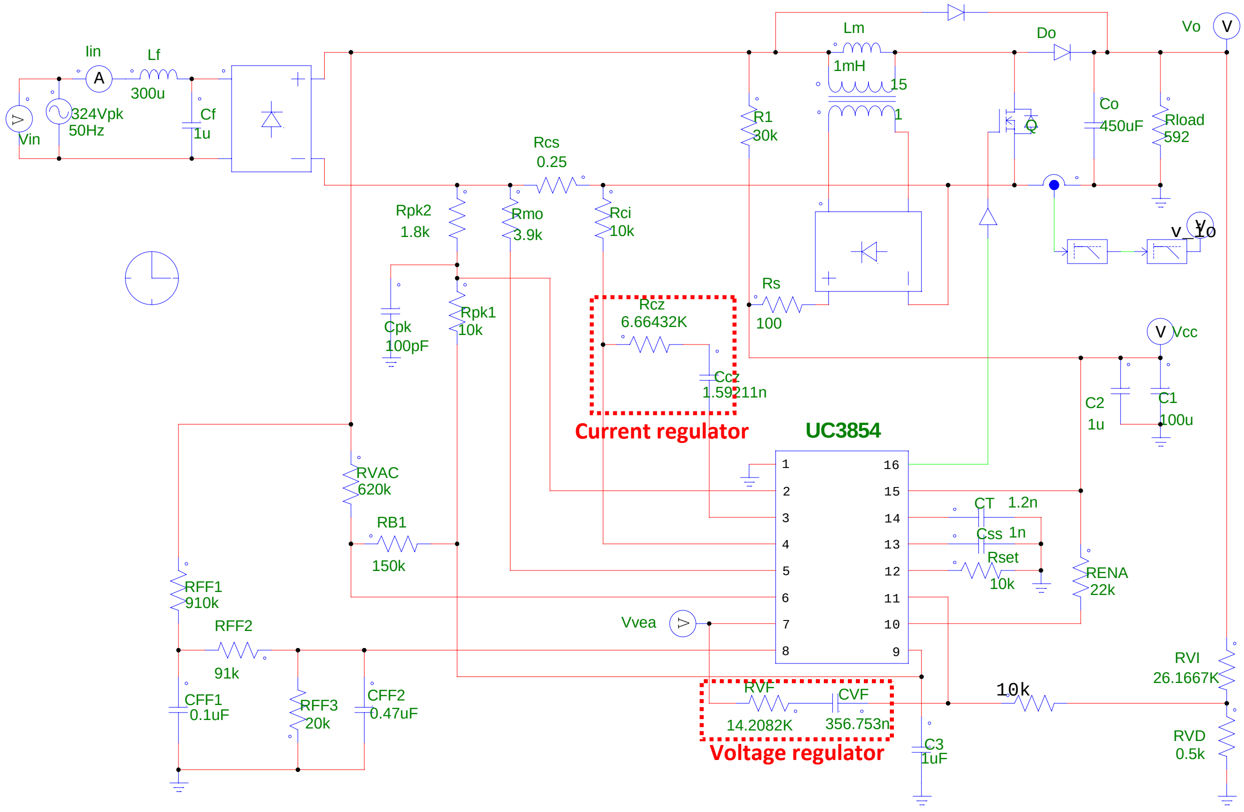
SmartCtrl Tutorial

Boost PFC Converter Control Loop Design

SmartCtrl[1] is a general-purpose controller design software specifically for power electronics applications. This tutorial is intended to guide you, step by step, to design the control loops of a PFC (power factor correction) boost converter with the SmartCtrl Software.

The example used in this tutorial is the PFC boost converter circuit that comes with the PSIM example set (the PSIM file is “UC3854_PFC.sch” under the folder “examples\PWM IC”). The PSIM schematic is shown below:



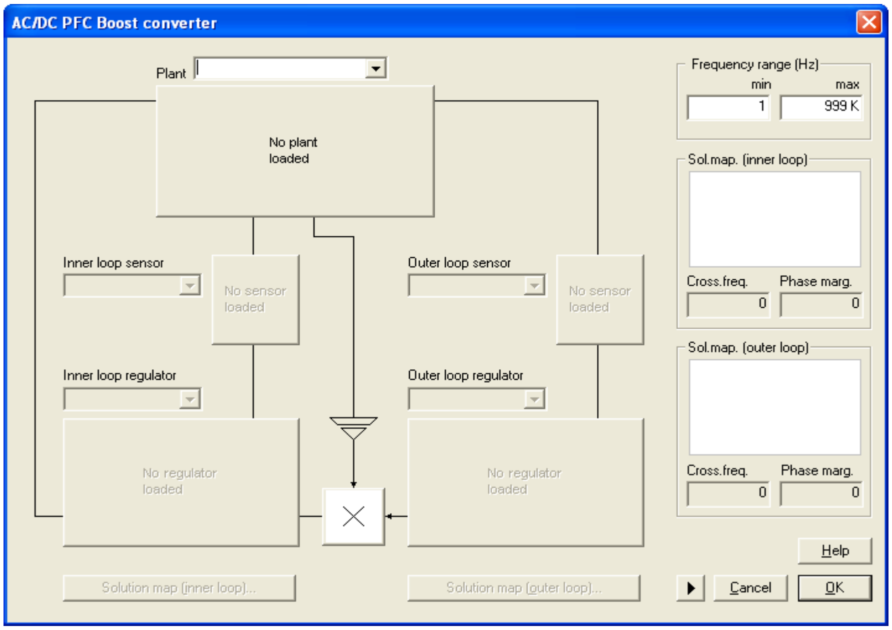
The circuit includes the inner current loop and the outer voltage loop. The current loop regulator parameters are the resistance Rcz and the capacitance Ccz, and the voltage regulator parameters are the resistance RVF and the capacitance CVF, highlighted in the red dotted boxes above.

Let’s assume that these values are unknown. The objective is to design the current/voltage regulators using the SmartCtrl software. The design procedure is described below.

To begin the design process, in SmartCtrl, click on the icon , or from the **Data** menu, select **Predefined topologies** -> **AC/DC converters** -> **PFC Boost converter**.

The dialog window will appear as follows.

[1] SmartCtrl is copyright in 2010 by Carlos III University of Madrid, GSEP Power Electronics Systems Group, Spain

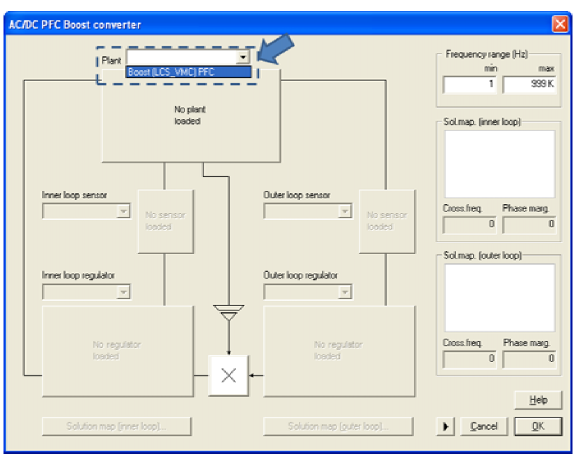


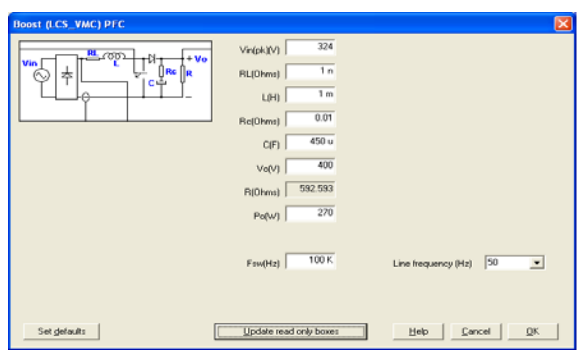
The PFC boost converter is controlled by a double loop control scheme. The inner loop is a current loop, and the outer loop is a voltage loop. Note that the PFC boost converter design must be carried out sequentially. The SmartCtrl program will guide you through this process.

Inner Loop Design

1. Define the converter

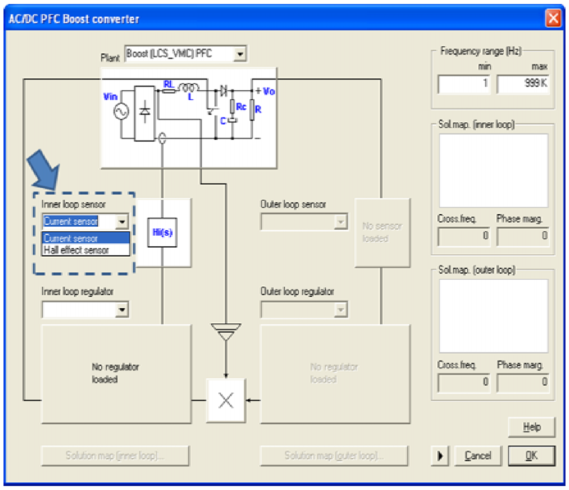
   Select the plant as **Boost (LCS_VMC) PFC** for boost PFC converter with the current loop and the voltage loop. Complete the parameters in the corresponding input data window. Note that the input voltage is the peak value. When finished, click OK to continue.

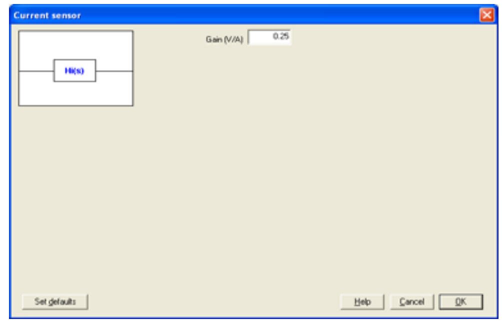




2. Select the current sensor

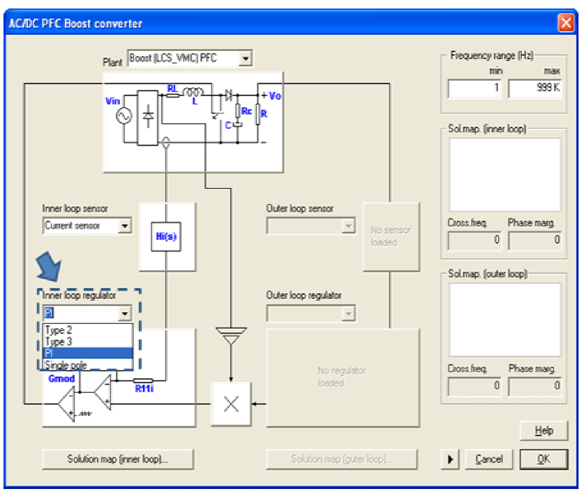
   Select the current sensor among the available types. In this example, select **Current sensor**, and set the gain to 0.25 based on the circuit (this value is equivalent to the sense resistor value).

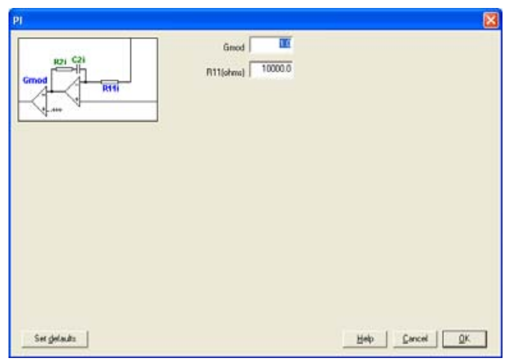




3. Select the current regulator

   From the inner loop regulator drop-down menu, select PI as the current regulator type.



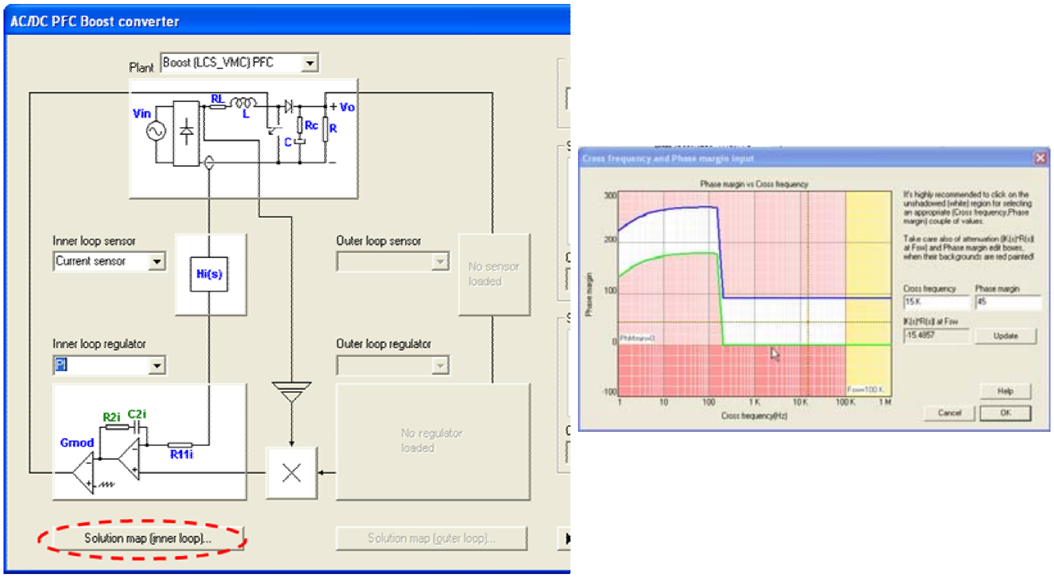


4. Select the crossover frequency and the phase margin

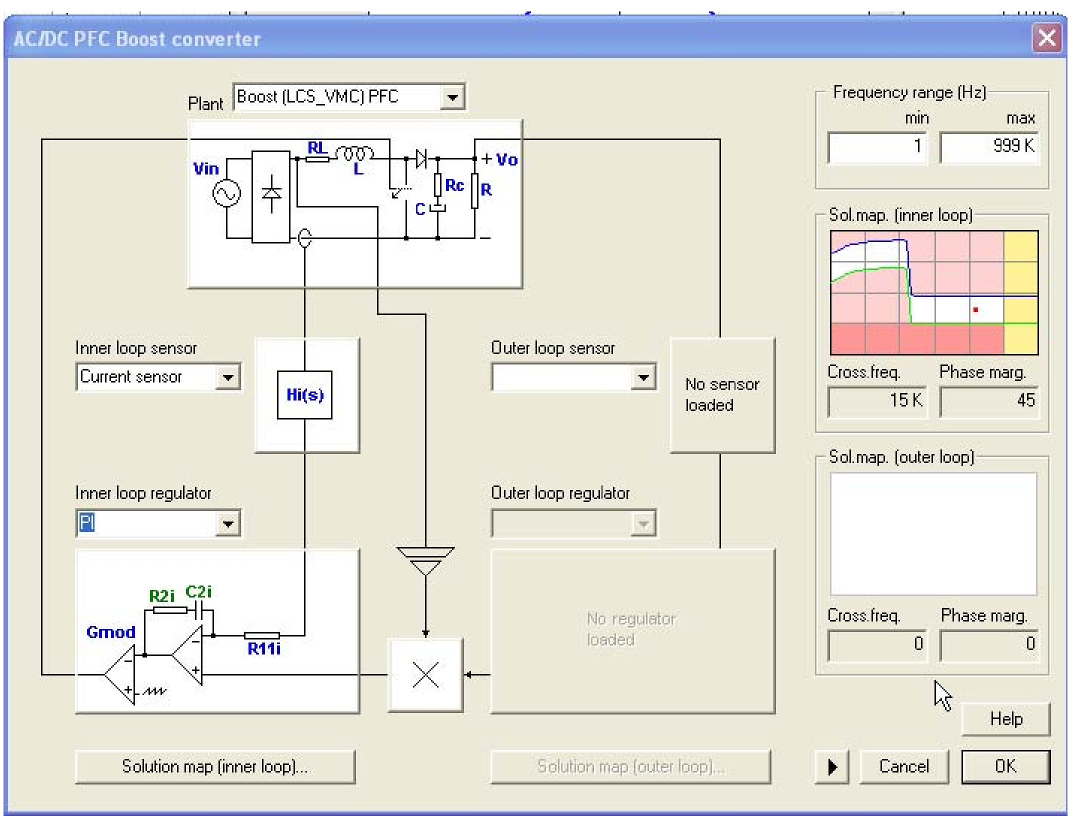
   SmartCtrl provides a guideline and an easy way of selecting the crossover frequency and the phase margin through the ***Solution Map***. Click on the **Set** button, and the Solution Map will be shown as below.

   In the Solution Map, each point within the white area corresponds to a combination of the crossover frequency and the phase margin that leads to a stable solution. In addition, when a point is selected, the attenuation given by the sensor and the regulator at the switching frequency is provided.

   To carry out the selection, left click a point within the white area, or enter the crossover frequency and the phase margin manually.



Once the crossover frequency and the phase margin are selected, the Solution Map will be shown on the right side of the converter input window. If, at any time, one needs to change the crossover frequency or the phase margin, click on the white area of the Solution Map, as shown in the figure below.

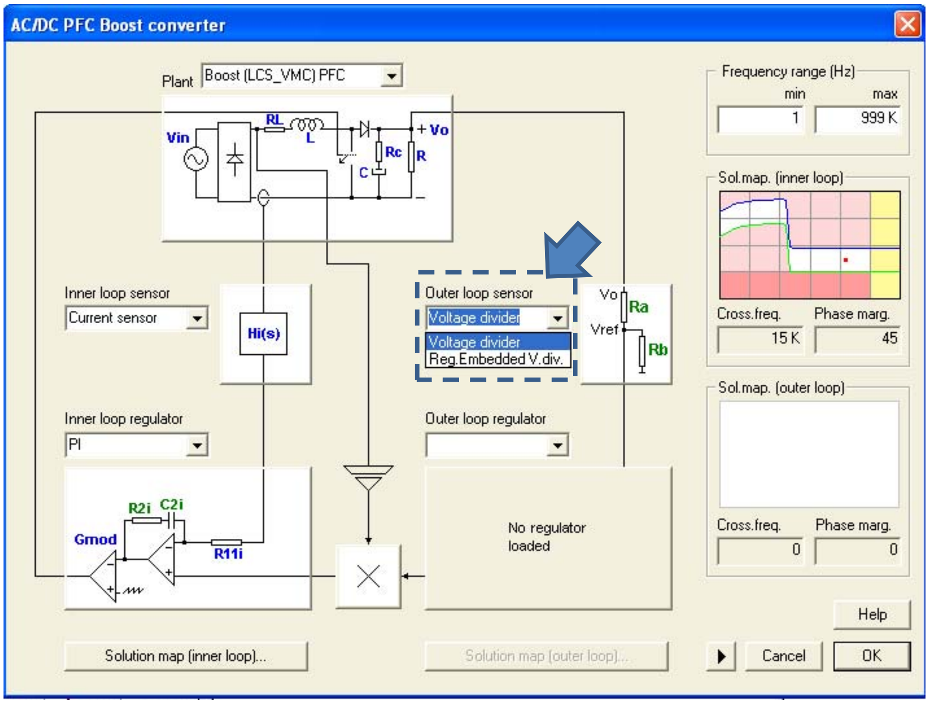


Once the inner loop design is completed, one can move on to the outer loop design.

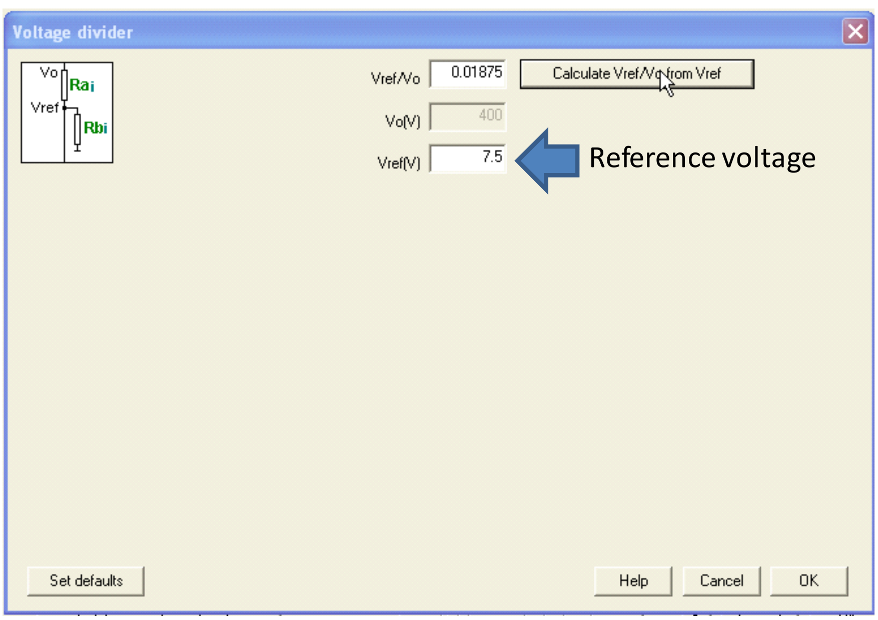
Outer Loop Design

The procedure of designing the outer loop is similar to that of the inner loop design. It includes the following:

1. Select the voltage sensor



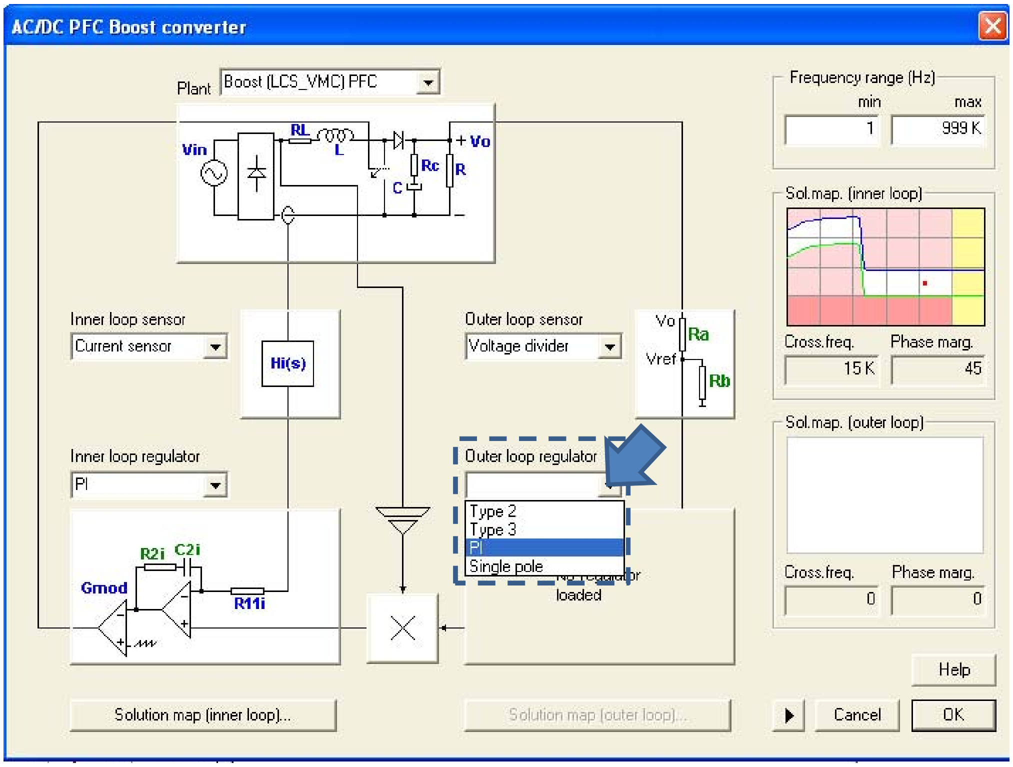
When using a voltage divider, one must enter the reference voltage, and the program will automatically calculate the sensor gain. In this example, the reference voltage is 7.5V. The sensor input data window is shown below.

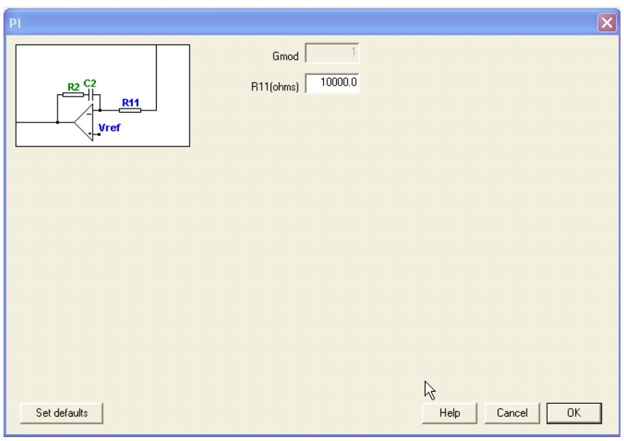


Then press on “Calculate Vref/Vo from Vref” button to set the attenuation of the voltage divider.

2. Select the outer loop regulator

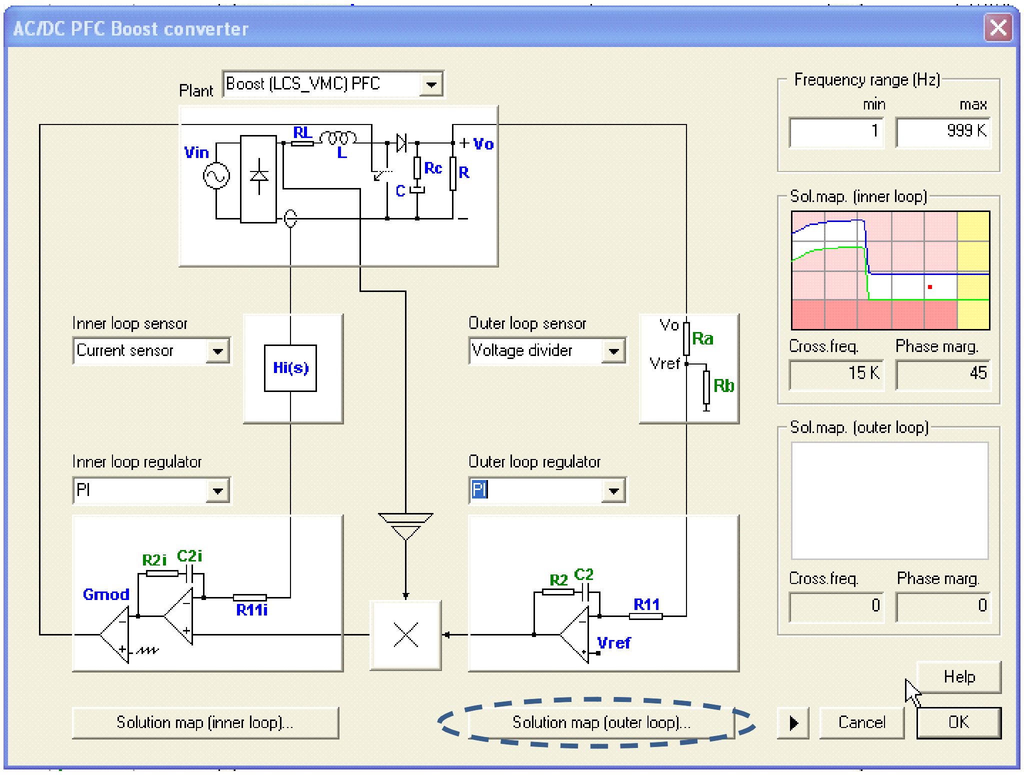
   In this example, select **PI** as the regulator type, and enter the regulator parameters as below.



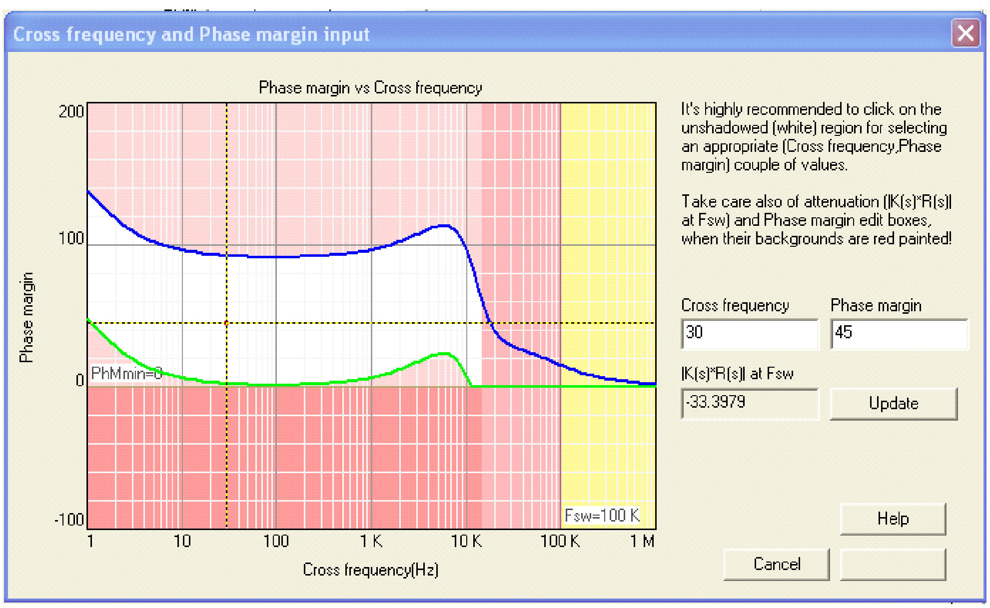


3. Determine the crossover frequency and the phase margin

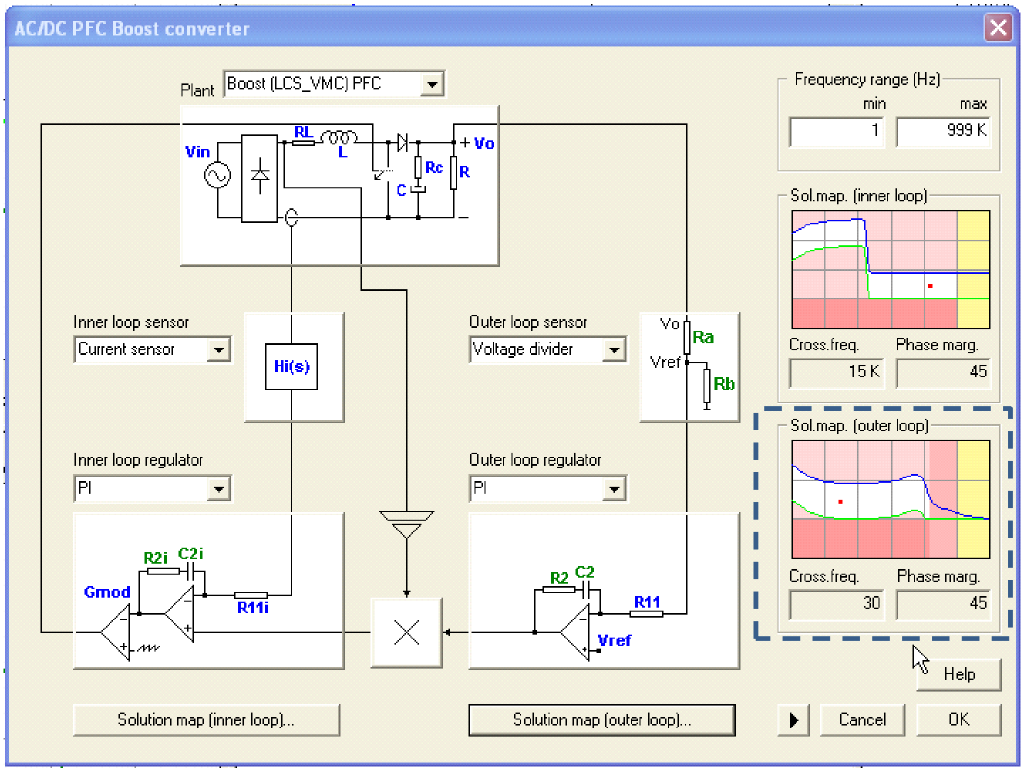
   Similar to the inner loop design, the crossover frequency and the phase margin of the outer loop must be selected. A Solution Map is also provided to help select a stable solution. Press the **Solution map (outer loop)** button and the solution map will appear.



Then select a point by clicking within the white area, and click OK to continue.

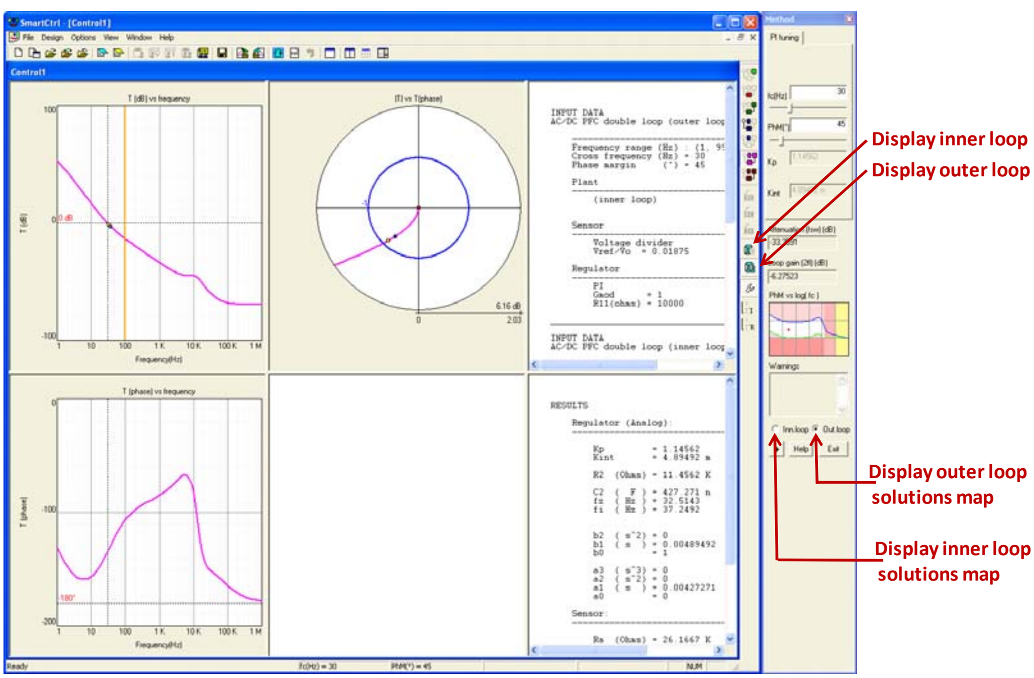


Once the crossover frequency and the phase margin are selected, the Solution Map will appear on the right side of the converter input window. If, at any time, these two parameters need to be changed, click in the white area of the Solution Map, as shown in the figure below.



Accept the selected design by clicking on OK. The program will automatically show the control system performance by means of the Bode plots, the Nyquist plot, phase margin, etc.

SmartCtrl provides the regulator component values needed to implement the regulators, as well as the voltage divider resistors. Since there are two control loops, one must select which one to display.



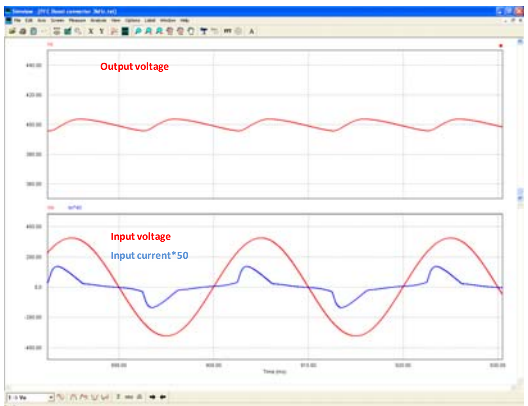
At this point, the control loop design of the converter will be completed. We will validate the design by means of time-domain simulation.

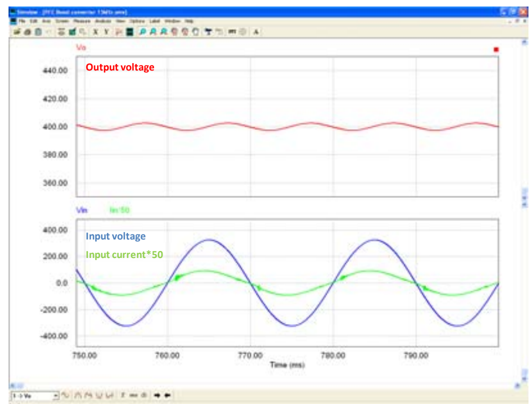
Design Verification through Simulation

In order to validate the design from SmartCtrl, time-domain simulation is carried out in PSIM.

We will compare the performance of the converter with two different designs.

Design #1		**Design #2**	
Inner loop: f_{cross}=3 kHz Phase Margin=45º	Outer loop: f_{cross}=30 Hz	Inner loop: f_{cross}=15 kHz Phase Margin=45º	Outer loop: f_{cross}=30 Hz





It can be observed that Design #2 shows a much less waveform distortion that Design #1, and that the waveform distortion can be greatly reduced by increasing the crossover frequency of the inner loop.

This example illustrates that SmartCtrl in combination with PSIM provide a fast and powerful platform for the design and validation of converter control for power factor correction applications.