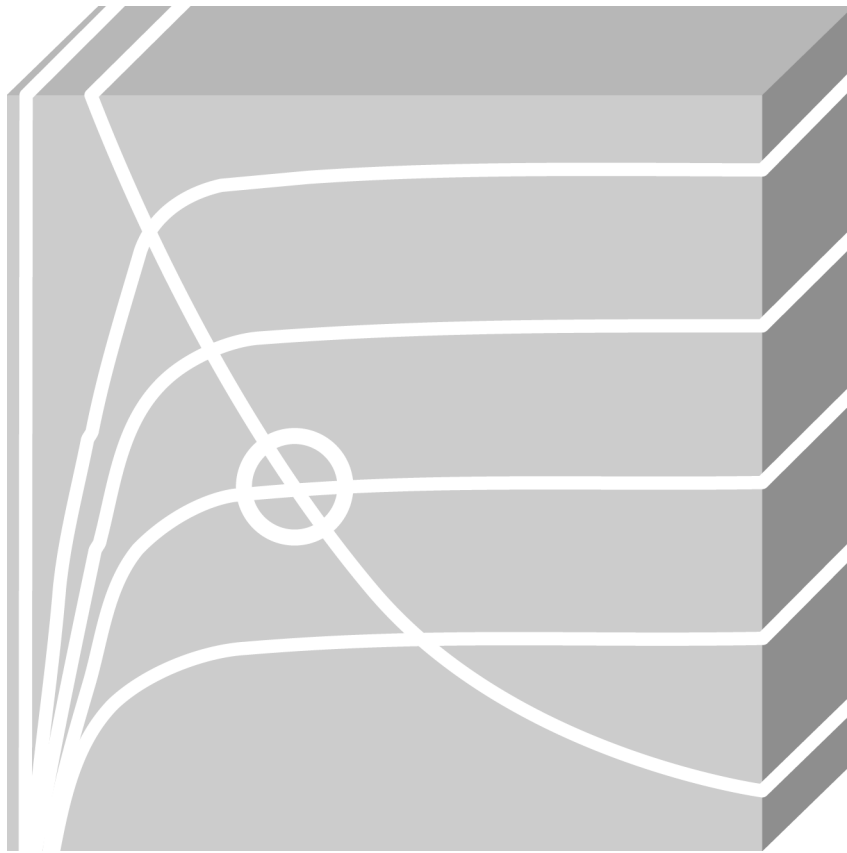


## **Configure PSpice Systems Option**



**Table of Contents**

1 Introduction ..... 3

2 Configuration for the SLPS Co-Simulation ..... 3

3 Use of SLPS Co-Simulation..... 5

4 MATLAB Visualization Interface ..... 10

5 MATLAB Functions in PSpice ..... 12

## 1 Introduction

This document explains how to configure PSpice Systems Option, it means, which paths are necessary to define in MATLAB and in PSpice so that the co-simulation works. Moreover, it will be explained with examples, how each feature works.

Before starting, it is necessary to comment what is possible to do with PSpice Systems Option. It includes four features:

1. PSpice MATLAB Visualization Interface
2. Evaluate MATLAB Functions in PSpice
3. Import MATLAB Modules in PSpice using DMI
4. Co-Simulation SLPS

### Software Requirements:

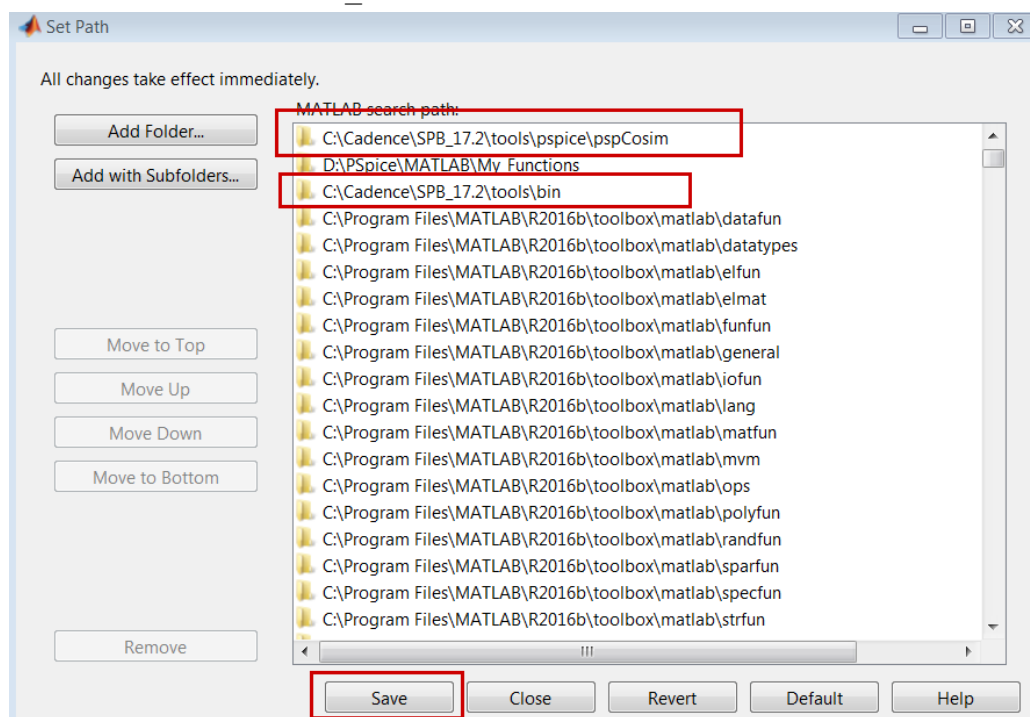
- MATLAB 2015b 64 bits or later
- OrCAD Capture 17.2
- OrCAD PSpice 17.2

## 2 Configuration for the SLPS Co-Simulation

If you want to co-simulate PSpice with MATLAB Simulink, follow next steps:

### MATLAB

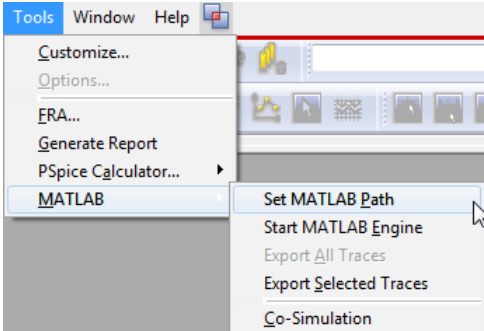
- Open **MATLAB R2015b** or later.
- Click on **Set Path** and define next directories:
  - <Installation>\SPB\_17.2\tools\pspice\pspCosim
  - <Installation>\SPB\_17.2\tools\bin



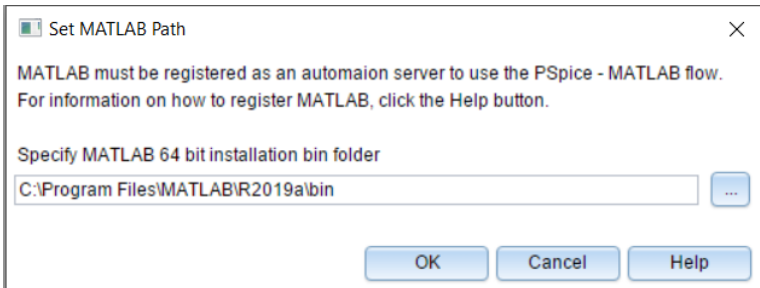
- Click on **Save**

## PSpice

- Open PSpice
- Click on **Tools > MATLAB > Set MATLAB Path**



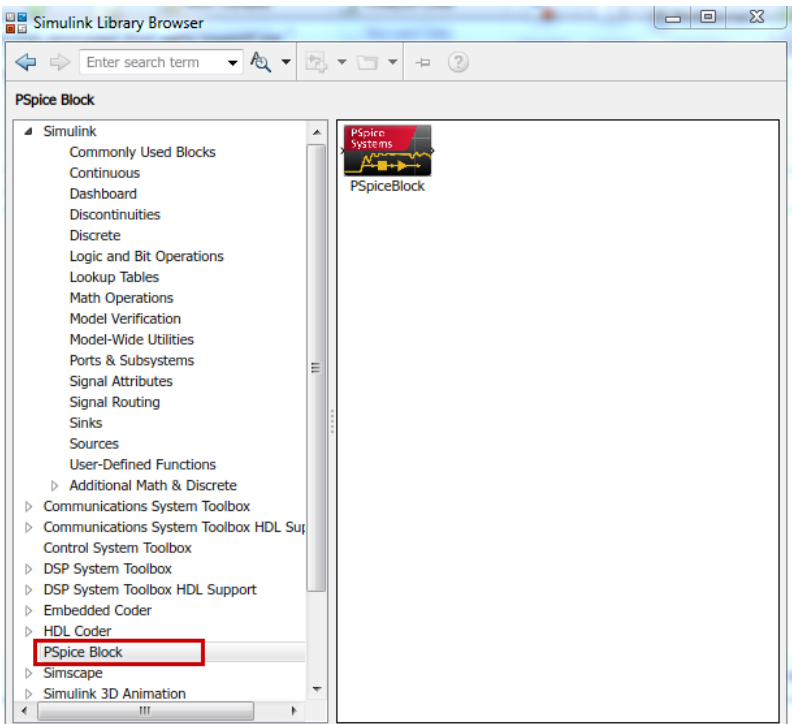
- Select the MATLAB Path.



- Open PSpice and click on **Tools > MATLAB > Co-Simulation**. Now you can simulate SLPS projects.

## Note

If you open right now a Simulink blank project, you will find the new SLPS Block in the Simulink Library:



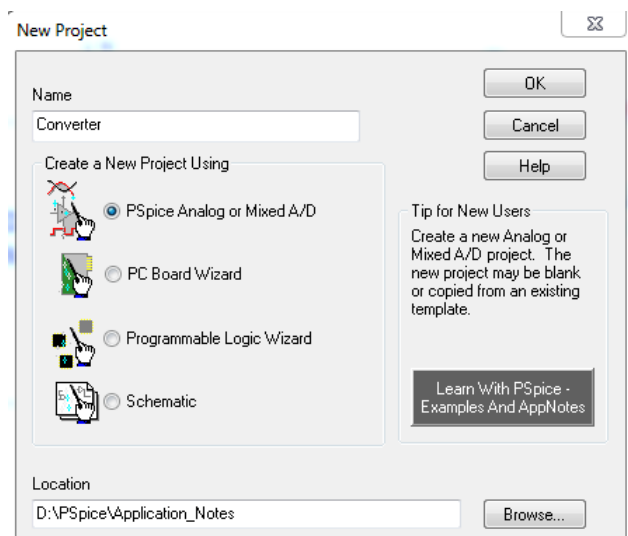
- **Open** a default **SLPS project** from the default Cadence examples and simulate it to test if it works properly.
  - <installation>\SPB\_17.2\tools\pspice\capture\_samples\CoSimulationDemos\PSpiceCoSim\RLC
- **Open** different **SLPS projects** from the folder **SLPS\_Examples** delivered with this Application Note.

## 3 Use of SLPS Co-Simulation

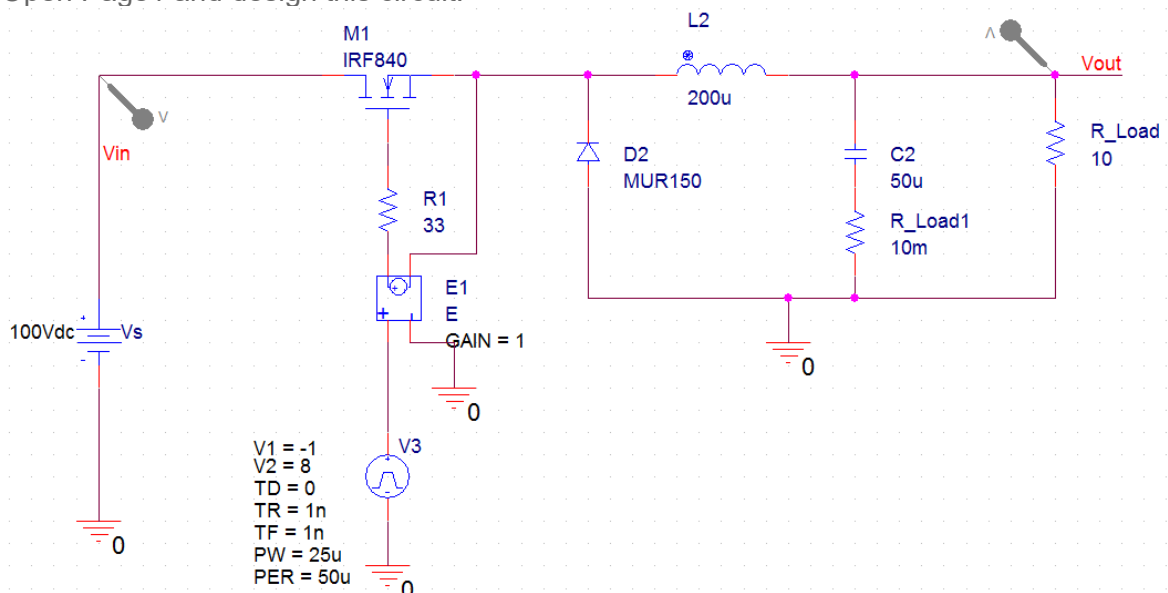
Once the configuration is done, you are going to create your first SLPS project. For that, follow next steps:

### OrCAD Capture Design

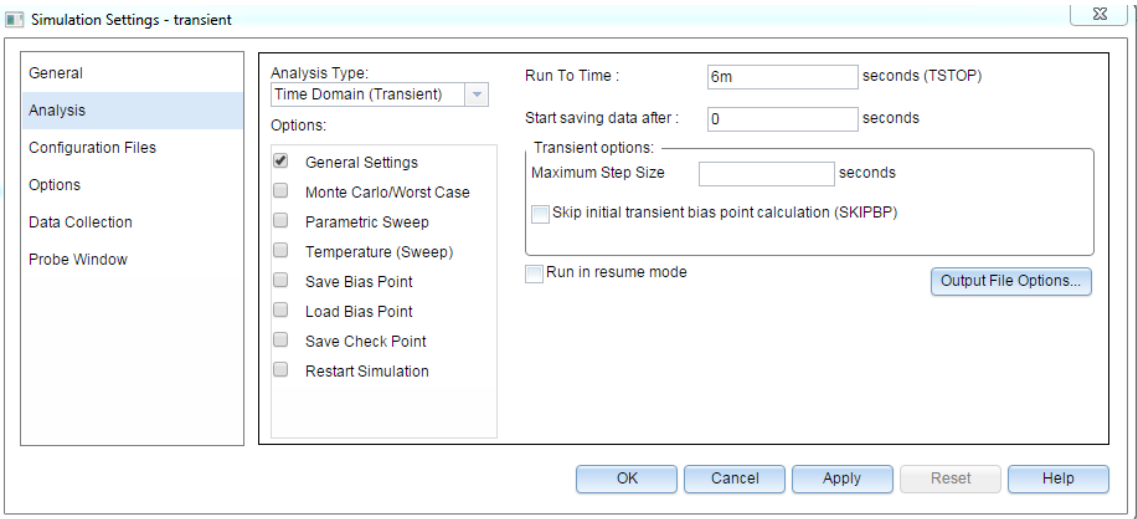
1. Open OrCAD Capture
2. Create a new blank project with the name **converter**:



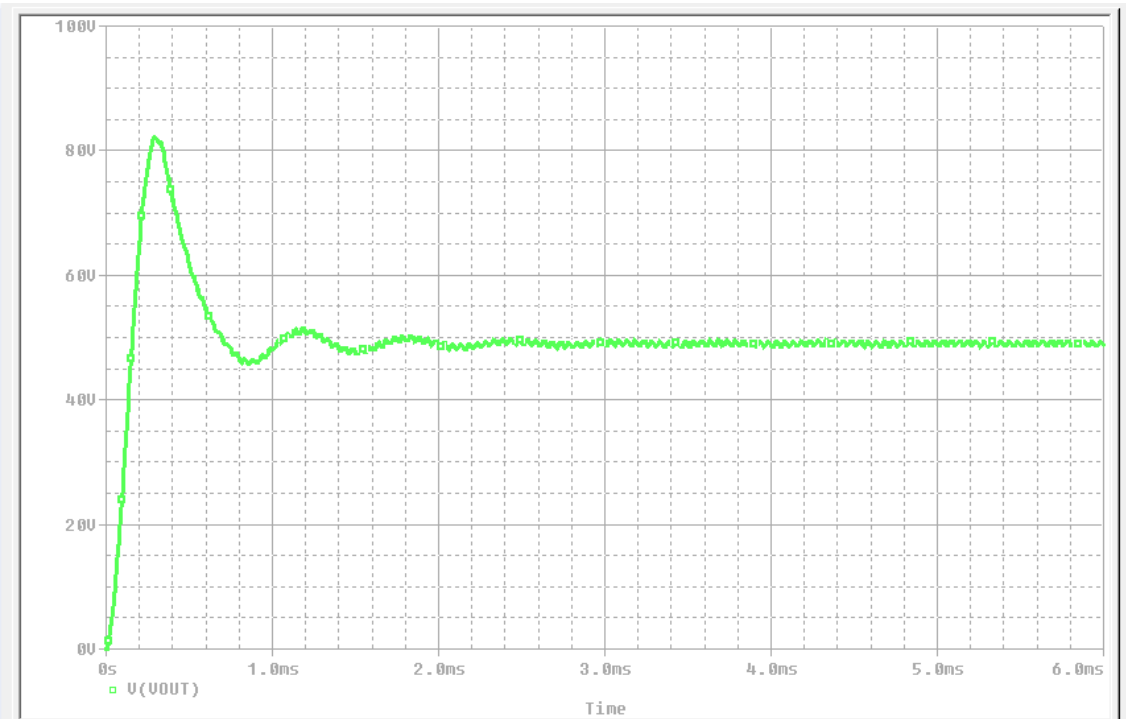
3. Open Page1 and design this circuit:



4. Create a new simulation profile and define it as in the image:

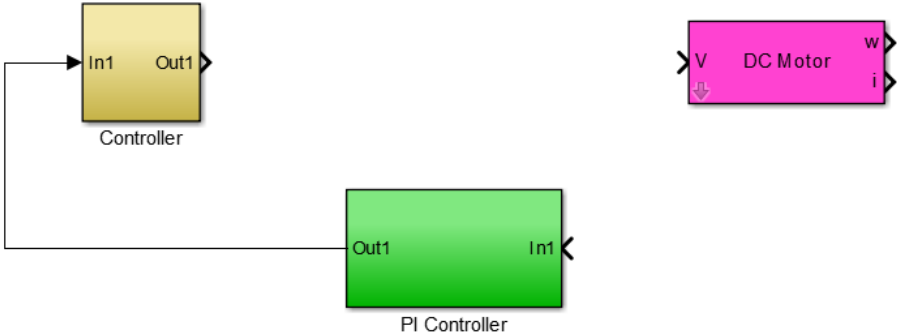


5. Click on OK and simulate. Place markers on the output pin.

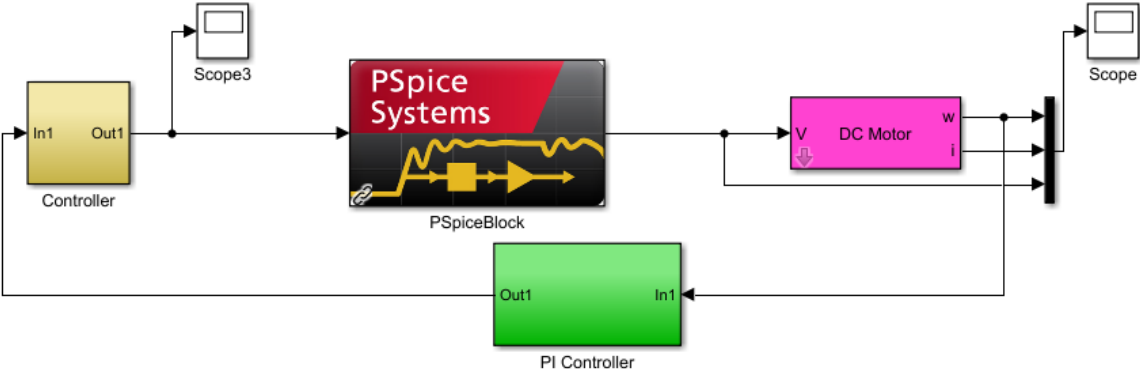


## Simulink Design

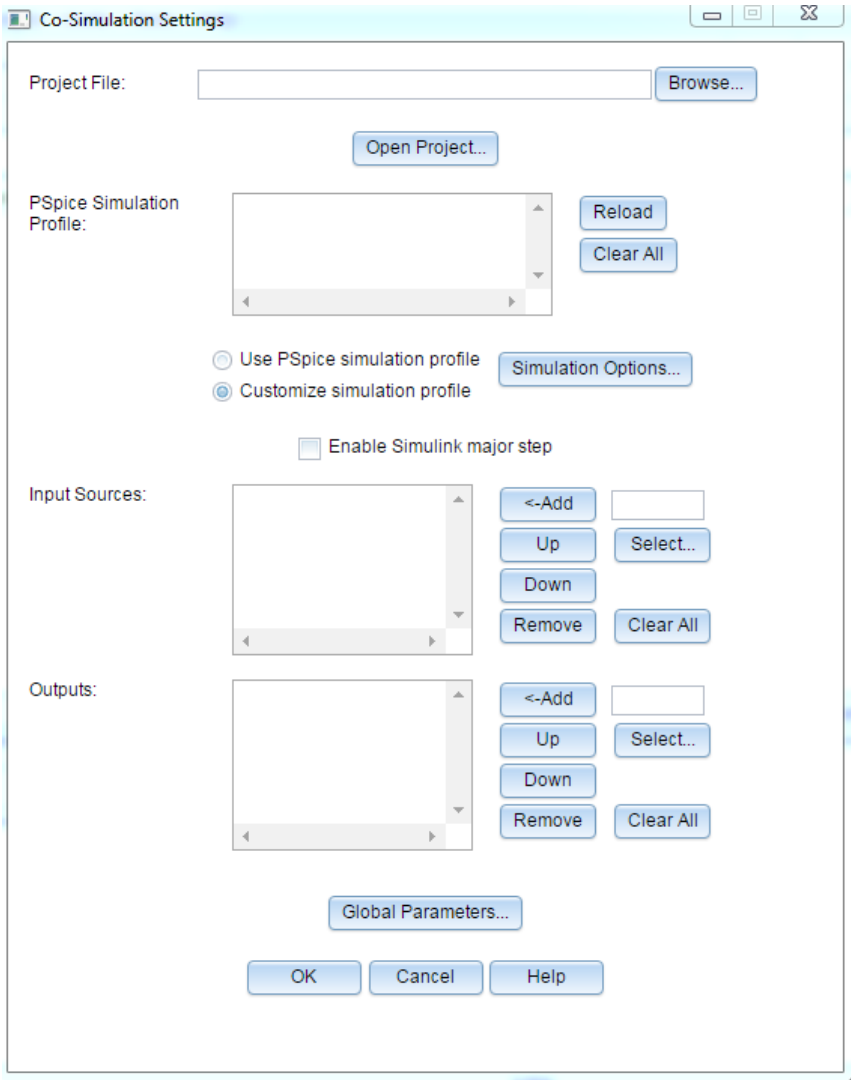
1. Open MATLAB.
2. Open the Simulink project slps\_start.mdl located under the folder **SLPS\_Start** in the file delivered with this AN.



3. Add SLPS block, Mux block and Scope. The Simulink design should look like that:

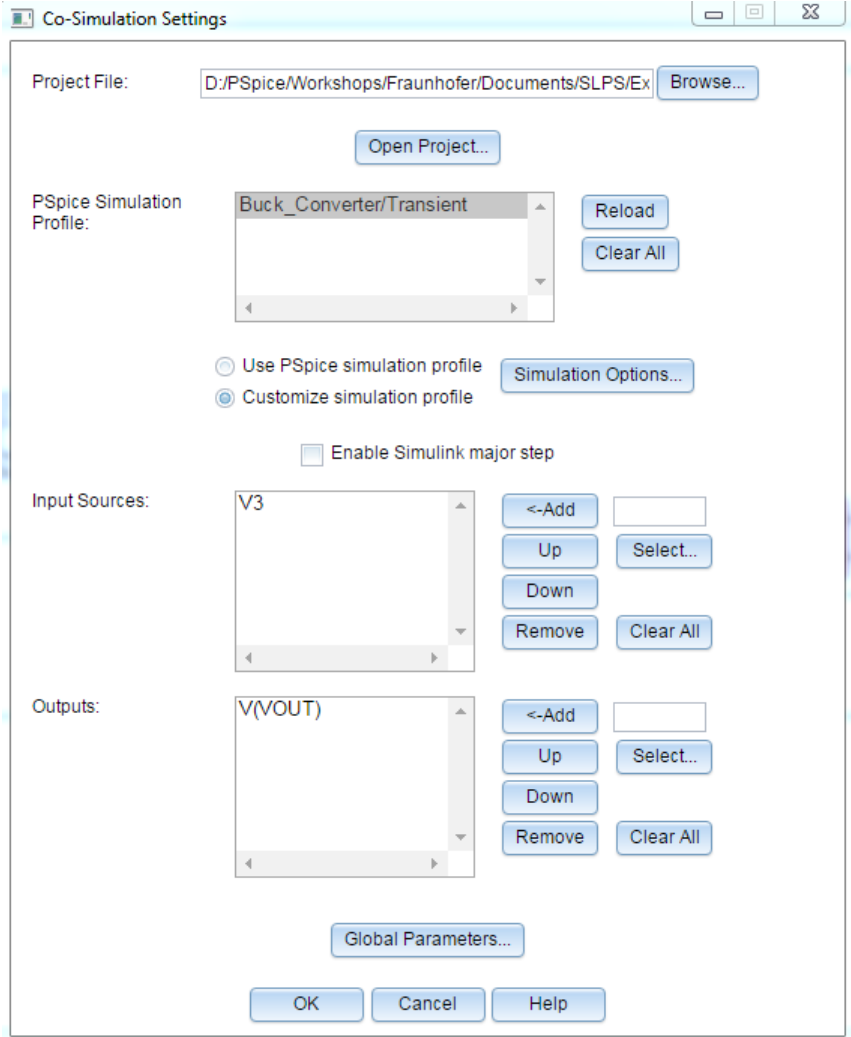


4. Make double click on the PSpice Block.

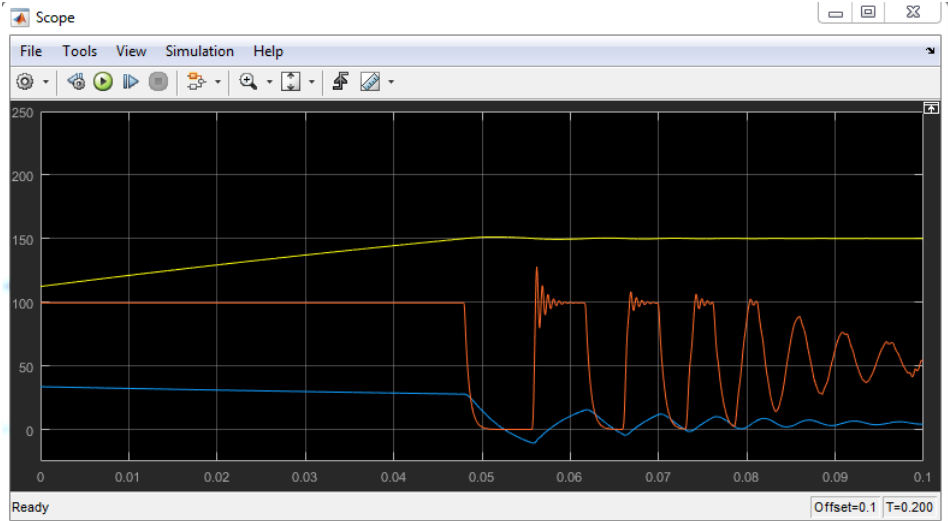




5. Browse your OrCAD Capture project (**converter.opj**).
6. Define Inputs and Outputs. We want to control the MOSFET of the converter with Simulink using a PWM signal and this controller must react depending on the output voltage.



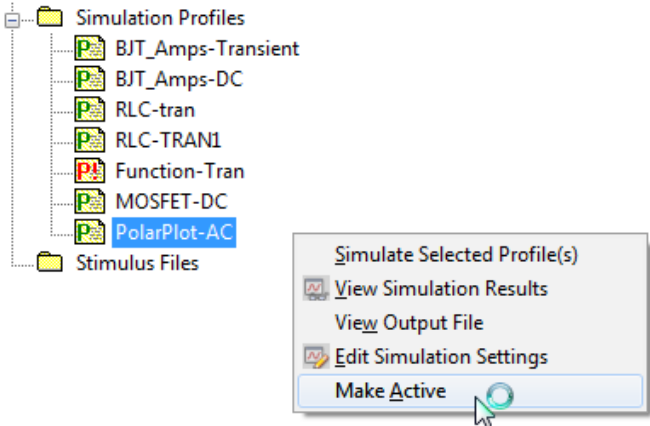
7. Click **OK** and simulate.



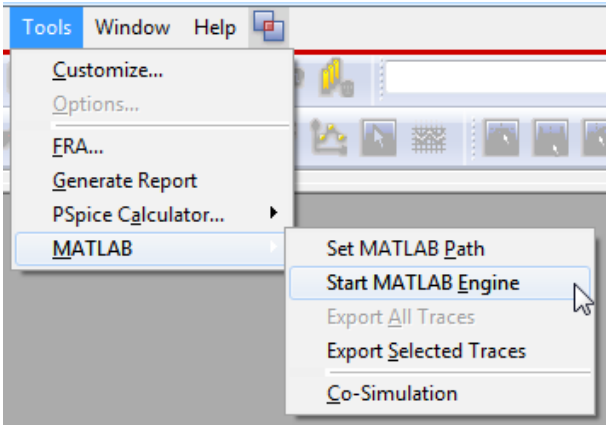
## 4 MATLAB Visualization Interface

With this feature you can use the MATLAB visualization capabilities for postprocessing and analysis of PSpice simulation results.

1. Open **OrCAD Capture** and load the project **bjt.dsn**.
2. Make Simulation Profile PolarPlot-AC active. Automatically the Schematic PolarPlot will be active.



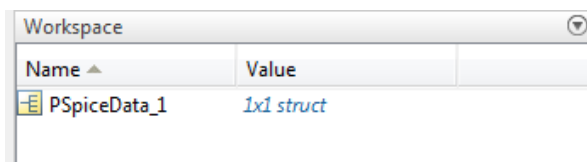
3. Open Page1 of the Schematic PolarPlot.
4. Open PSpice Probe Window.
5. Click on **Tools > MATLAB > Start MATLAB Engine**. A MATLAB command window will pop up. Depending on the computer it can take some time to be opened.



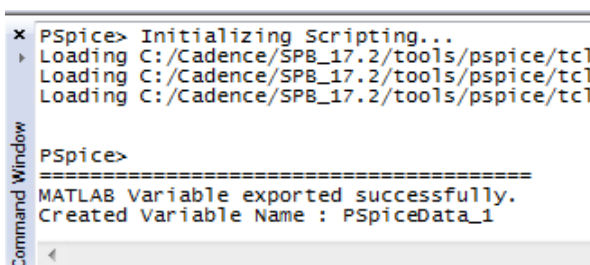
- Write Desktop in the command window to open MATLAB.



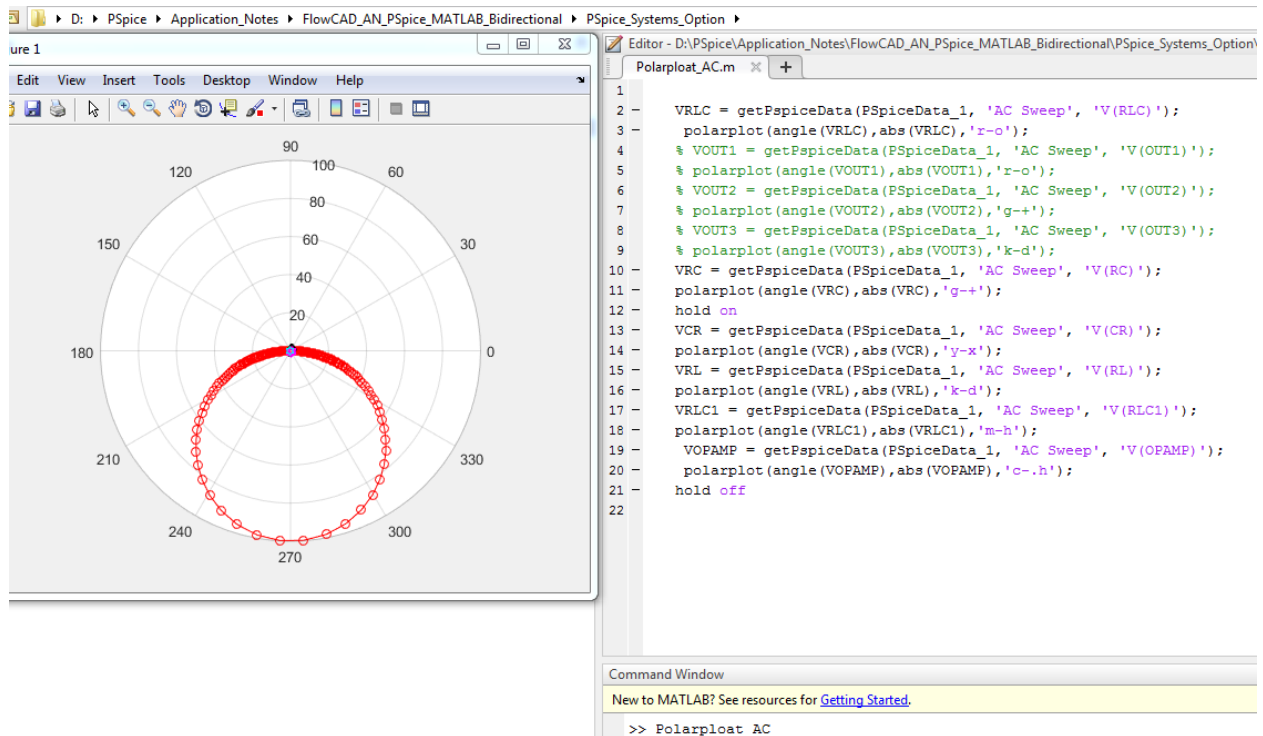
- Simulate the circuit.
- Click on **Tools > MATLAB > Export selected Traces**. Select next ones:
  - V(CR)
  - V(OPAMP)
  - V(RL)
  - V(RLC)
  - V(RLC1)
- Click on Export. Just close the MATLAB traces windows that appear when you click on Export. Automatically a Variable is defined in MATLAB. In this case, as we did not say anything, the name of it is PSpiceData\_1. Consecutive variables are generated when you export more traces.



You also get this information in the command window in PSpice.



- Once the PSpice results are already in MATLAB environment, change MATLAB Path to the location where you have downloaded and extracted the zip. The function Polarplot\_AC.m must be there.
- Write Polarplot\_AC in MATLAB and view the results. Even analyze how MATLAB take the results from the variable automatically generated by PSpice.



## 5 MATLAB Functions in PSpice

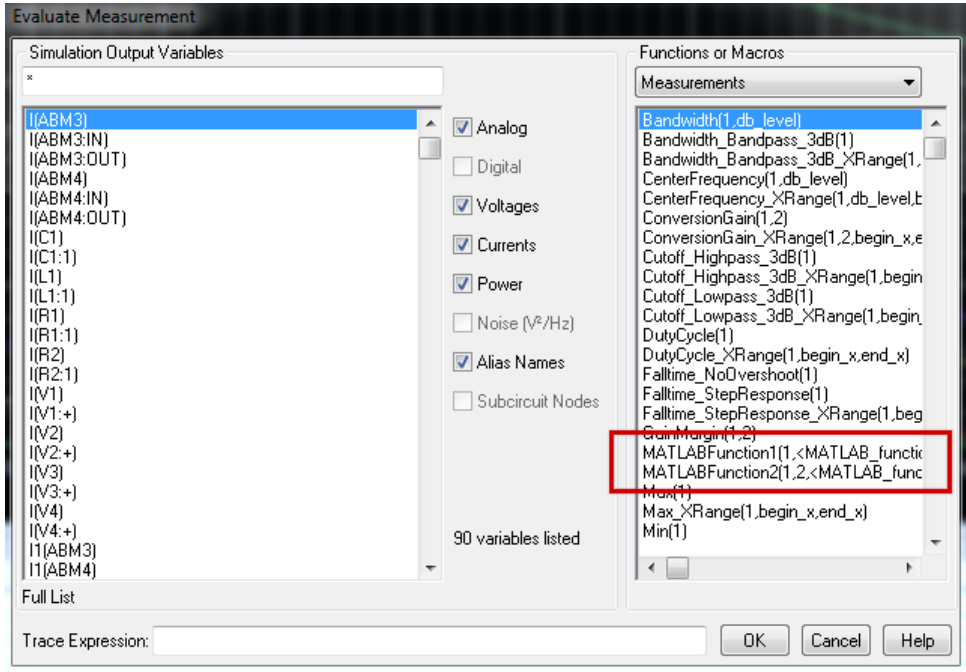
With these features you can use MATLAB functions in the PSpice environment for system simulation when the whole electronic design is not completely designed yet or even to do complex mathematical measurements.

1. Open **OrCAD Capture**.
2. Open the project **bjt.opj**.
3. Make simulation profile **Function-Tran** active.
4. Open Page1 of the schematic function.
5. Observe how MATLAB function were defined and also the syntax. It is possible to use own defined scripts (.m) and it is possible to use functions with more inputs. These symbols come from the library abm.olb (abm1, abm2, abm3).

### Note

If you use your own functions, the path where they are located must be defined in the MATLAB Path.

6. Simulate and observe the results in the probe window.
7. Also see the measurements, as it is also possible to use MATLAB Functions for measurements.



**Note**

For more information about these two features, use the documentation to open the pdf related to the PSpice MATLAB Interface.

