ALTAIR
ONLY FORWARD

SimCoupler Module

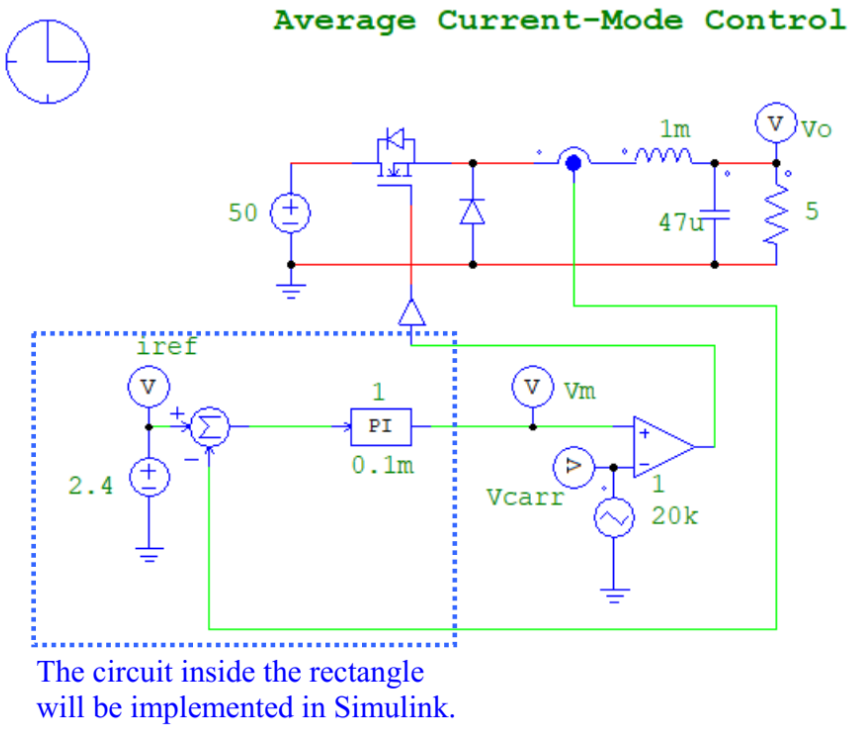
Altair PSIM Tutorial

Overview

The SimCoupler Module provides the link between PSIM and Matlab/Simulink* for co-simulation. The objective of this tutorial is to show how to use the SimCoupler Module and set it up in PSIM and Simulink.

A simple buck converter with current feedback control (file “chop1q_ifb.psimsch”), as shown below in the PSIM environment, will be used as the example. In this circuit, the inductor current is measured and compared with a reference. The error signal is sent to a PI controller, and the PI output is compared with a carrier waveform to generate the gating signal for the switch in the power circuit.

This tutorial illustrates how to split the circuit so that part of a control circuit (as shown below inside the dotted rectangle) is implemented in the Simulink environment, and the rest of the circuit stays in PSIM.



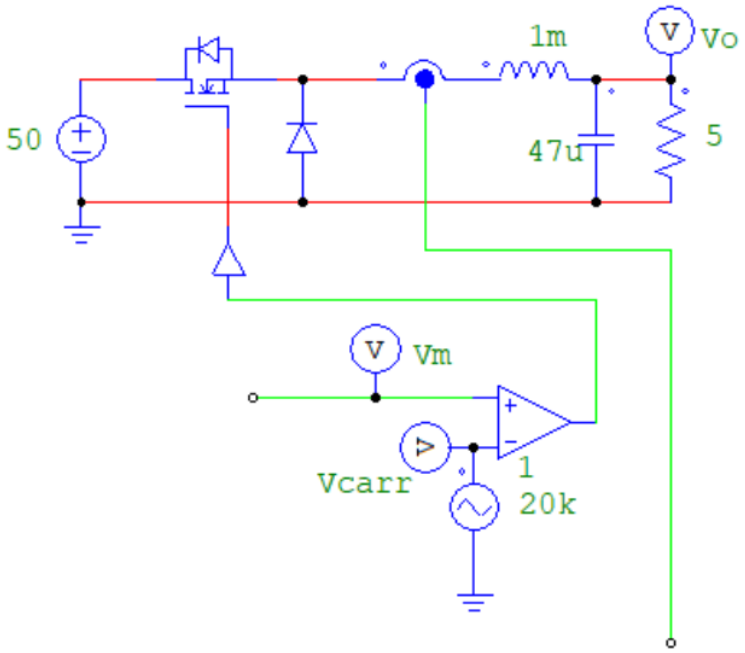
Matlab version R2018b is used for this tutorial.

** : Matlab and Simulink are registered trademarks of MathWorks, Inc.*

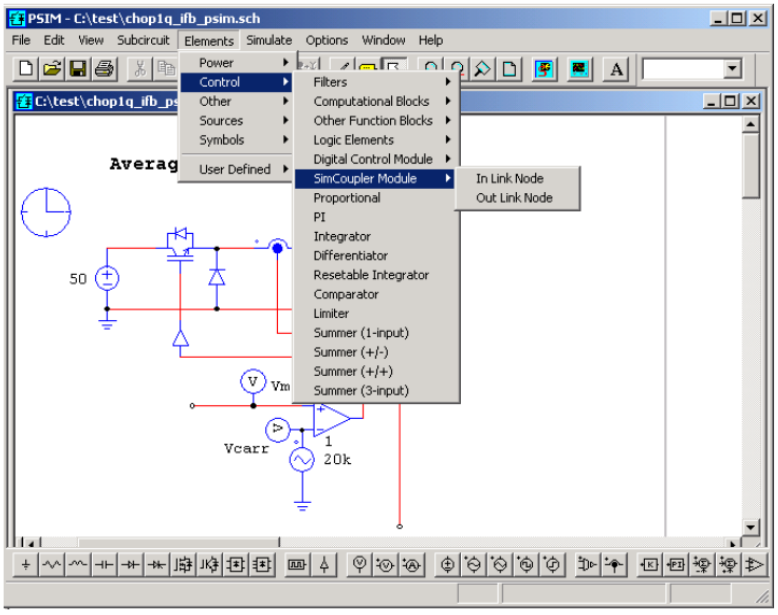
Setting up for Co-Simulation

Below are the steps to set up the co-simulation in PSIM:

- If multiple versions of PSIM are installed on the computer, in order to associate the PSIM version that you currently use to Matlab/Simulink for co-simulation, from the specific PSIM version, go to **Utilities** >> **SimCoupler Setup**. Note that this needs to be done only once until next time you need to associate another version of PSIM with Matlab/Simulink.
- Launch PSIM, and open the file “chop1q_ifb.psimsch”. The file can be found in the sub-folder “examples\SimCoupler” in the PSIM directory.
- Save the file to a different name, “chop1q_ifb_psim.psimsch”, in the directory “c:\test”.
- Modify the circuit by deleting the reference source, the summer, and the PI controller. After the modification, the circuit looks as follows:



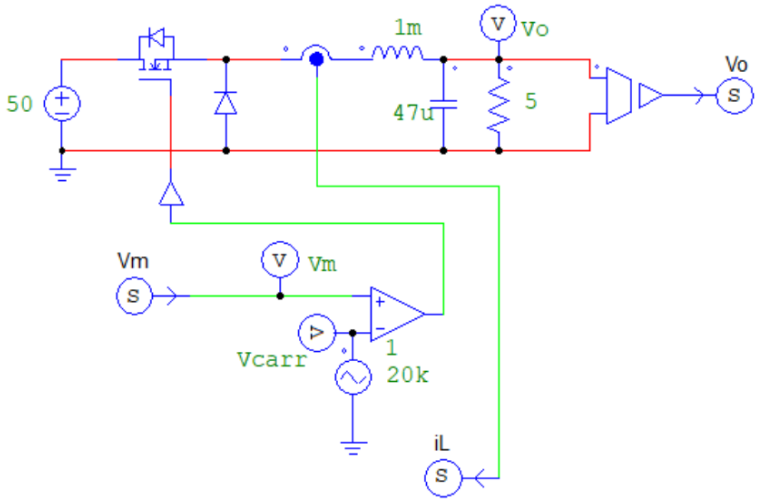
- Go to **Elements** >> **Control** >> **SimCoupler Module** as shown below:



Select the **Out Link Node**. Connect it to the current sensor output, and rename it to “iL”. Similarly, select the **In Link Node**. Connect it to the comparator input, and rename it to “Vm”.

The SimCoupler Module uses the SLink nodes to establish interface between PSIM and Simulink. In Link Nodes receive values from Simulink, and Out Link Nodes send values to Simulink.

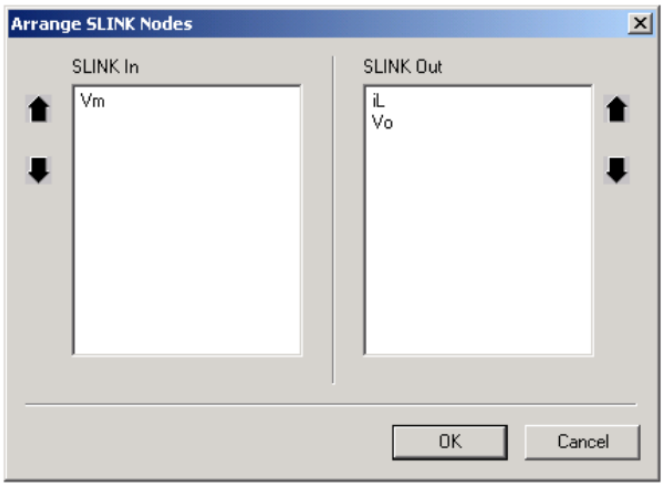
Multiple In/Out Link Nodes can be used in a circuit to exchange values between PSIM and Simulink. In this case, for example, we are going to measure and send the load voltage to Simulink by connecting a voltage sensor across the load resistor and placing an Out Link Node at the voltage sensor output. The Out Link Node will be renamed as “Vo”. After this, the circuit will look as below.



- If there are more than one In Link Node or Out Link Node (such as in this case), you may wish to arrange the order of the link nodes. Go to **Simulate** >> **Arrange SLINK Nodes**, and

a dialog window will appear as shown below. Arrange the order of the In nodes and Out nodes to be the same as how the input/output ports would appear in the SimCoupler model block in the Simulink environment. The order of the ports is from the top to the bottom. In this case, the output port corresponding to “iL” will be on the top, and the output port corresponding to “Vo” will be on the bottom.

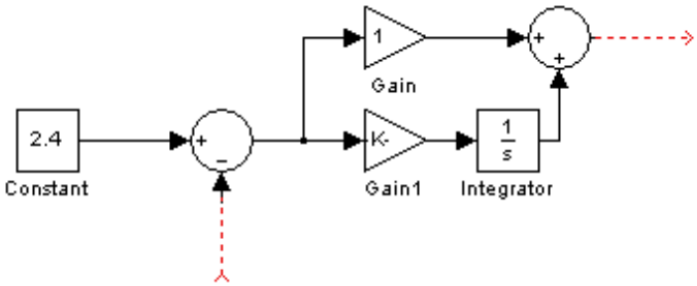
To re-arrange the node sequence, highlight the name of the node, and click on the up or down arrow.



- Save the schematic file. In this example, the file will be saved to “c:\test\chop1q_ifb_psim.psimsch”.

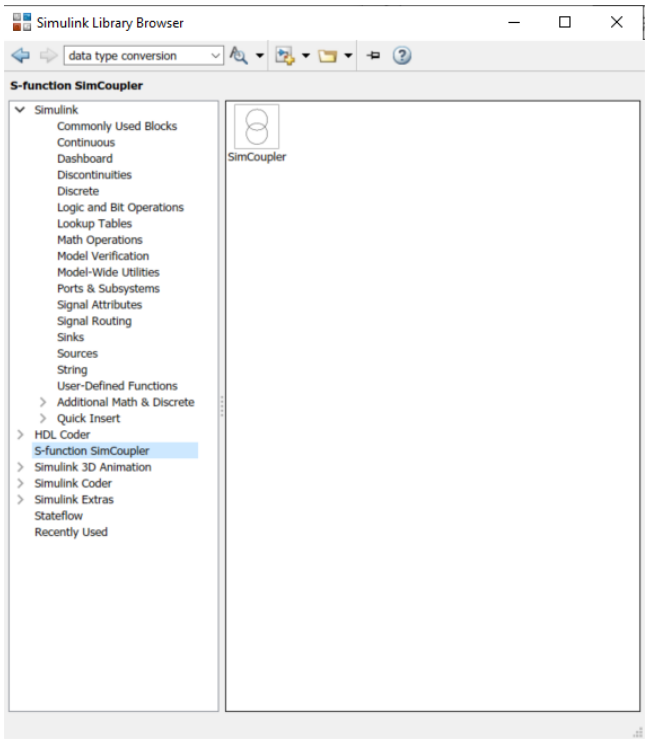
  At this point, the setup in PSIM is complete, and we will start the setup in Simulink.

- Start Matlab.
- Launch Simulink, and create a new file corresponding to the control circuit that was deleted from the PSIM circuit in Step 4, as shown below.

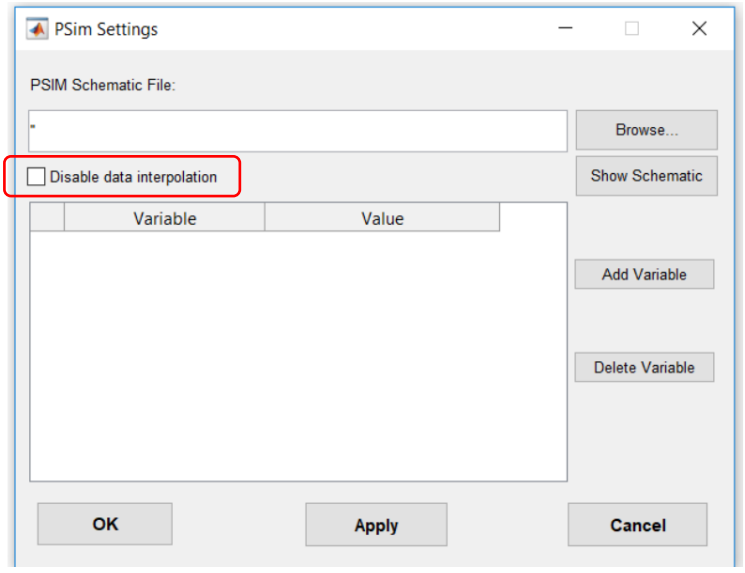


Save the file to “c:\test\chop1q_ifb_simulink.slx”.

- Go to the Simulink library browser, and you will see the SimCoupler block under the menu *S-function SimCoupler*, as shown below:



- Highlight the SimCoupler block, and drag it into the schematic of the file "chop1q_ifb_simulink.slx". Double click on the SimCoupler block, and the following dialog window will appear:



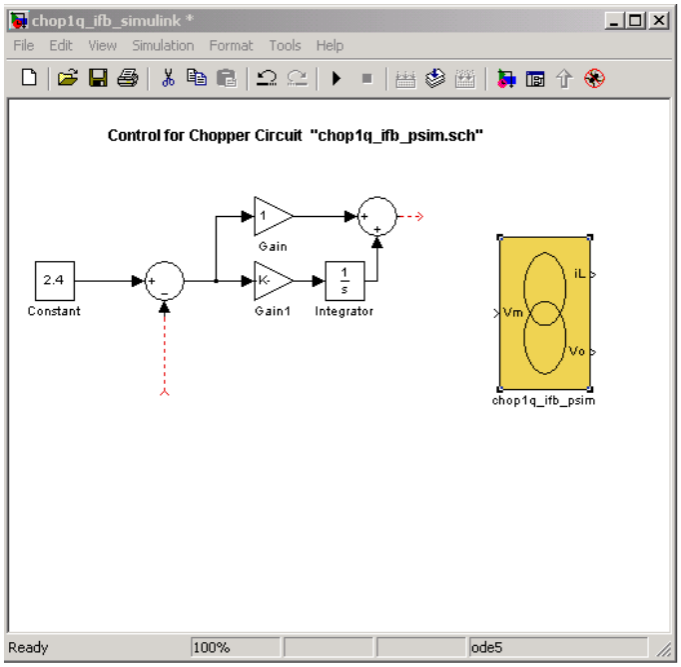
Click on the **Browse…** button to find and select the file "c:\test\chop1q_ifb_psim.psimsch". Click on the **Show Schematic** button will launch PSIM and display the schematic file. Click on **Apply** to accept the file, and then **OK** to close this window.

There is a checkbox "Disable data interpolation" in the SimCoupler block dialog window. By default, this box is unchecked. When it is unchecked, interpolation/extrapolation will be used when PSIM and Simulink exchanges data. When it is unchecked, interpolation/extrapolation will be disabled, and data from one program will be sent to the

other as they are without interpolation/extrapolation. When a logic signal is exchanged between PSIM and Simulink, this box may need to be checked.

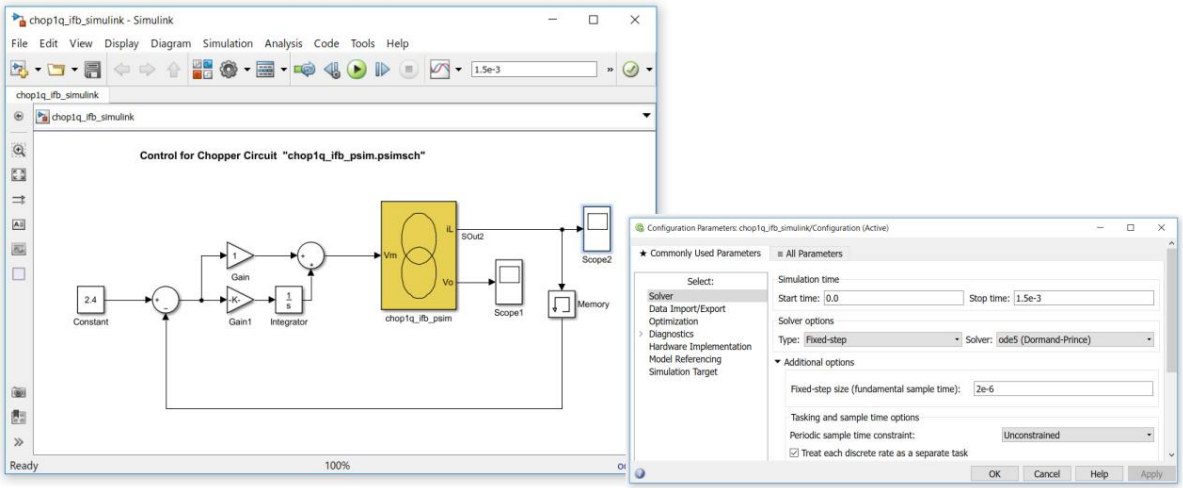
The image of the SimCoupler block will change to match the number of input/output ports to the number of IN/OUT Link Nodes as defined in the PSIM schematic. In this case, there are one input port (corresponding to the In Link node Vm) and two output ports (corresponding to Out Link nodes iL and Vo). The link node names will appear next to the input/output ports.

In order to differentiate the SimCoupler block from the rest of the Simulink elements, the block background color is changed to yellow.

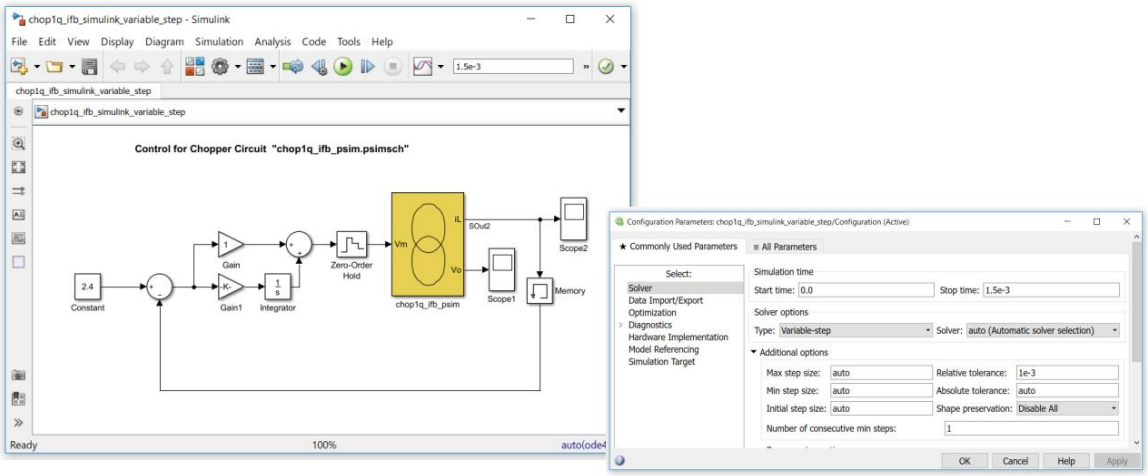


If the number of link nodes in the PSIM schematic is changed later, in Simulink, go to **Simulation** >> **Update Diagram** to update the SimCoupler block.

- Complete the connection in Simulink, then go to **Simulation** >> **Model Configuration Parameters**. Under the **Solver Options**, the *Type* can be set to either *Fixed-step* or *Variable-step*. If the fixed-step type is chosen, the fixed step size should be the same as or close to PSIM's time step (in this example, the PSIM's time step is 2us). The Simulink circuit will look as follows.



Alternatively, we can use the variable-step solver option. However, in this case, a zero-order-hold (ZOH) must be inserted to each input port of the SimCoupler model block, as shown below, and the sampling time of the ZOH must be equal or close to PSIM's time step (2us in this case). Please see additional remarks at the end of this document regarding the selection of the solver type.

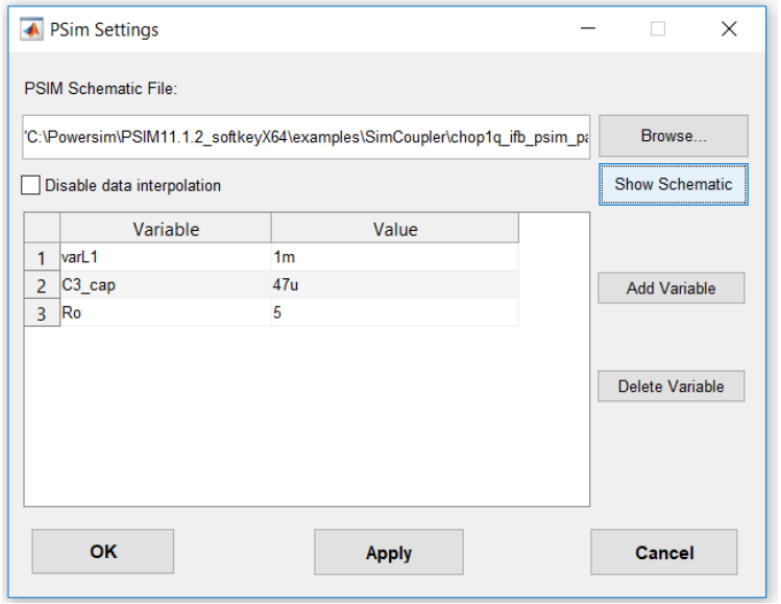


- The setup in Simulink is now complete for the co-simulation. In Simulink, go to **Simulation** >> **Run**, and start the simulation.

Passing Parameters from Simulink to PSIM

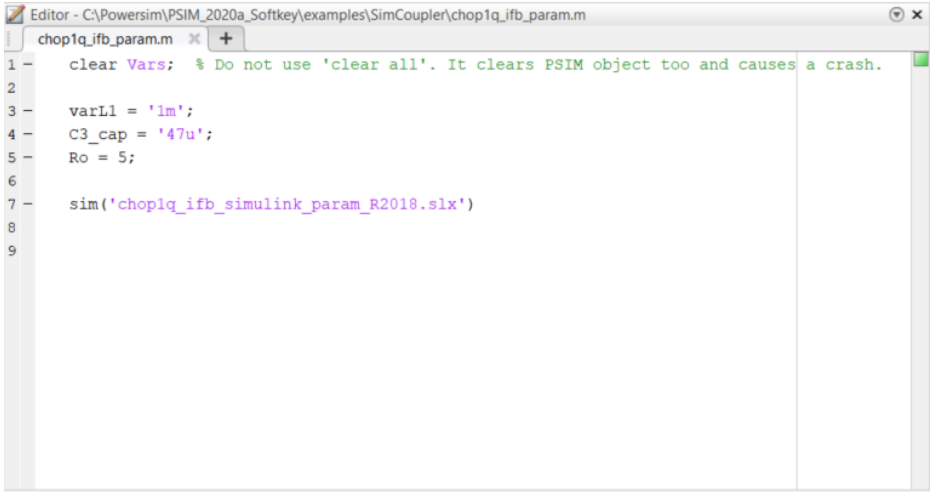
To set some parameter values in Simulink and pass them to PSIM, in the PSIM schematic file, change the desired value to a variable name. For example, to set the inductance of the inductor L1 in Simulink, change it to *varL1*.

In Simulink, double click on the SimCoupler block to open the property dialog, and click on **Add Variable**. Click on the newly added row in the list and type the variable name and value, as shown below.



Example files **“**chop1q_ifb_simulink_param_R2018.slx” and “chop1q_ifb_psim_param.psimsch” in the “examples\SimCoupler” folder demonstrate this feature.

It is also possible to set the variable values in a Matlab M-File as demonstrated in sample files “chop1q_ifb_param.m” and “chop1q_ifb_param2.m”, as shown below.



After starting the simulation in Simulink, one can view waveforms in both Simulink and PSIM.

Additional Remarks

- Each time that you use the SimCoupler Module from a different PSIM version, or if the Matlab folder has changed, you need to run the menu command **Utilities** >> **SimCoupler Setup** from PSIM.

  For example, if you have both PSIM v12 and PSIM v11 on the same computer, to use the SimCoupler Module in v12, run the command “SimCoupler Setup” from the PSIM v12 folder. Alternatively, if you wish to use the SimCoupler Module in v11, run “SimCoupler Setup” from the PSIM v11 folder.

  Similarly, if SimCoupler was set to work with your existing Matlab R2018b, and you just installed Matlab R2020a on the computer, To have the SimCoupler Module work with the new version, run “SimCoupler Setup”, and choose Matlab R2020a.

- When the SimCoupler model block is used in a feedback system in Simulink, the SimCoupler model block may be part of an algebraic loop. Some versions of Matlab/Simulink cannot solve the system containing algebraic loops, while others can solve the system but with degraded performance. To break an algebraic loop, place a memory block at each output of the SimCoupler model block. The memory block will introduce one time step delay.
- Certain restriction is imposed on the selection of the solver type and the time step in Simulink when performing the PSIM-Matlab/Simulink co-simulation. Simulink can be set up to have the Solver Type as either Fixed-step or Variable-step. When the Solve Type is fixed step, the time step must be the same or close to the PSIM time step. If the Solver Type is variable step, a zero-order-hold must be used at each input of the SimCoupler model block with the sampling time to be the same as or close to PSIM’s time step.
- Because Simulink and PSIM can have different time steps, it is not recommended to exchange logic signals (0 and 1) between the two programs. If logic signals need to be exchanged, one is recommended to check the box “Disable data interpolation”.
- By default, the checkbox “Disable data interpolation” in the SimCouple model block is unchecked, and interpolation/extrapolation is used when data are exchanged between PSIM and Simulink. This is to taken into account the fact that Simulink and PSIM may use different time steps, and their solution data points are not in synchronization with each other. However, in certain cases (for example when exchanging logic signals between PSIM and Simulink), users may wish to check the box “Disable data interpolation” and disable data interpolation so that data sent from one program to another are unchanged. This is possible when Simulink uses the fixed time step solver, and the time step of PSIM or Simulink is integer times the time step of the other program.