



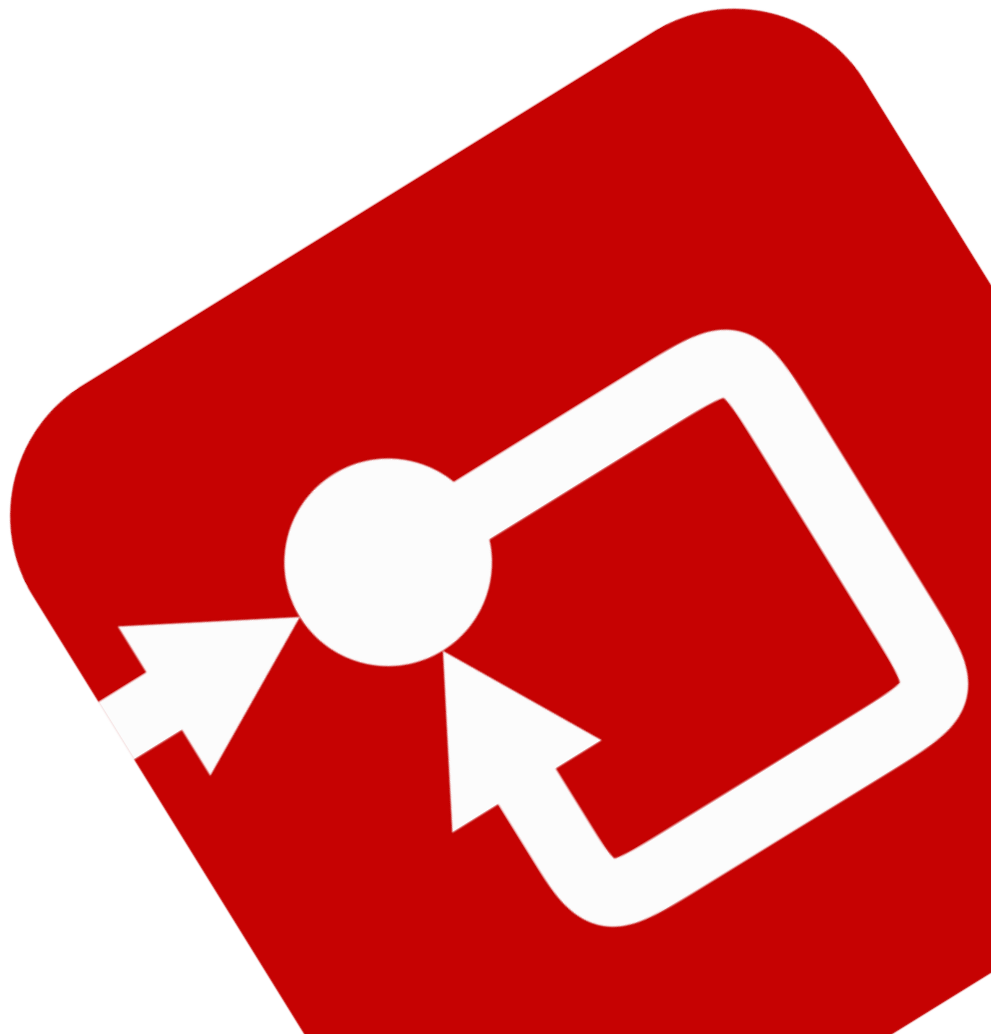
**smart**  
**ctrl**

control design for power electronics

# Ways of entering the small-signal model of a converter

Case of analysis: buck converter

Tutorial –December 2018-



## How to Contact:



[info@powersmartcontrol.com](mailto:info@powersmartcontrol.com)



[www.powersmartcontrol.com](http://www.powersmartcontrol.com)

SmartCtrl Copyright © 2015-2018 Power Smart Control S.L.

All Rights Reserved.

No part of this tutorial may be reproduced or modified in any form or by any means without the written permission of Power Smart Control S.L.

### 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.



## General Index

1. Introduction.....	3
2. Enter the converter transfer function .....	4
3. Enter the s domain transfer function.....	8

## Figure Index

Figure 1: Psim schematic with the AC sweep included .....	4
Figure 2: Result of Psim AC sweep.....	4
Figure 3: Psim AC sweep data points.....	5
Figure 4: shortcut in SmartCtrl for importing data files .....	5
Figure 5: Importation form in SmartCtrl.....	5
Figure 6: SmartCtrl loop wizard.....	6
Figure 7: Unattenuated PI compensator parameters.....	6
Figure 8: solution map.....	7
Figure 9: SmartCtrl analysis and solution window.....	7
Figure 10: PI compensator constant values.....	7
Figure 11: Access to the equation editor in SmartCtrl.....	8
Figure 12: Equivalent small signal circuit .....	9
Figure 13: SmartCtrl s model of the plant.....	9
Figure 14: SmartCtrl plant transfer function .....	10
Figure 15: Sensor definition .....	10
Figure 16: compensator definition .....	11
Figure 17: solution map fully defined .....	11
Figure 18: SmartCtrl main window.....	11
Figure 19: PI (sensor attenuated) coefficients .....	12

## 1. Introduction

There are many predefined topologies, compensators and control types in SmartCtrl that allow a straightforward design of the control loop.

However, according to Murphy's Law, the converter one wants/requires is not in these long list; if this is the case, SmartCtrl allows user to introduce the small signal model of the converter directly.

To do so, two workflows are available:

1. Importing the frequency response or AC sweep of the converter. This data can be obtained from a Psim AC sweep or directly from an impedance analyzer measure. This way can be considered the empirical one.
2. Entering the s domain transfer function. This way can be considered the theoretical one.

Thanks to these capabilities, the designer is able to design the control loop and optimize it with the help and tools provided by SmartCtrl any plant.

This tutorial is intended to show you the capabilities and potentialities of both workflows.

As this is a conceptual tutorial, a case of study has been done: a voltage mode controlled Boost converter.

## 2. Enter the converter transfer function

The first step is to have a converter whose transfer function can be measured, this applies to almost any electronic simulator or any device with the ability to measure open loop gain of converters (impedance analyser devices).

For simplicity, in this tutorial, it has been done an AC sweep in Psim being the case of study a buck converter. So, the first step is to create the schematic in Psim with the AC sweep. See Figure 1.

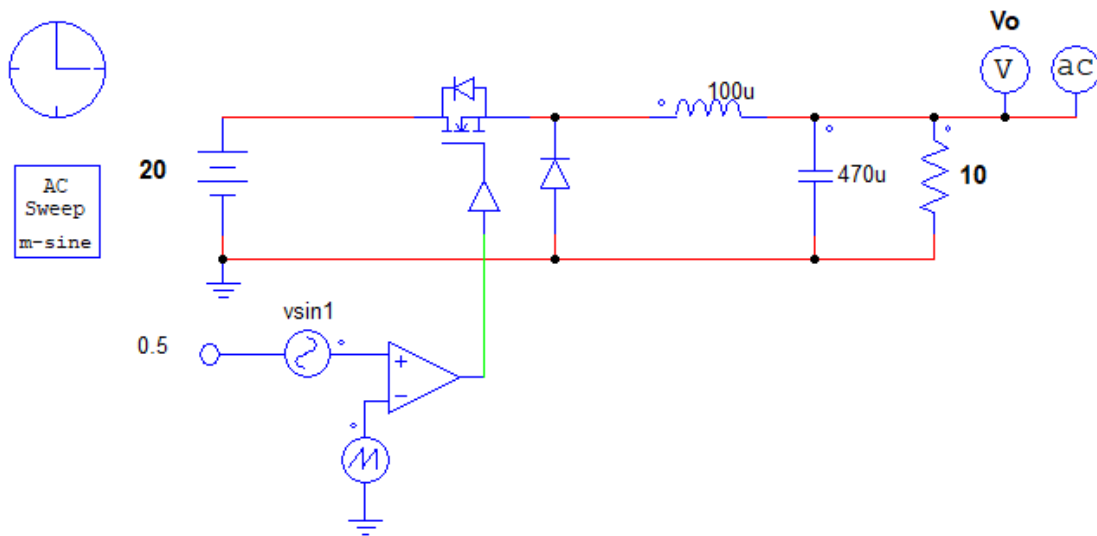


Figure 1: Psim schematic with the AC sweep included

The result of the AC sweep is shown in Figure 2.

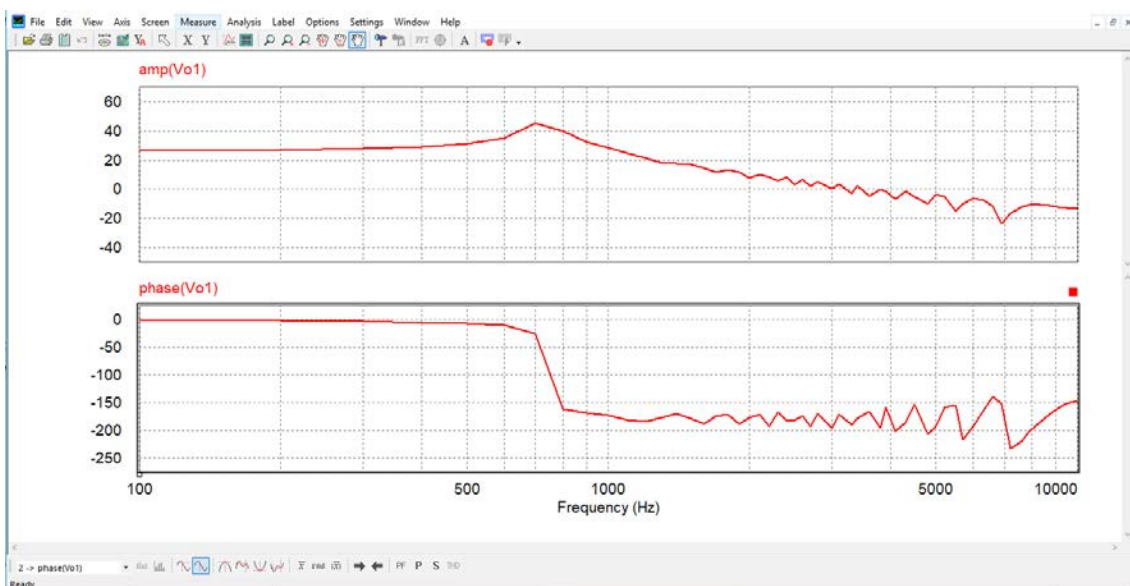
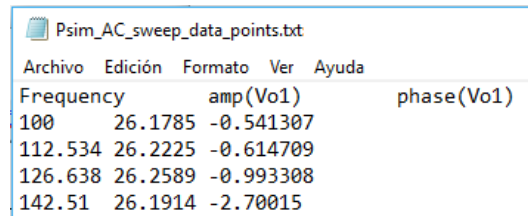


Figure 2: Result of Psim AC sweep

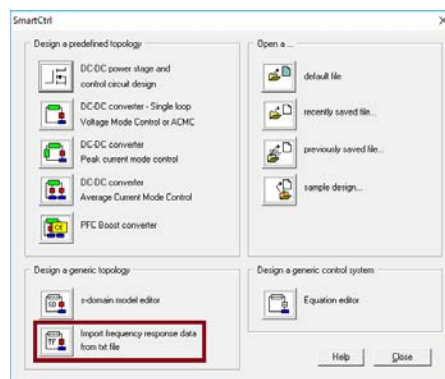
Psim generates an Excel compatible comma tab separated data file which has the shape of Figure 3. In case an impedance or other software has been used, a file similar to this has to be constructed. Most of the software and impedance analyser has the capability of generating this kind of files already included, so it is a quite straight forward process.



Frequency	amp(Vo1)	phase(Vo1)
100	26.1785	-0.541307
112.534	26.2225	-0.614709
126.638	26.2589	-0.993308
142.51	26.1914	-2.70015

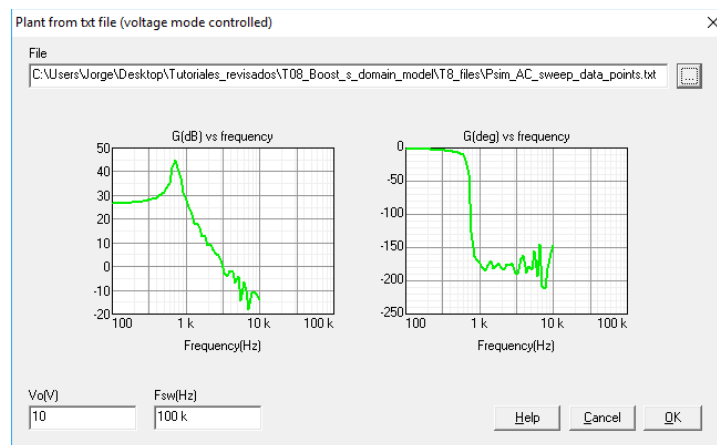
**Figure 3: Psim AC sweep data points**

To import this file (whatever its source) just open SmartCtrl and click in Import frequency response data from txt file and select if the measured data corresponds to a voltage or to a current. In this case, select voltage. See Figure 4.



**Figure 4: shortcut in SmartCtrl for importing data files**

Select the path of the data file and type the switching frequency of the system and the desired output voltage. In this case, the buck converter is commutated at 100kHz and the wanted output voltage is 10V. See Figure 5.



**Figure 5: Importation form in SmartCtrl**

When clicked OK, SmartCtrl will show the window of Figure 6 asking for the rest of the components of the loop. In this window define an embedded voltage sensor and a PI compensator unattenuated with the parameters shown in Figure 7.

Note that the modulator details (and gain) are included in the compensator definition. See Figure 7.

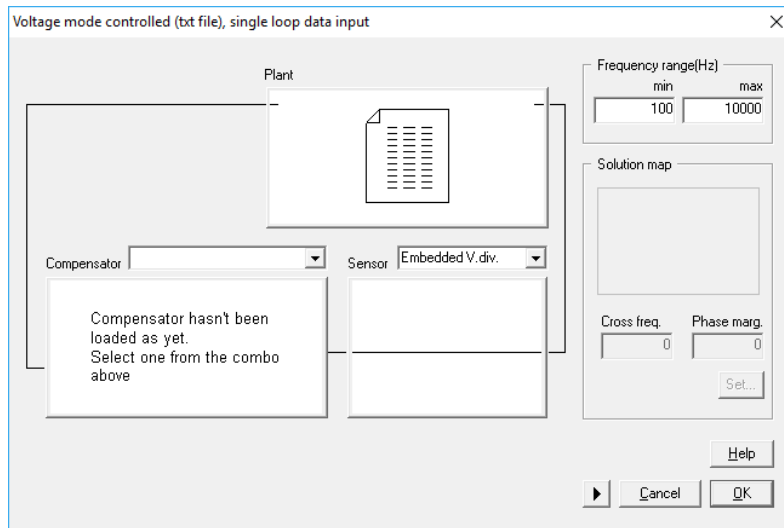


Figure 6: SmartCtrl loop wizard

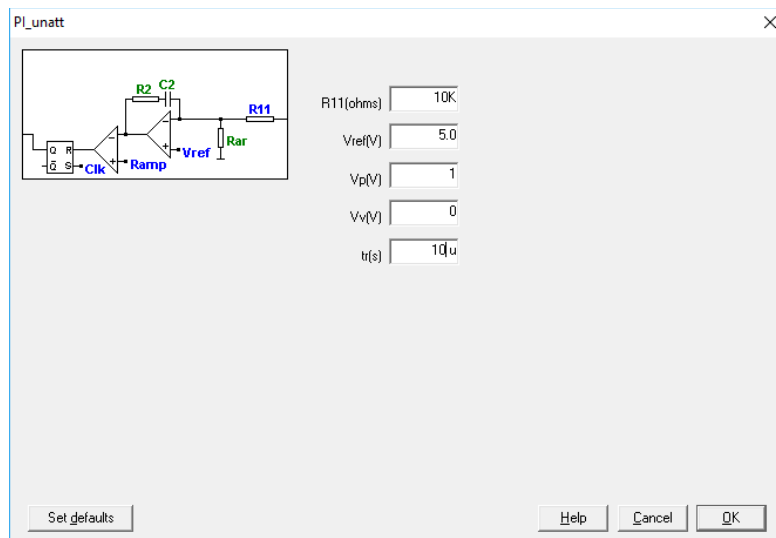


Figure 7: Unattenuated PI compensator parameters

Now click OK to close the compensator and click in set button in the right side of the window to access the solution map.

The solution map is a graphical tool which shows the designer a white area where the pairs cross frequency and phase margin generates stable solutions. In this case, a cross frequency of 600Hz and a phase margin of 90 degrees have been selected. See Figure 8.

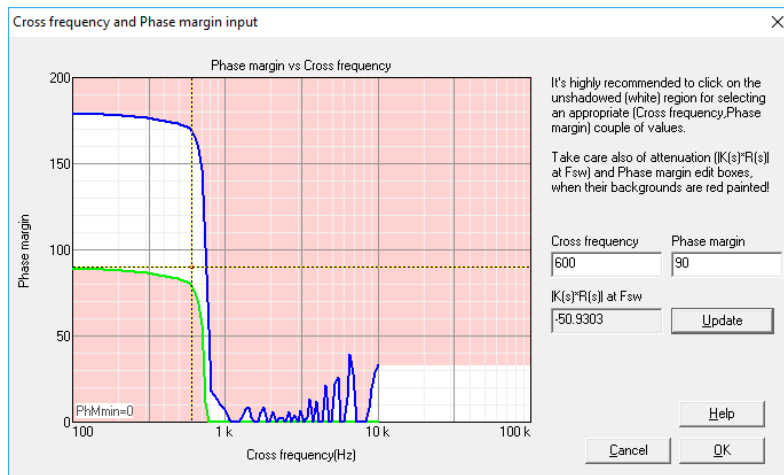


Figure 8: solution map

Click OK to close the solution map window and OK again to show the loop wizard. Now SmartCtrl will show the main window where the design can be optimized by checking the bode plots. See Figure 9.

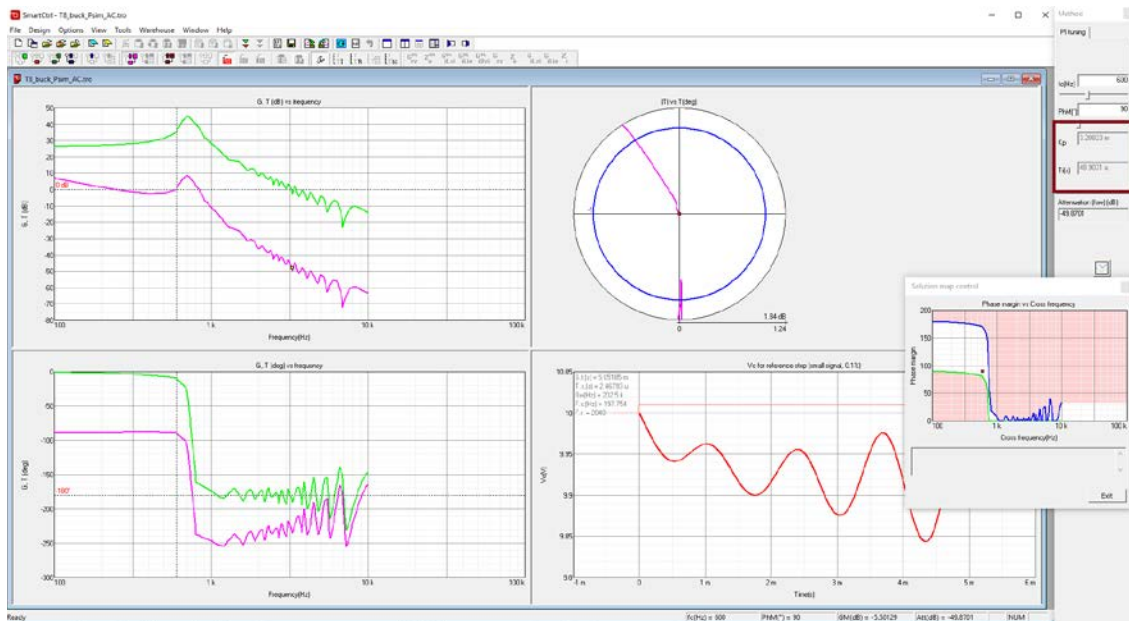


Figure 9: SmartCtrl analysis and solution window

In this case, the compensator data is the cone collected in Figure 10.

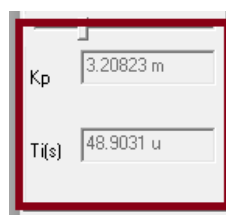


Figure 10: PI compensator constant values



At his point, the compensator will be fully defined and it can be implemented from

### 3. Enter the s domain transfer function

There are certain cases in which the s domain editor can aport a lot:

1. If the converter is not in the list of already defined converters
2. If the converter has not been simulated or constructed.
3. If the AC sweep measure cannot be performed.
4. If it the control loop is to be taken to the limits.

In all this situations, the s domain editor, included in the equation editor, will make the design possible.

S domain editor will make the difference; however, it needs some small signal knowledge by the user. To access it, open SmartCtrl and open the Equation editor. See Figure 11.

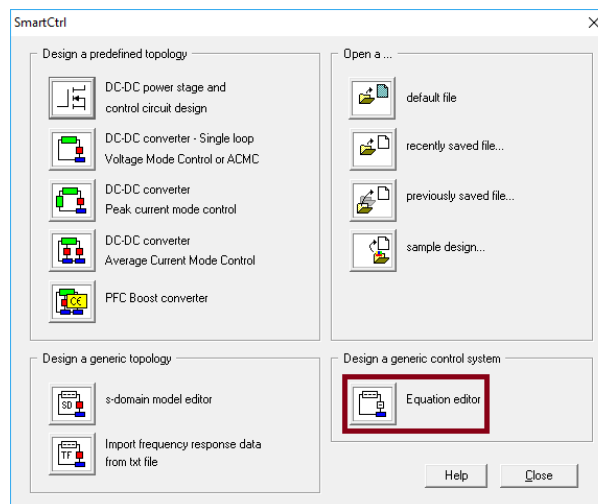


Figure 11: Access to the equation editor in SmartCtrl

As it is a buck converter what it is been defined, the following assumptions has been made:

- a) No parasitic elements have been defined
- b) The converter is supposed to be a LTI system (linear time-invariant).

These assumptions allow to obtain the equivalent converter shown in Figure 12 where the desired transfer function in output voltage ( $V_o$ ) divided by duty cycle ( $D$ ).

To relate  $V_o$  with  $D$  it is just necessary to realise that the equivalent converter is just an impedance divider. If this idea is fully understood, the code shown in Figure 13 will be perfectly clear.

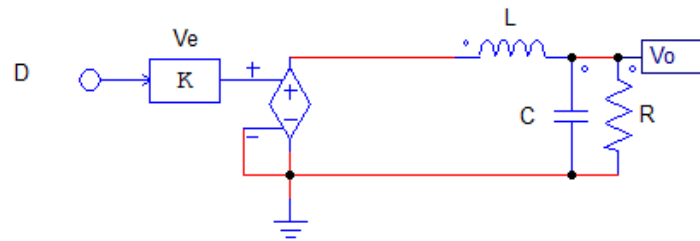


Figure 12: Equivalent small signal circuit

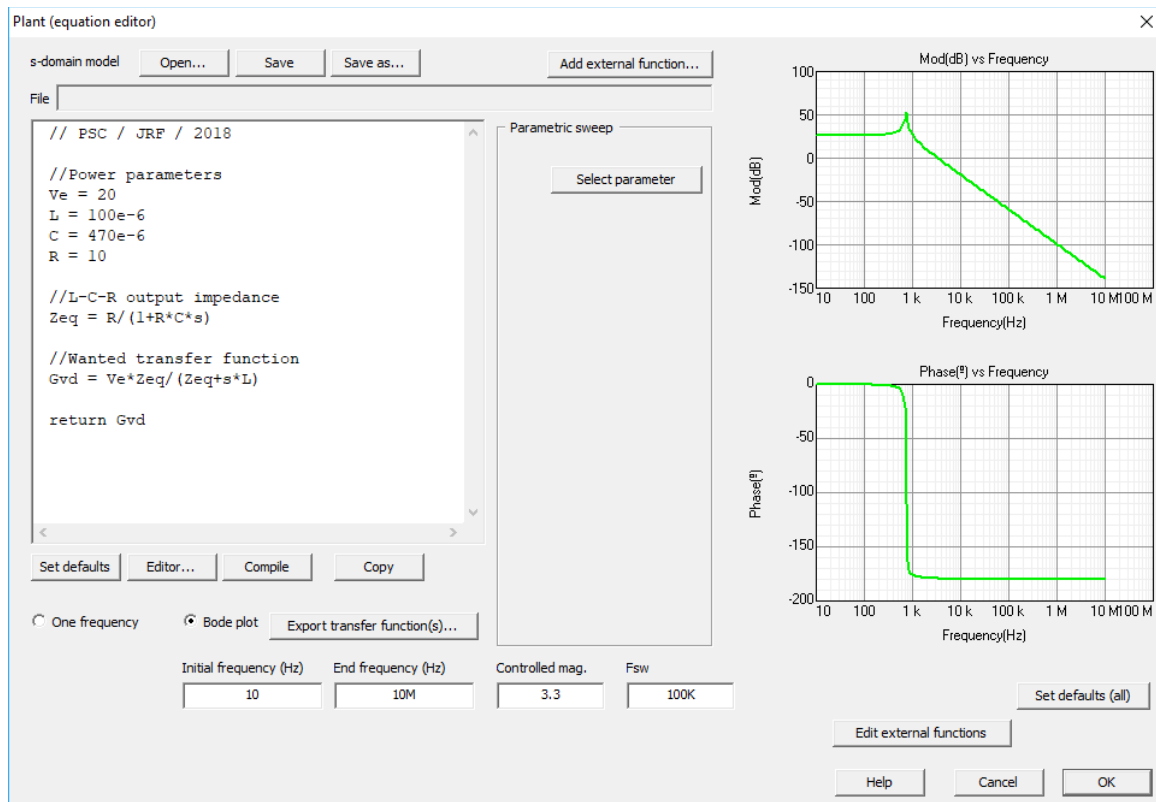
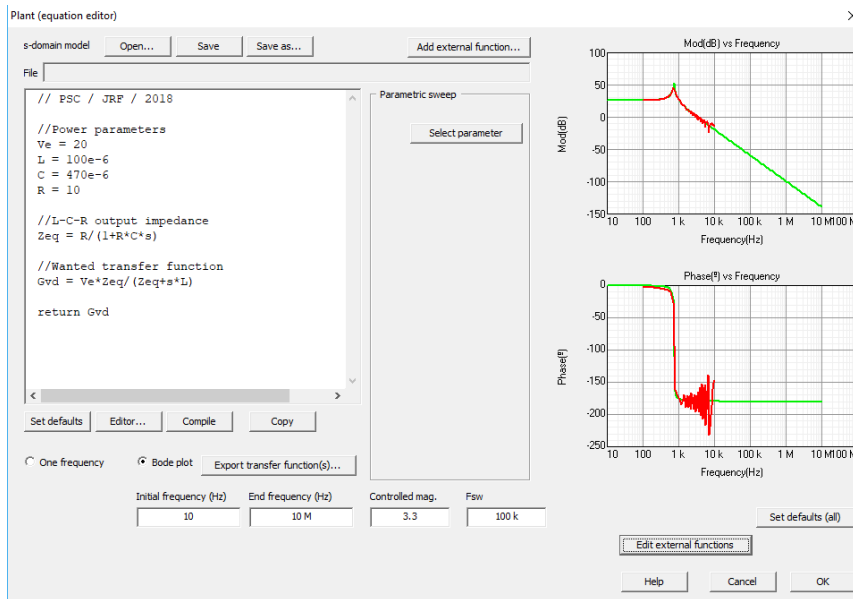


Figure 13: SmartCtrl's model of the plant

It can be seen that the AC response obtained in Figure 13 is not the same that the one obtained by the AC sweep in Psim. However, its general form is pretty similar.

In fact, it is possible to load the Psim AC sweep data by clicking in the button "Add external trace". See Figure 14.

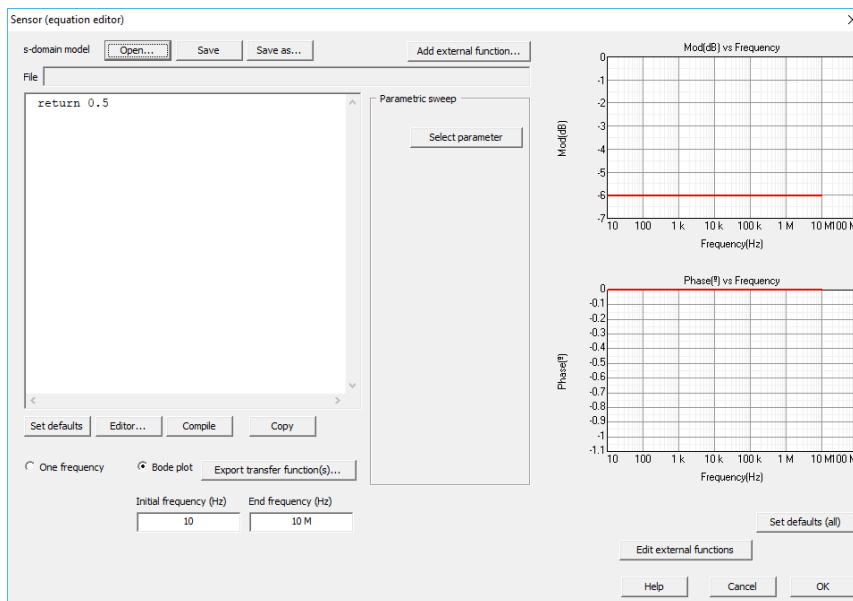
As it can be seen in Figure 14 the main difference between them is the height of the peak that is present in the bode plot and it is consequence of having made a simulation in Psim with not enough data points.



**Figure 14: SmartCtrl plant transfer function**

Once the plant is fully defined by Figure 14, it is time to define the sensor. To do that click in the sensor block and configure it as it has been done in Figure 15. In this way, the sensor will be the same that before.

Note that the sensor used before in the embedded PI has a gain of 0.5V/V because the output voltage of the converter is 10V and the Vref is 5V, this can only be achieved if R11 is equal to Rar so the voltage divider gain has to be 0.5V/V. See Figure 7. This gain is the one which should be inserted as a sensor in this workflow. See Figure 15.



**Figure 15: Sensor definition**

Define the compensator as it has been done in Figure 16 and click in set to choose a point in the solution map.

In the solution map choose the same point than before: 600Hz and 90 degrees and click OK twice to access the SmartCtrl main window. See Figure 17 and Figure 18.

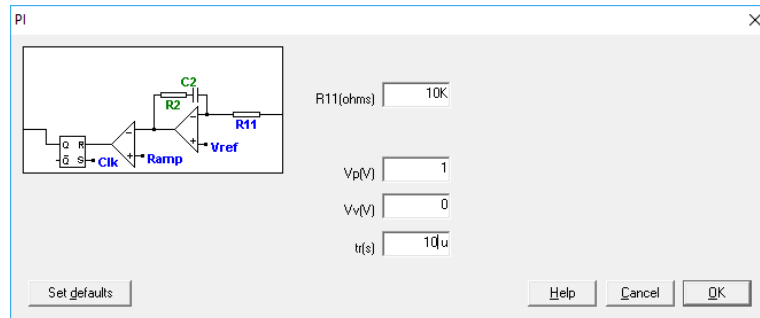


Figure 16: compensator definition

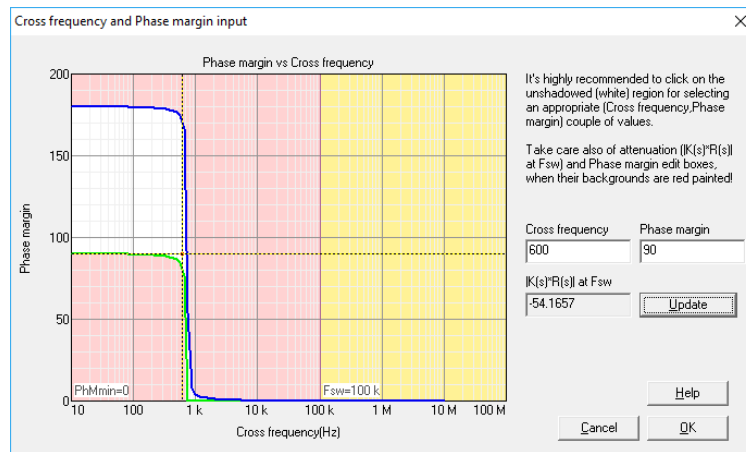


Figure 17: solution map fully defined

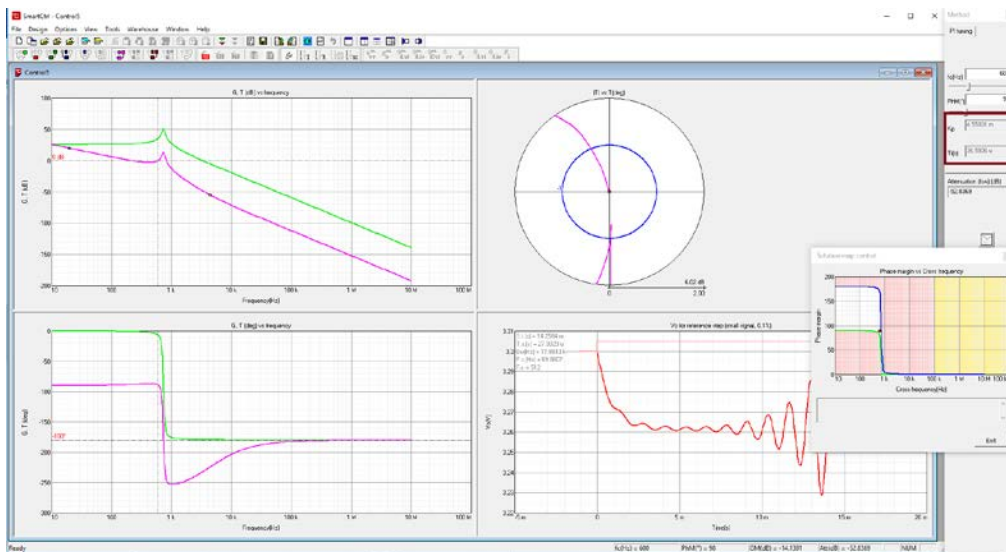
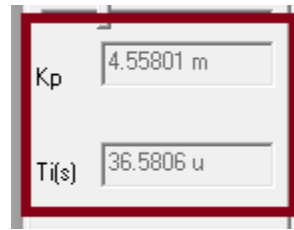


Figure 18: SmartCtrl main window

The output of the program is the values of Figure 19. If compared with the ones obtained before (Figure 7), it can be seen that they are really close.



**Figure 19: PI (sensor attenuated) coefficients**

In this way, it has been shown how SmartCtrl is not only restricted to predefined topologies; it is, in fact, a quite versatile software which can be used successfully in any situation.