

$$\delta = C_0$$

$$F_\downarrow \triangleq (A_1 \cdot X_\downarrow + B_1 \cdot V) - (A_o \cdot X_\downarrow + B_0 \cdot V)$$

$$X_\downarrow = \left(I - e^{A_1 \cdot D \cdot T_{sw}} \cdot e^{A_0 \cdot D' \cdot T_{sw}}\right)^{-1} \cdot \left[-e^{A_1 \cdot D \cdot T_{sw}} \cdot A_0^{-1} \cdot \left(I - e^{A_0 \cdot D' \cdot T_{sw}}\right) \cdot B_0 - A_1^{-1} \cdot \left(I - e^{A_1 \cdot D \cdot T_{sw}}\right) \cdot B_1\right] \cdot V$$

In order to operate with the above expressions, it is necessary to use some mathematical calculations software. As the controller is going to be designed using the SmartCtrl's "equation editor", it is not necessary to enter the digital delay in the plant model, since SmartCtrl allows entering the delay later. Therefore, we consider a $t_{sync} = 0$ and $N_r = 1$. As a result, the following discrete transfer function is obtained:

$$G_{vu}(z) = \frac{0.08763 \cdot z - 0.04781}{z^2 - 1.993 \cdot z + 0.996}$$

On the other hand, the transfer function in the frequency domain using the standard averaged modeling approach is given by:

$$G_{vd}(s) = \frac{R_{Load} \cdot V_g \cdot (C \cdot ESR \cdot s + 1)}{[C \cdot L \cdot R_{Load} + C \cdot L \cdot ESR] \cdot s^2 + [L + C \cdot r_L \cdot R_{Load} + C \cdot ESR \cdot r_L + C \cdot ESR \cdot R_{Load}] \cdot s + r_L + R_{Load}}$$

Now we compare both transfer functions. To graph two transfer functions at the same time, go to menu tools then select "Equation editor" option.

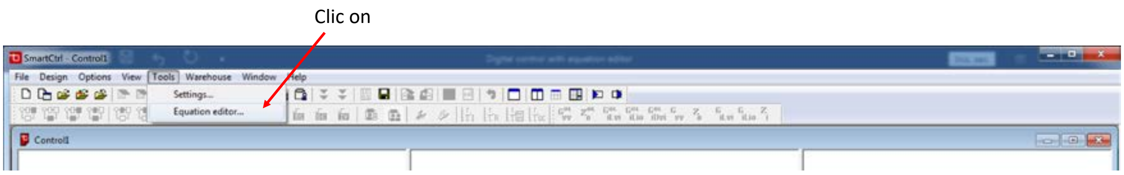

Figure 3: Equation editor option

Then enter both transfer functions.

Important: In order to use the z-transform in the equation editor, you must enter the sampling period (T_s). The script of both transfer function is in Figure 4.

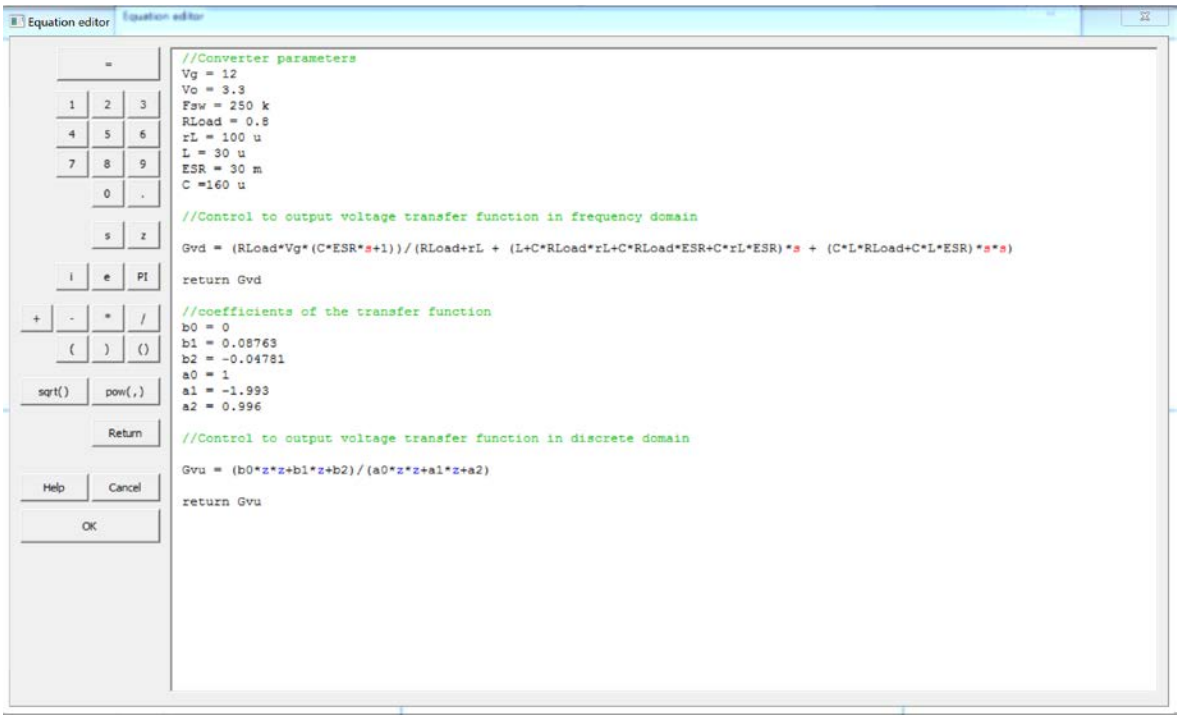

Figure 4: z-transform in equation editor

Clicking on the “Compile” button opens a dialog box in which you can enter the sampling period (T_s).

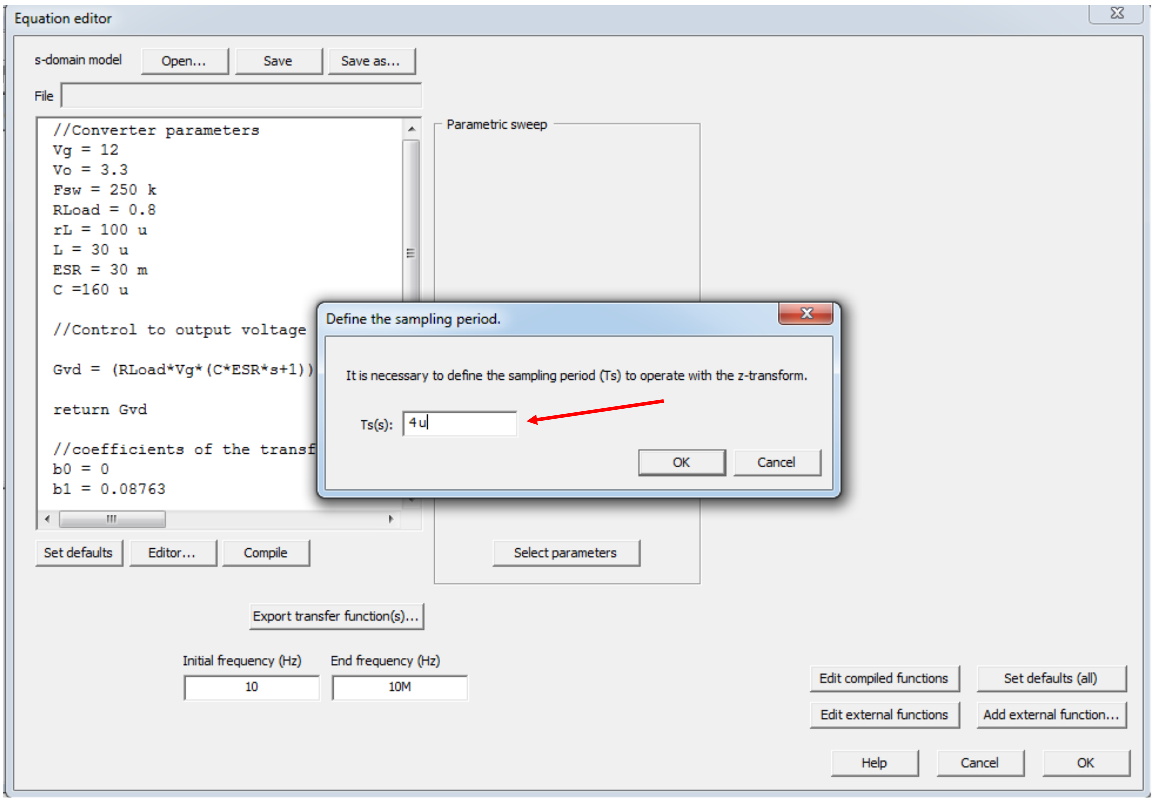

Figure 5: Defining the sampling period

Finally, click on the "OK" button. In Figure 6 it can be observed the phase of the transfer function that is in discrete-time (G_{vu}) falls more sharply than the transfer function that is defined in frequency domain (G_{vd}). Therefore, for this example we will use the discrete transfer function in order to design the compensator because it is the most restrictive situation.

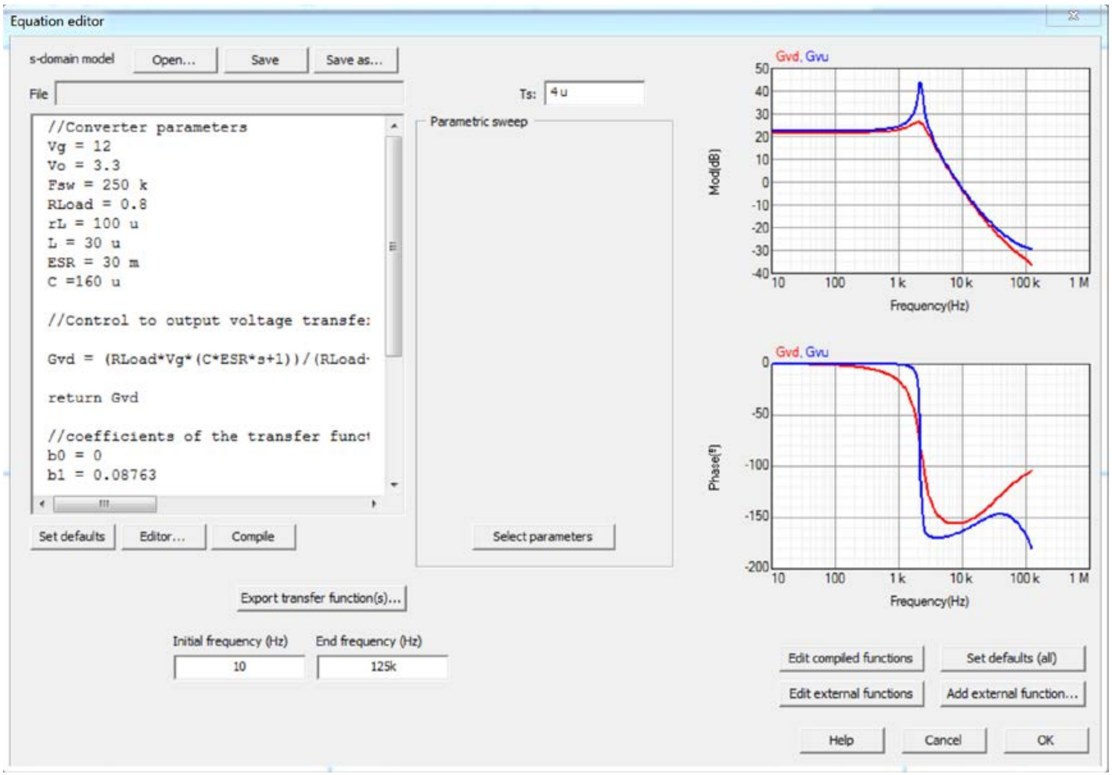

Figure 6: Comparison of transfer function using SmartCtrl's equation editor

3. Design flow of the digital control using the equation editor

First, we select the option "Equation editor" that is in the section “design a generic control system”.

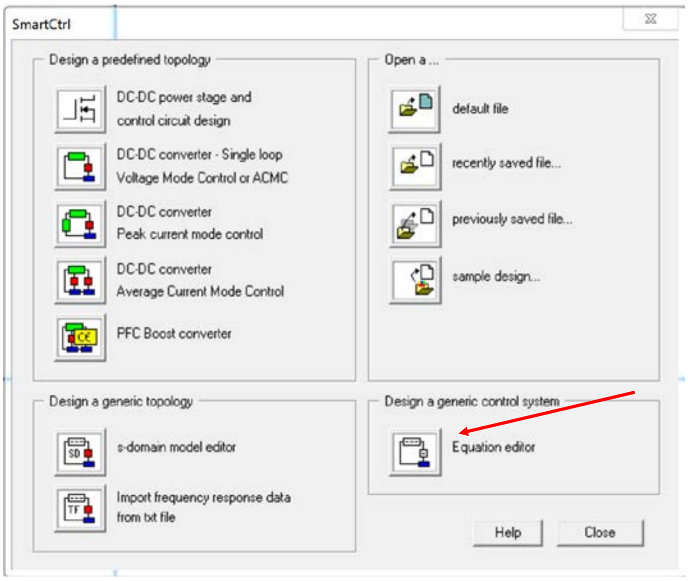

Figure 7: Clicking on design a generic control system

Then we insert the discrete transfer function that is in the previous section. In order to define the sampling period (T_s), click on “Compile” button. After we insert the switching frequency. In the "end frequency" box we will enter the Nyquist rate that corresponds to half of the switching frequency, as it is shown in Figure 8.

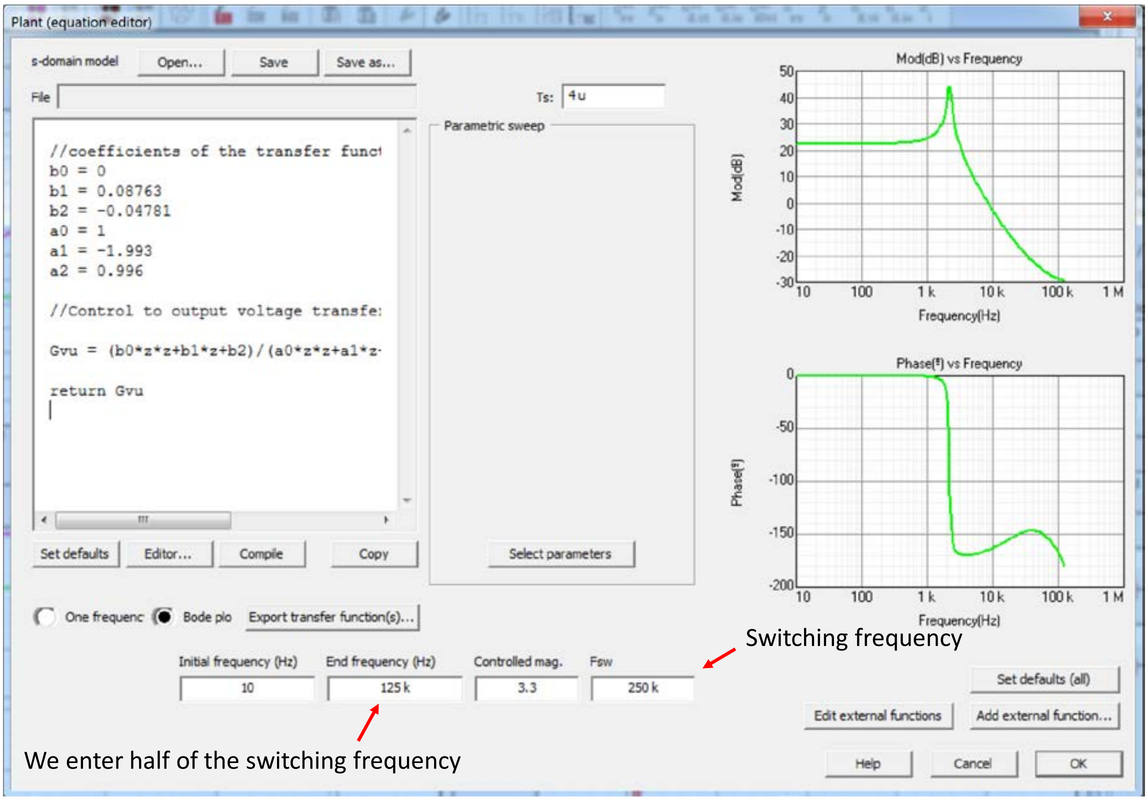

Figure 8: Plant parameters

Once "ok" is clicked, the sensor transfer function must be entered. In Figure 9 you can see another way that you can insert the coefficients when it is using the z-transfom. For this example, we are considering an isolated voltage sensor. (Low pass filter)

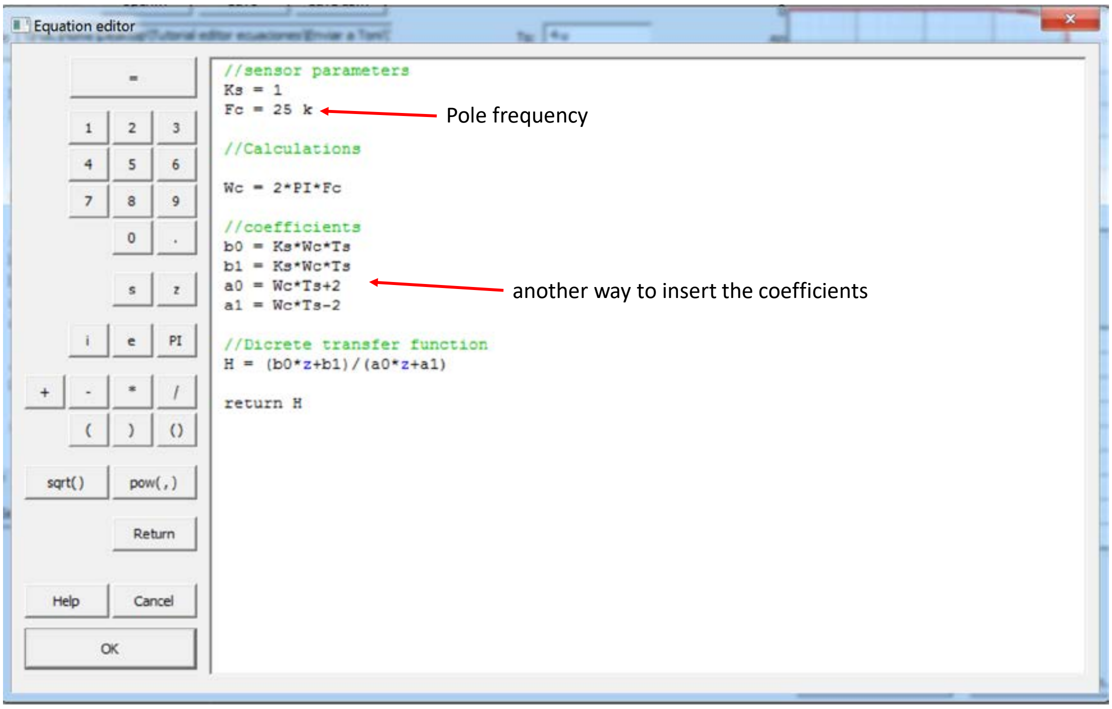

Figure 9: Sensor transfer function

The sampling period that appears in Figure 10, it is the same as we define in the plant section. If we want to change the value of T_S we can do it in this part of the design or later, since it will be updated for all the sections.

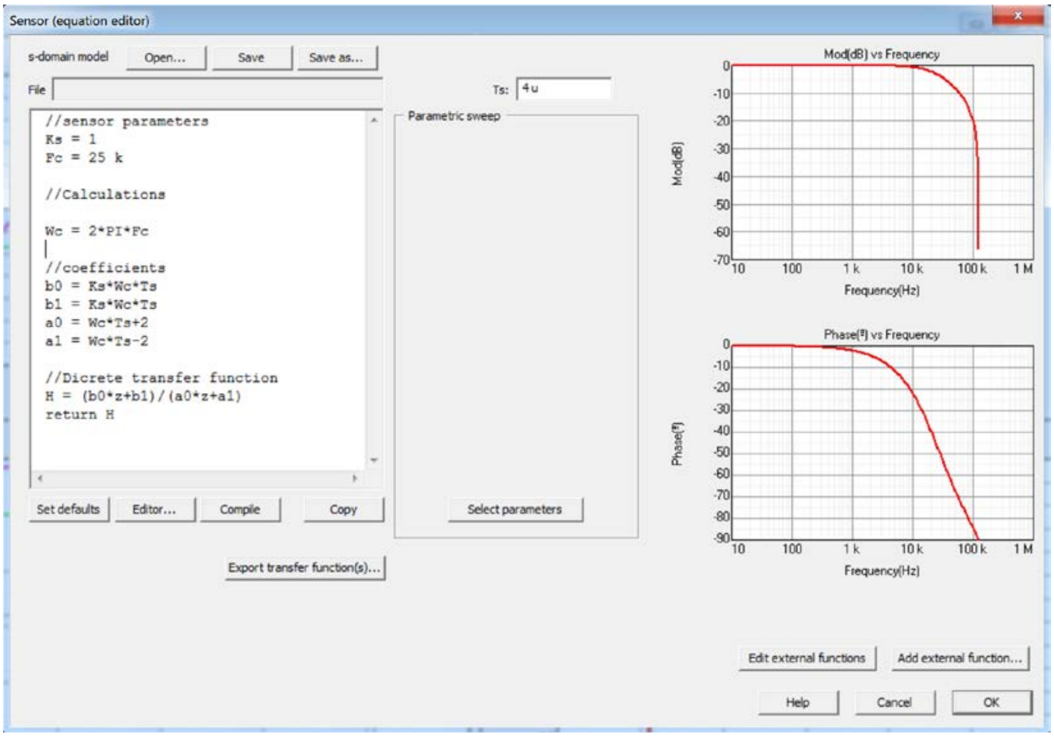

Figure 10: Sensor (equation editor)

Then you choose the digital option.

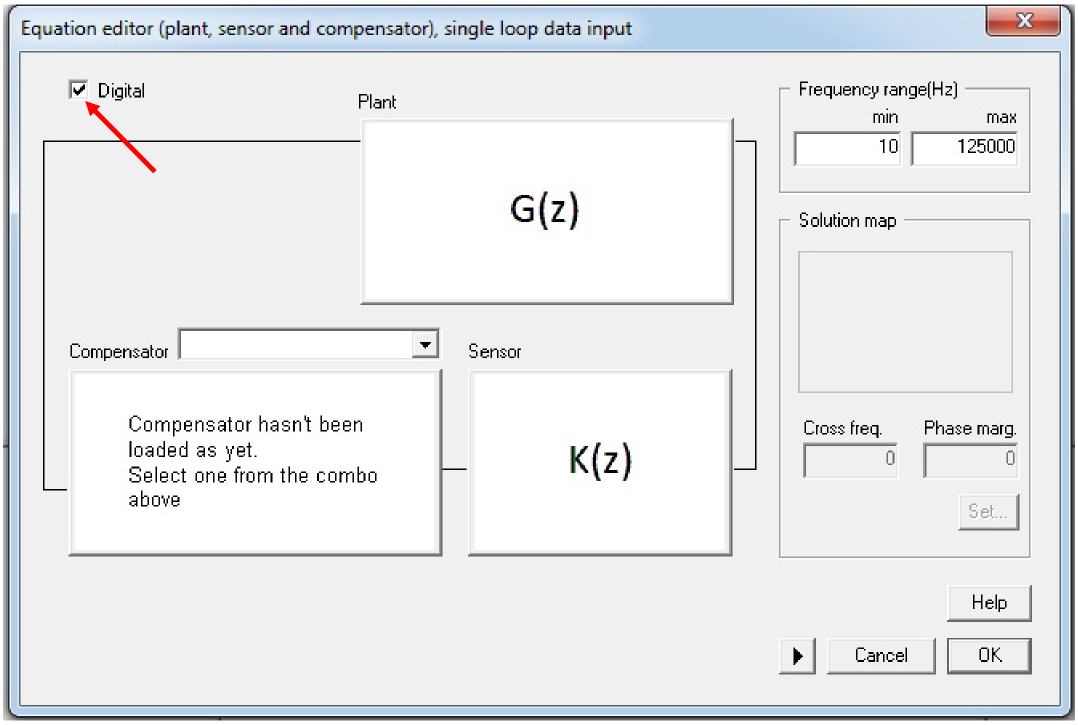

Figure 11: Digital option

You can select some of the predefined compensators such as digital PI and digital PID or you can choose the option of the “equation editor” which allows entering the transfer function of the compensator. For this example, we select the option "Equation editor".

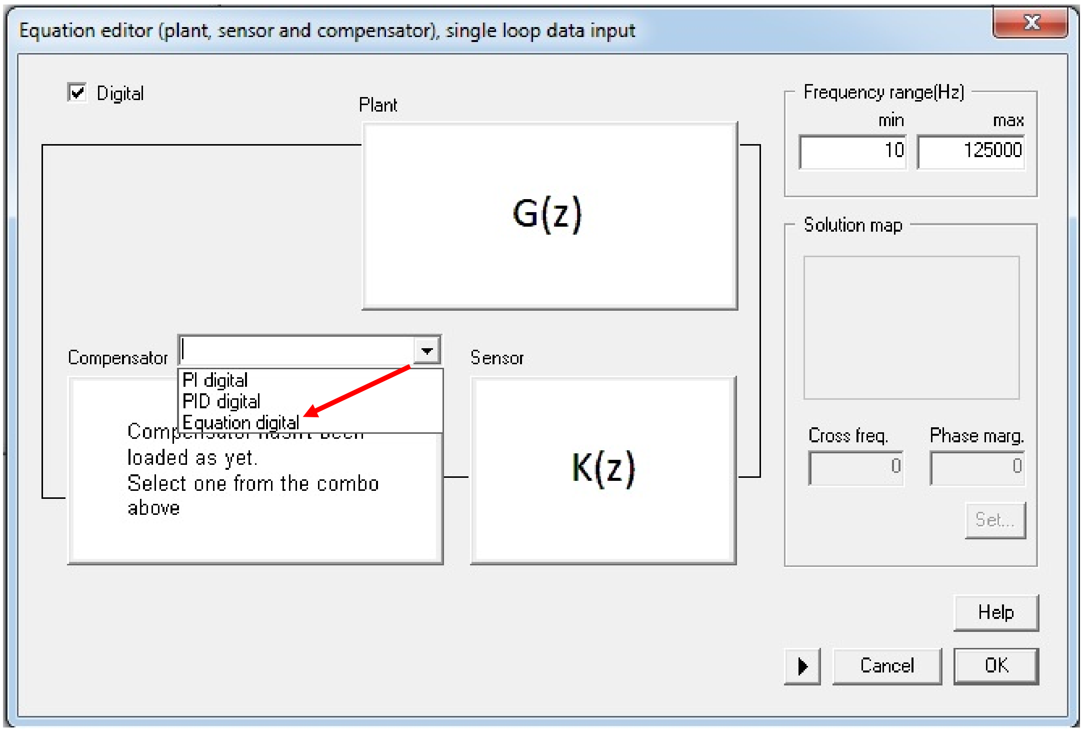

Figure 12: Choosing Equation editor option

In this example we will use a discretized PID compensator by Tustin integral operator approach. The transfer function is given by:

$$R(z) = \frac{b_0 \cdot z^2 + b_1 \cdot z + b_2}{a_0 \cdot z^2 + a_1 \cdot z + a_2}$$

$$b_0 = K_p \cdot \left[1 + \frac{T_s}{2 \cdot T_i} + \frac{T_d}{T_s}\right]$$

$$b_1 = K_p \cdot \left[-1 + \frac{T_s}{2 \cdot T_i} - \frac{2 \cdot T_d}{T_s}\right]$$

$$b_2 = K_p \cdot \left[\frac{T_d}{T_s}\right]$$

$$a_0 = 1$$

$$a_1 = -1$$

$$a_2 = 0$$

Where:

K_p: is the proportional gain.

T_i: is the integral gain.

T_d: is the derivative gain.

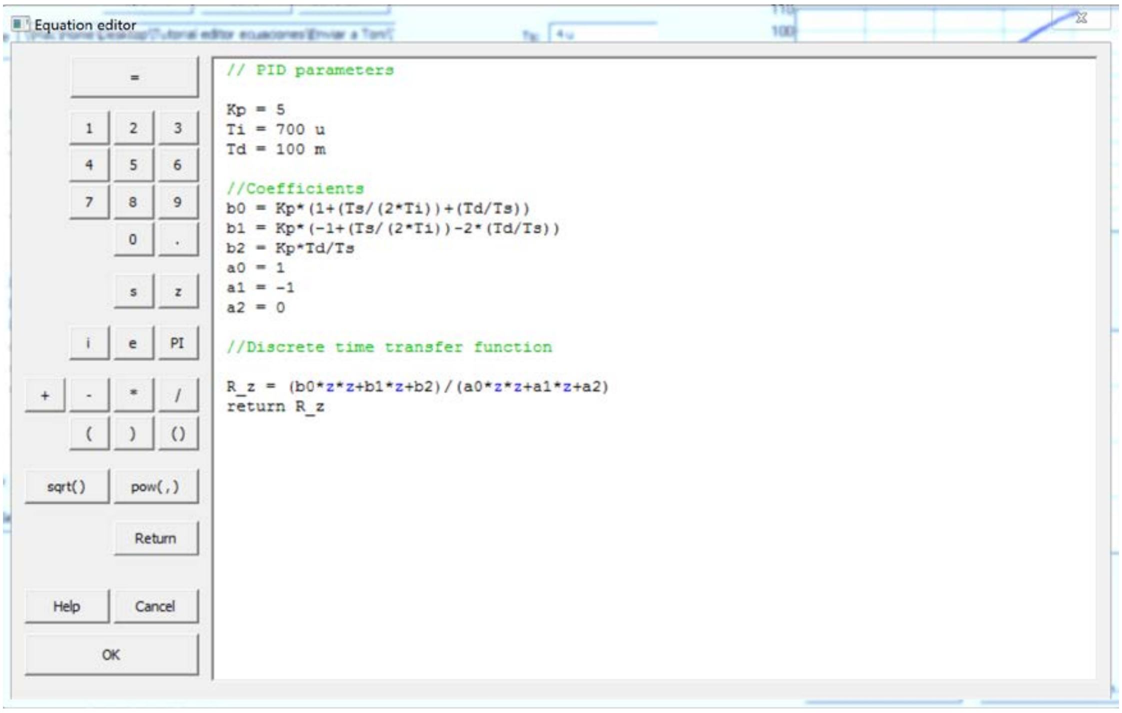

Figure 13: Equation editor of compensator section

If the "Select parameters" button is clicked, the program detects the numerical parameters and allows you to vary them with the sliders that appear in Figure 14. In this way we can analyze the frequency response as the parameters are varied. The compensator will be tuned later.

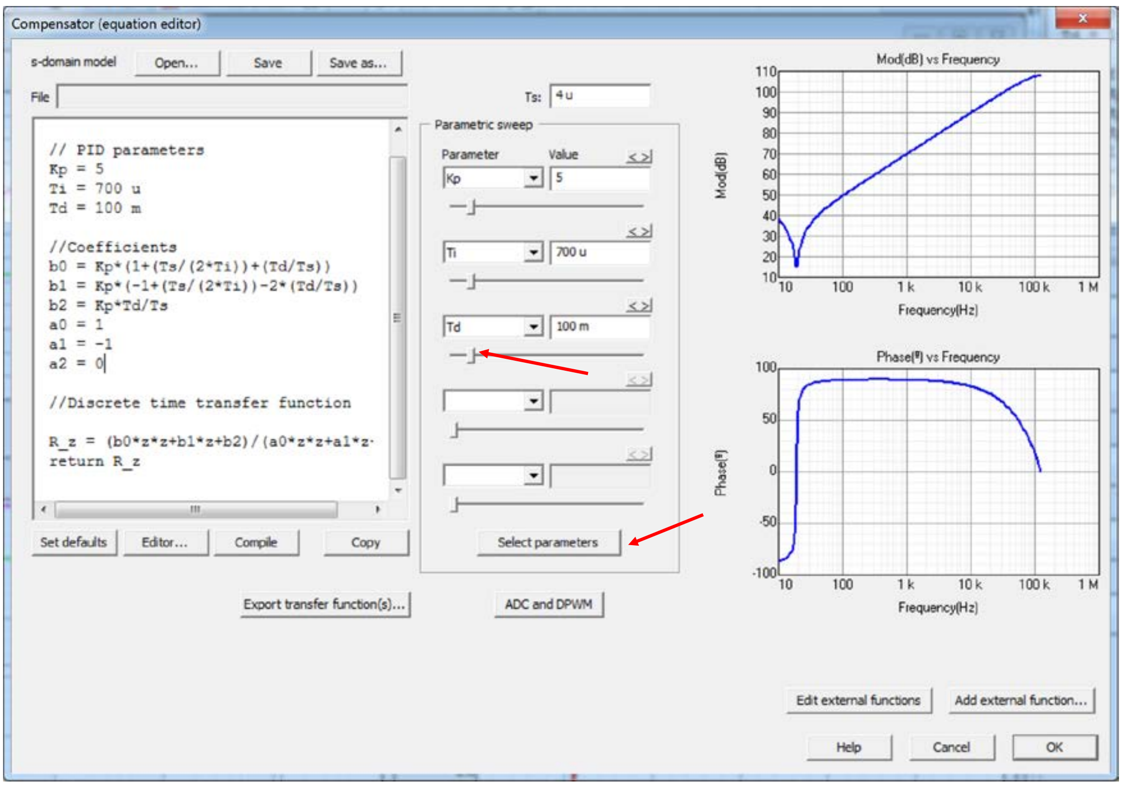

Figure 14: Choosing "Select parameters" option

As the digital delay was not considered in the transfer function that was determined in section 1, it is important to enter it in the ADC and DPWM option.

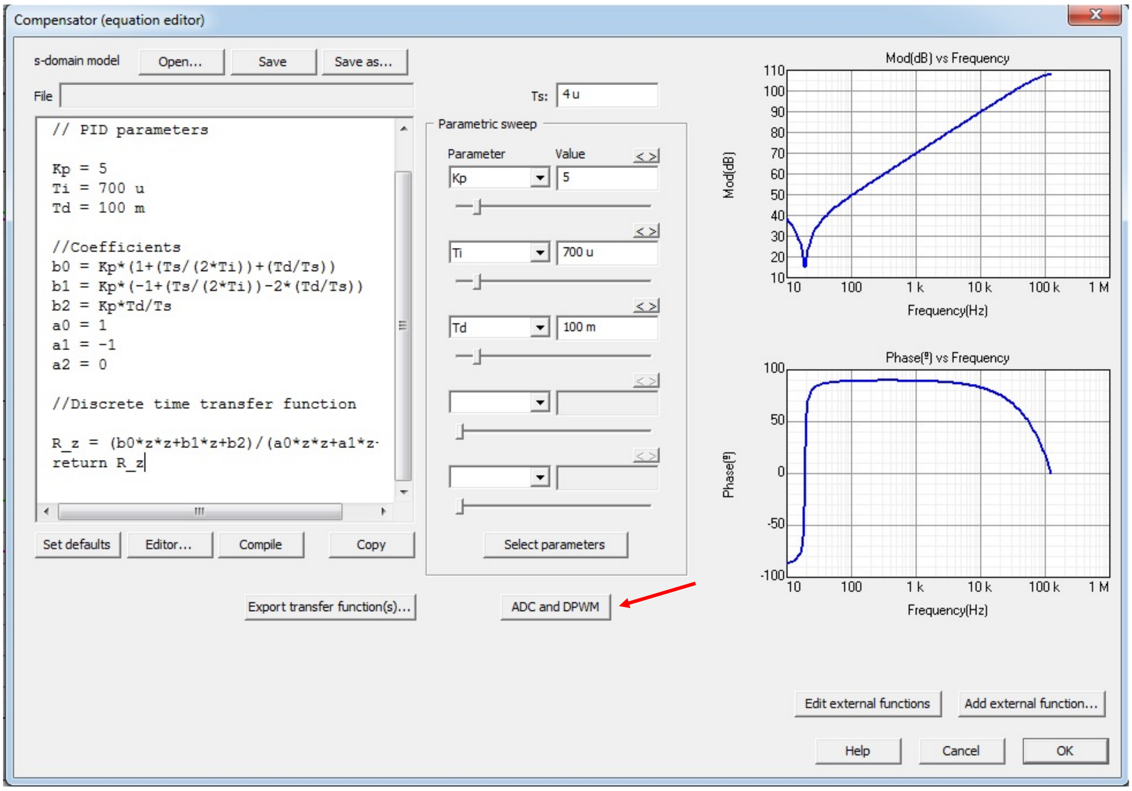

Figure 15: Choosing “ADC and DPWM” option

ADC panel:

V_{min}: minimum voltage the ADC is able to read, used to calculate its gain.

V_{max}: maximum voltage the ADC is able to read, used to calculate its gain.

N_{bits}: number of bits of the ADC to represent the analog input value.

F_{samp}: sampling frequency of the digital regulator.

t_{sync}: it accounts for the time difference between the moment when a signal is sampled and when it is used to update the regulator output.

Different types of carriers can be selected. It is necessary to insert the duty cycle. For this example, we consider a total delay of approximately 1 us, the carrier of trailing edge and duty cycle equal to 0.275.

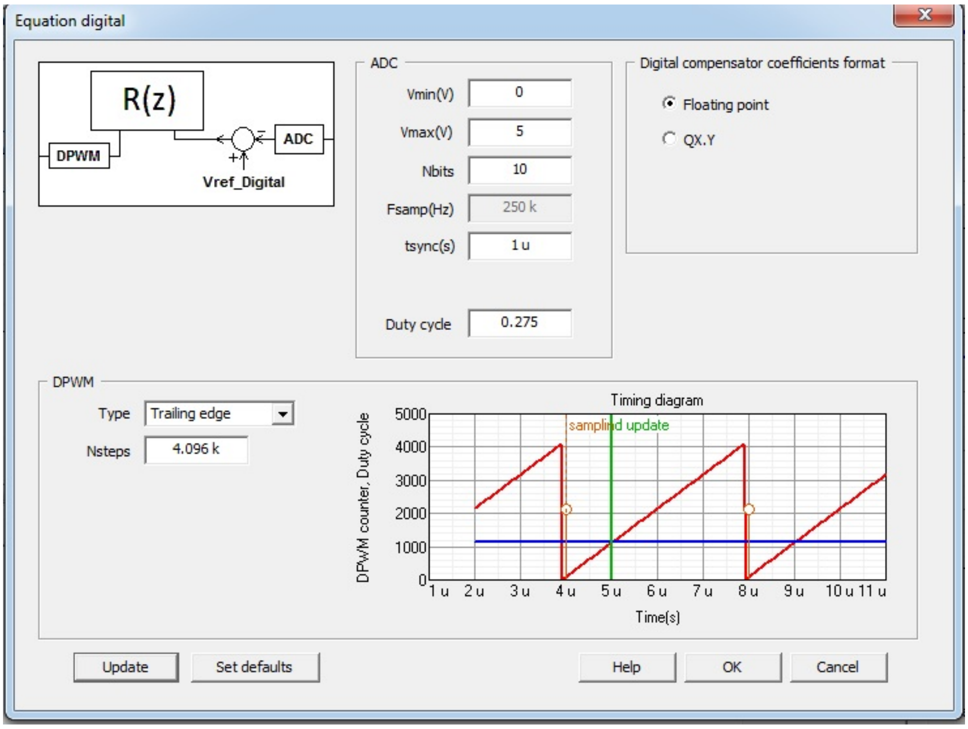

Figure 16: "ADC and DPWM" parameters

Note: If the total delay is already taken into account in the discrete model of the plant, a value of 0 must be considered in t_{sync}.

Finally, we click on "ok" button and proceed to tune the controller. With the sliders on the right, you can tune the compensator as shown in Figure 17.

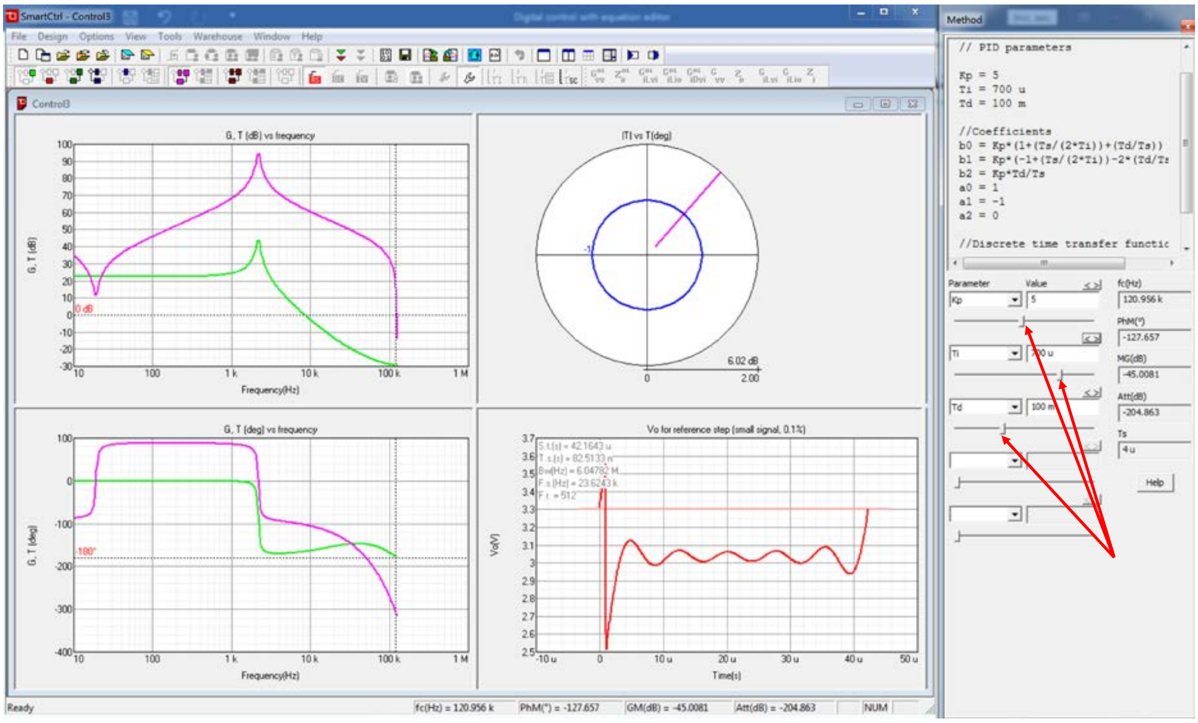

Figure 17: Tuning of compensator

For this example, a crossing frequency of 3.6 kHz and a phase margin of approximately 32.5 ° have been selected. As a result, we have the following constants:

$$K_p = 2.225$$

$$T_i = 163.6\ \mu$$

$$T_d = 39.92\ \mu$$

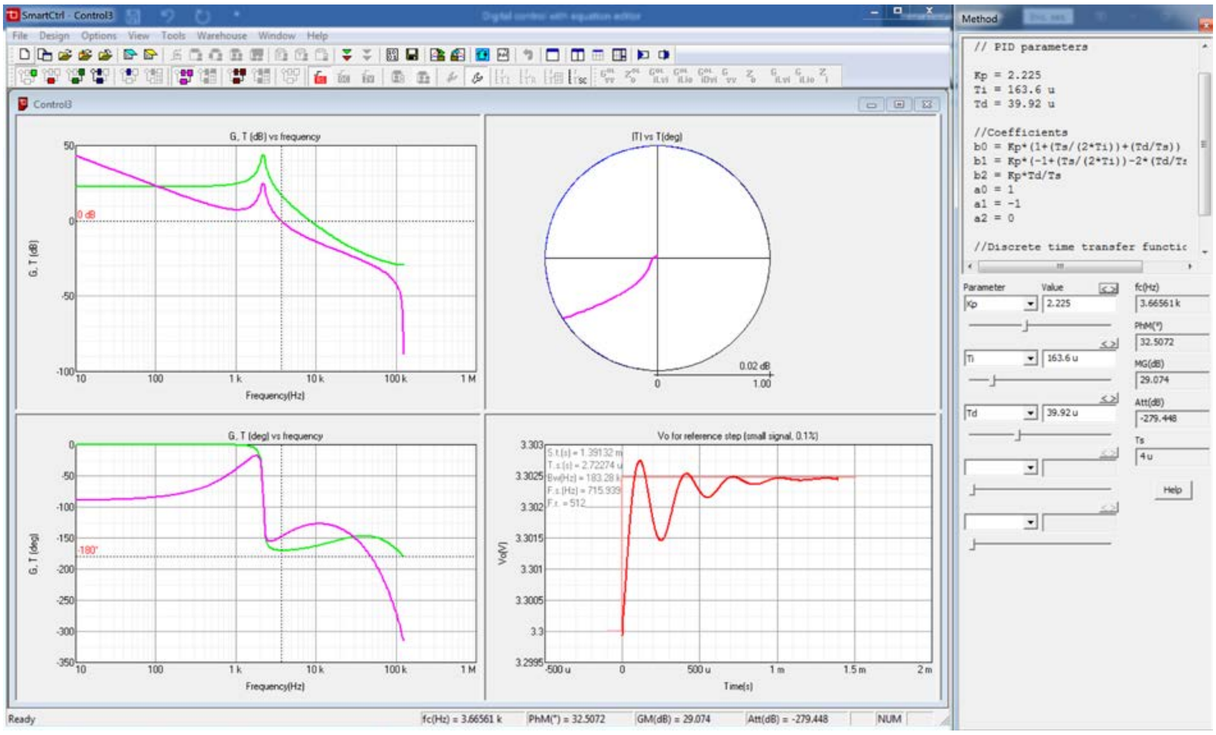

Figure 18: Compensator parameters

4. Psim simulation

Once the regulator has been designed, the simulation of the control in psim is carried out. In Figure 19 the schematic can be observed.

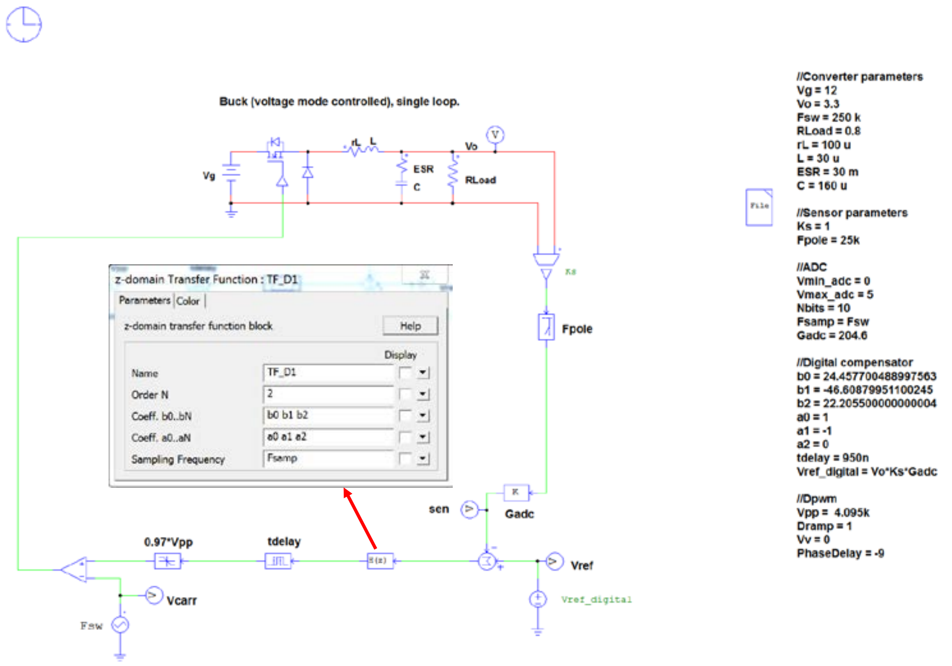

Figure 19: Psim schematic

From the parameters of the PID compensator, the coefficients of the transfer function are calculated.

$$b_0 = K_p \cdot \left[1 + \frac{T_s}{2 \cdot T_i} + \frac{T_d}{T_s}\right] = 24.457700488997563$$

$$b_1 = K_p \cdot \left[-1 + \frac{T_s}{2 \cdot T_i} - \frac{2 \cdot T_d}{T_s}\right] = -46.60879951100245$$

$$b_2 = K_p \cdot \left[\frac{T_d}{T_s}\right] = 22.205500000000004$$

The ADC gain is defined by:

$$G_{adc} = \frac{2^{N_{bits}} - 1}{V_{max} - V_{min}}$$

Finally, in Figure 20 you can see the transient response of the control against a load perturbance.

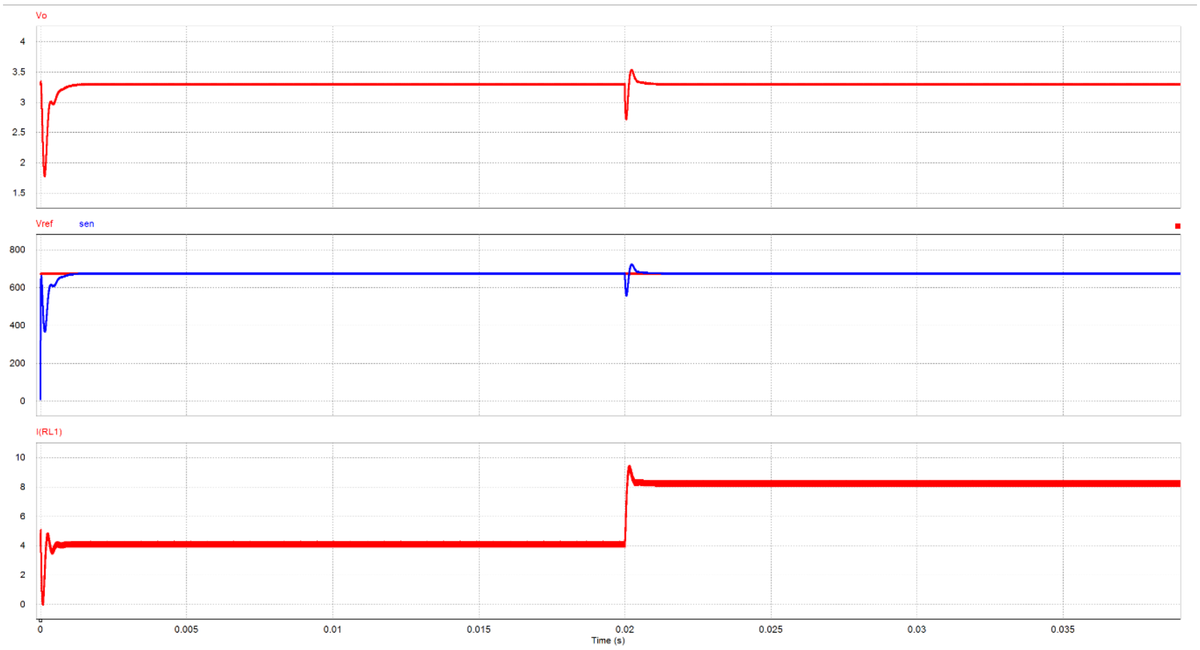

Figure 20: Transient response