# Graphical System Design Guide to Power Electronics Co-Simulation with Multisim and LabVIEW

Updated, 11/1/2011

This is a guide to power electronics demonstration samples for Multisim/LabVIEW co-simulation. These new tools are intended to enable desktop development of power electronics control IP for LabVIEW FPGA, enabling you to develop/test/debug/validate before you compile to physical hardware. By exploring these samples, you can learn basic features of the co-simulation environment as well as the techniques you will need to accurately simulate LabVIEW FPGA control systems with your power electronics circuits. By exploring the samples and suggested exercises in this document with live interactive LabVIEW front panels, you can also enhance your intuitive understanding of the switch mode power electronics circuits and control systems.

The goal of these graphical system design tools is to significantly accelerate the development of inverter control systems, by enabling you to develop your exact deployment code on the desktop (without long FPGA compiles), and using formal software verification techniques (build each IP block to specification and perform functional verification testing on the desktop whenever changes or updates are made).

NI hopes the new tools will serve as a catalyst and platform for innovation and creativity and would like to invite you to expand and enhance the samples contained in this guide. We encourage you to develop, document and share through the online Power Electronics Development Center community site the following: 1. Power electronics circuit examples, 2. Power electronics control algorithms for LabVIEW FPGA, 3. Multisim power electronics component models, 4. Comparisons between physical test data and simulation results, and 5. Training materials or curriculum for industry professionals or university students.

In some cases, you may be able to negotiate discounted access to the NI hardware and software tools to reward and encourage the development and sharing of valuable material for the power electronics developer community. Please email proposals to NI Clean Energy Product Manager, Brian.MacCleery@ni.com.

**You are invited! Beta Program Opened in October 2011**

National Instruments would like to invite you to become a beta tester for our new Multisim/LabVIEW co-simulation tools, planned for release in 2012. The tools take advantage of our ability to simulate LabVIEW FPGA code on the desktop to simulate, test and tune your LabVIEW FPGA inverter control (or smart grid power analyzer or any other circuit related algorithms) on the desktop with high fidelity SPICE models before compiling/downloading to the NI RIO target. These desktop virtual prototyping tools enable you to debug your LabVIEW FPGA code using the full featured capabilities of LabVIEW for Windows and test and refine your code without the risk of damaging physical hardware.

To request access to the beta program software installers along with a 3-month evaluation license to (any or all) of the recommended software tools below, email the NI beta program coordinator, Nestor.Sanchez@ni.com. We ask all beta testers to please share your candid feedback on the tools, both

negative and positive. Please aggressively report any bugs or issues you encounter, let us know if you find gaps or missing capabilities and share with us your recommendations and creative ideas for improvement.

**Recommended Software Tools**

- Required: NI LabVIEW 2011, Full or Professional recommended (ni.com/labview)
- Required: NI LabVIEW Control Design & Simulation Module 2011 (help files)
- Required: NI Circuit Design Suite 12.0 Beta 0 (or higher) (ni.com/multisim)
- Recommended: NI LabVIEW FPGA 2011 (ni.com/fpga)
- Recommended: NI LabVIEW Real-Time 2011 (includes NI PID Toolkit) (ni.com/realtime)
- Recommended: NI SoftMotion 2011 f1 (or higher) (ni.com/motion)
- Recommended: NI Electrical Power Suite (to request access to the pioneer program, email jack.arnold@ni.com)
- Recommended: NI CompactRIO Waveform Reference Library (download here)
- Recommended: NI Veristand 2011 (ni.com/veristand)
- Recommended: NI Unit Test Framework (learn more)

# Table of Contents


Graphical System Design Guide to Power Electronics Co-Simulation with Multisim and LabVIEW ..... 1

Background ..... 4

Introduction to Co-Simulation ..... 5

RLC Circuit ..... 6

TRIAC Light Dimmer ..... 14

Simple Chaotic Circuit ..... 23

Flyback Converter ..... 35

Boost Converter ..... 41

Buck Converter (Closed Loop PID Control) ..... 50

3-Phase Single Level Inverter ..... 65

Brushed DC Motor H-Bridge (Basic) ..... 89

Brushed DC Motor with Gate Drive Circuit ..... 104

Brushless DC Motor with Hall Sensors (Trapezoidal Flux) ..... 108

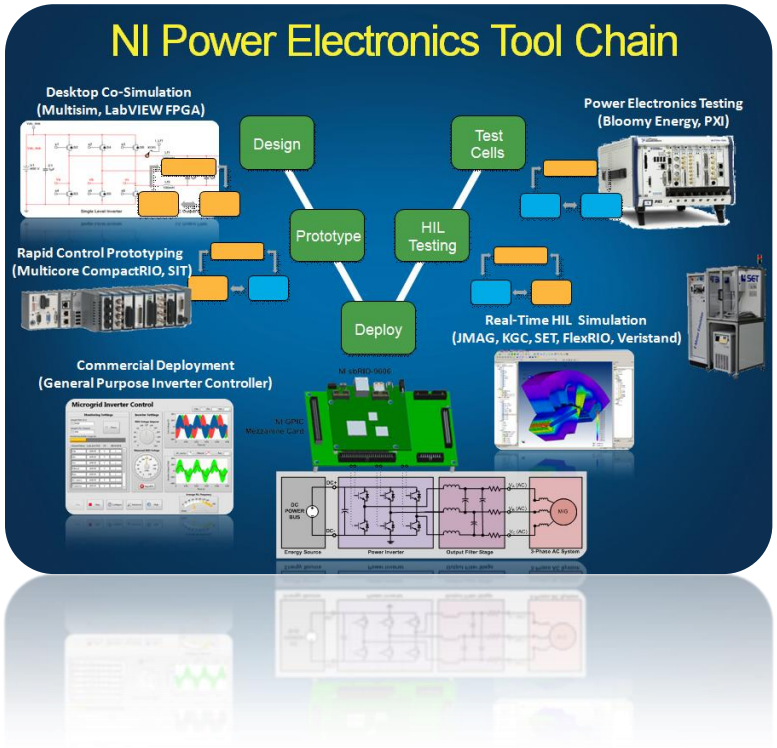
Invitation: Please Join the National Instruments Circuit Design Community! ..... 119

## Background

National Instruments is making major investments in power electronics with the goal of creating a complete platform for system level graphical design, desktop co-simulation, rapid control prototyping, volume commercial deployment embedded systems, real-time HIL testing/validation, physical test cells, remote smart grid integration and field upgradability. For a preliminary datasheet for the NI Single-Board RIO General Purpose Inverter Control (GPIC) system (shown below), please email Clean Energy Product Manager, Brian.MacCleery@ni.com.

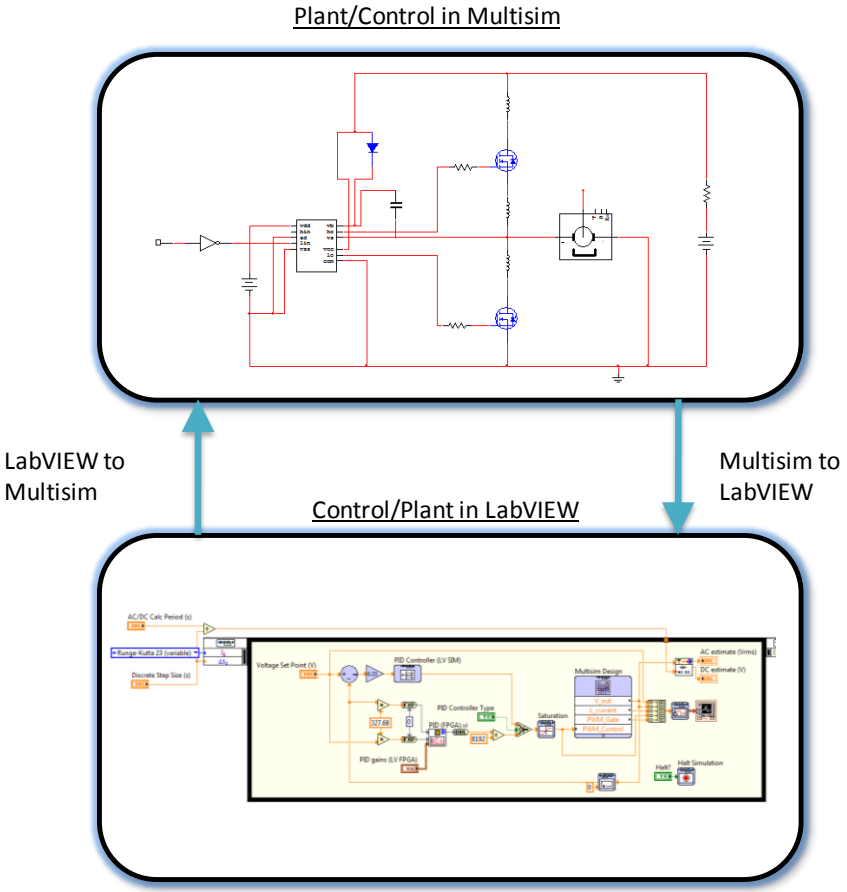
**Key Applications**

- Grid tied solar inverters
- Wind turbine power converters
- Utility scale energy storage systems
- Medium voltage motor drives and pumps
- Electric and hybrid electric vehicles, automobiles, trains and industrial/agricultural vehicles



## Introduction to Co-Simulation

Co-simulation applications are typically developed in the following manner. First, the power stage circuitry (the plant of the control system) is designed in Multisim– a SPICE-based circuit design tool. Then the LabVIEW code to control the circuit is developed, placed inside of a LabVIEW Control Design and Simulation loop and connected to the Multisim circuit for co-simulation using a Multisim Design interface block. Co-simulation is executed as a large-signal, time-domain transient simulation in which Multisim simulates the circuitry and LabVIEW executes the control system algorithms. The two simulators typically exchange data in a coordinated, variable time step manner, in which both solvers obtain convergence around an accurate simulation result, even in the case where coupled dynamics exist between the Multisim and LabVIEW parts of the system. Thus, it is also possible to include additional plant model dynamics, such as mechanical or thermal models, in the LabVIEW Control Design and Simulation loop and obtain accurate simulation results for the overall system. In LabVIEW, these simulation subsystems are typically expressed in state-space, transfer function or differential equation format.



## RLC Circuit

**Co-Simulation Concepts:**

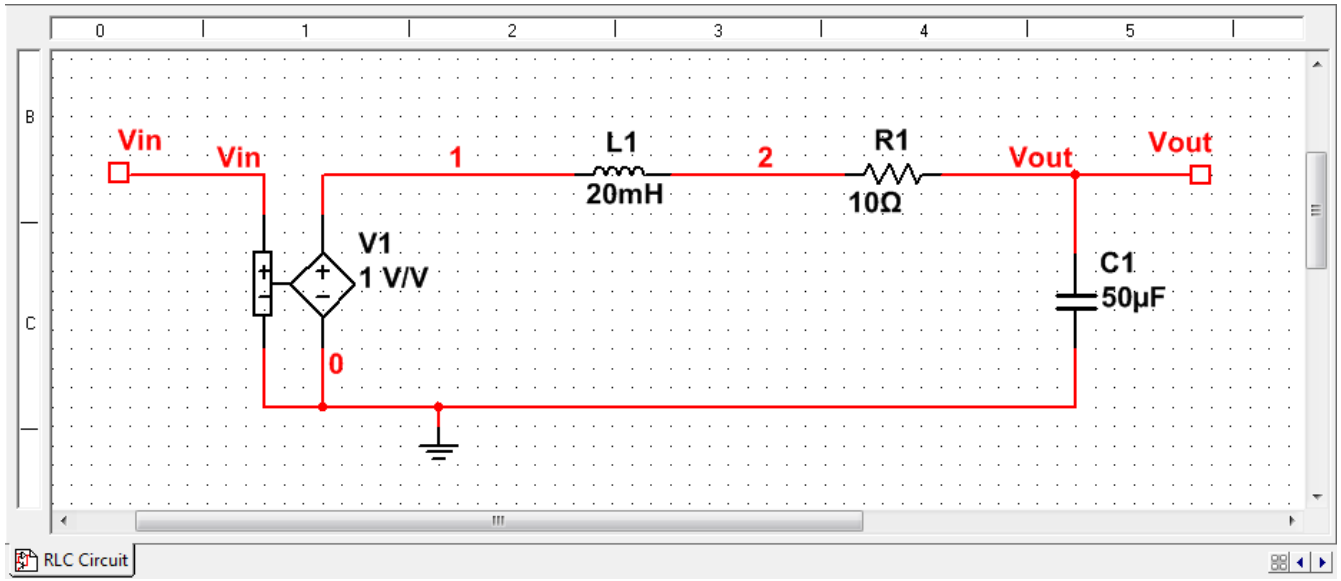
- Adding LabVIEW Co-simulation Terminals (known as **HB/SC connectors**) to your Multisim schematic
- Adding a **Multisim Design** interface block to your LabVIEW Control & Simulation Loop
- Creating a LabVIEW graphical user interface to interact with a running co-simulation experiment

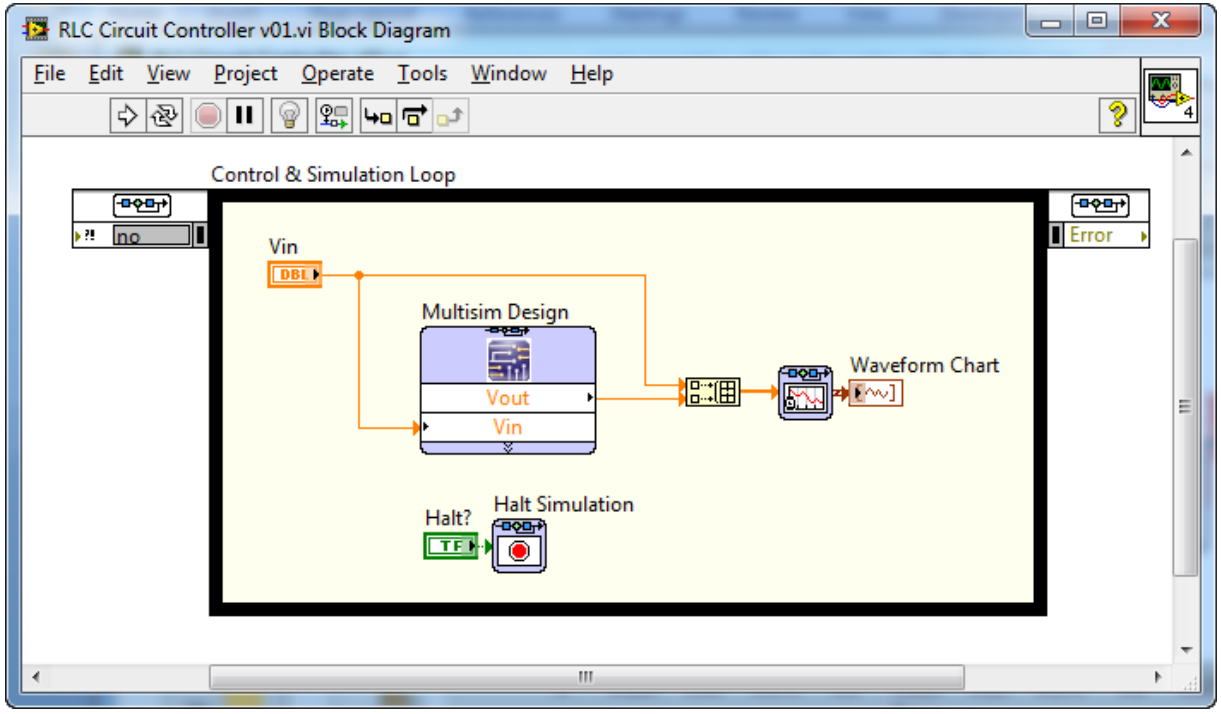
**Power Electronics Concepts:**

- Fundamentals of RLC circuits (learn more)
- Understanding second order system response characteristics including damping, natural frequency, rise time and percent overshoot
- Understanding the transient voltage and current characteristics of basic energy storage circuit elements, inductors and capacitors

**Required Toolkits or Modules:**

- NI LabVIEW 2011 (ni.com/labview)
- NI LabVIEW Control Design & Simulation Module 2011 (help files)
- NI Circuit Design Suite 12.0 Beta 0 (or higher)(ni.com/multisim)


Vin
Vin
1
L1
20mH
2
R1
10Ω
Vout
Vout
V1
1 V/V
C1
50µF
0
RLC Circuit


RLC Circuit Controller v01.vi Block Diagram
File Edit View Project Operate Tools Window Help
Control & Simulation Loop
no
Error
Vin
DBL
Multisim Design
Vout
Vin
Waveform Chart
Halt?
TF
Halt Simulation

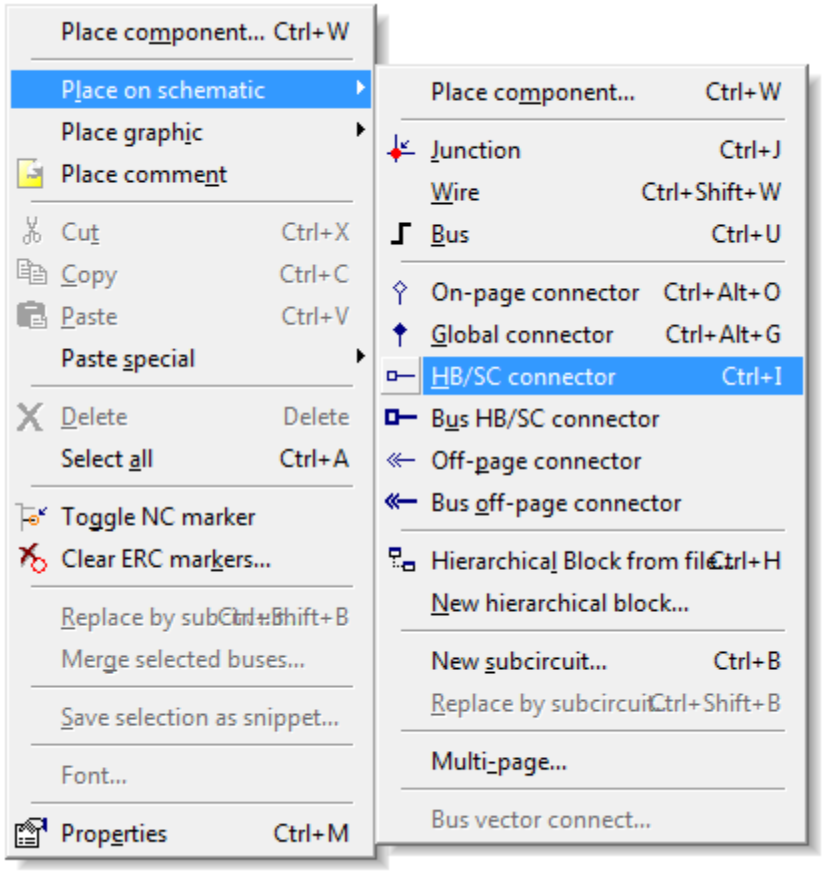
**Simulation Results**

- The RLC circuit demonstrates second order system behavior that can be characterized by rise time, percent overshoot and steady state error in response to changes in the **Vin** voltage, acting as a damped harmonic oscillator (learn more).

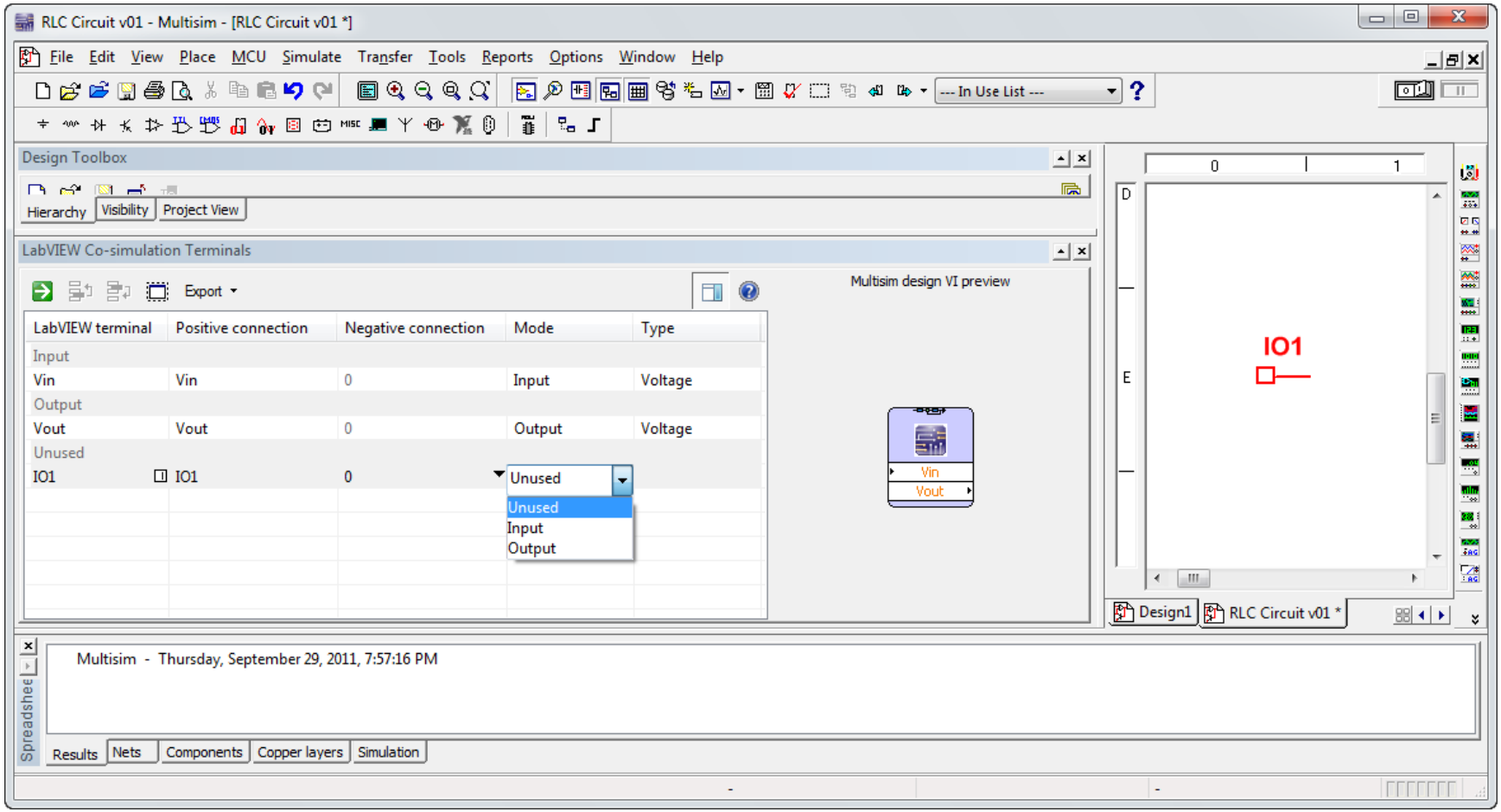
**Suggested Exercises**

- Recreate the example above from scratch.
- Instructions and hints:
  - If you make changes to your Multisim schematic, you must save the file for the changes to be reflected in the simulation.
  - To place a new LabVIEW Co-simulation Terminal, known as HB/SC connectors ( 

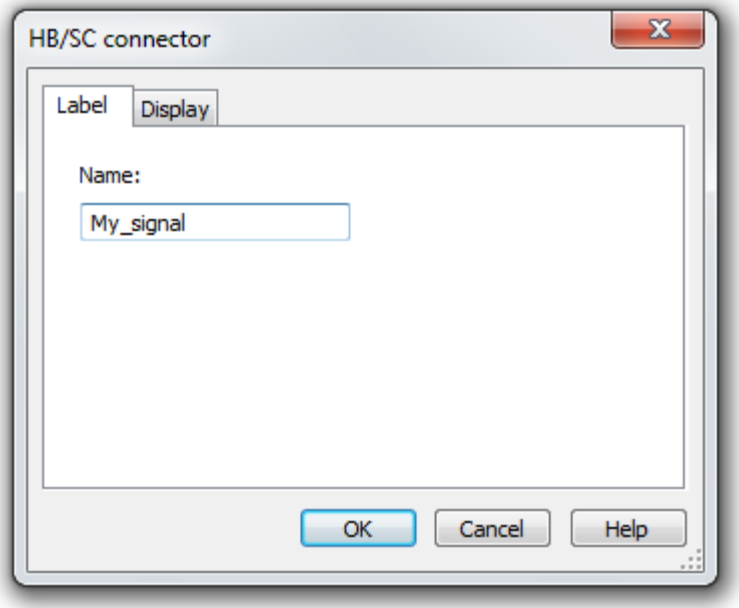
), right-click on any blank area of the schematic and select **Place on schematic>HB/SC connector**.



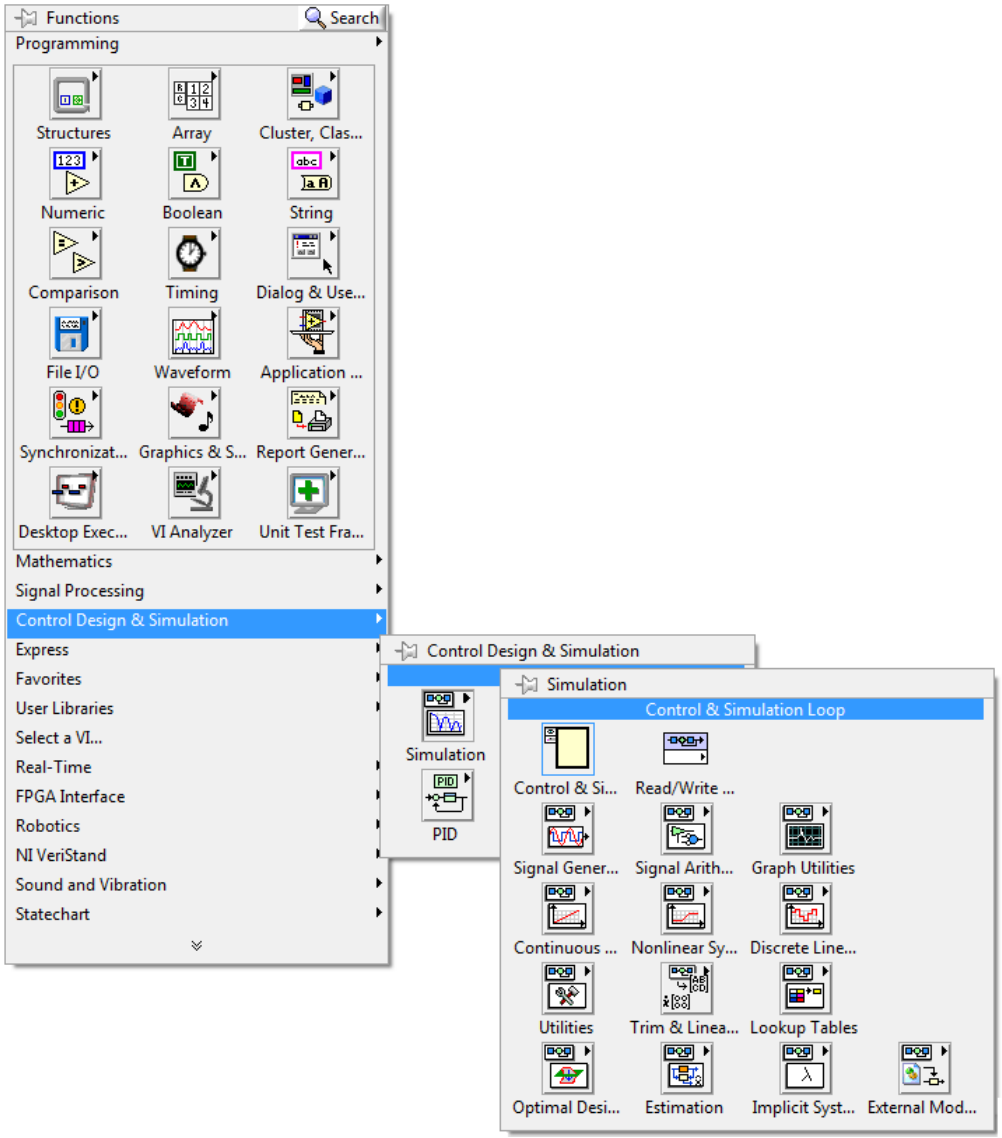
- Next, in the LabVIEW Co-simulation Terminals pane to the left, select the mode for the HB/SC connector (Input or Output).



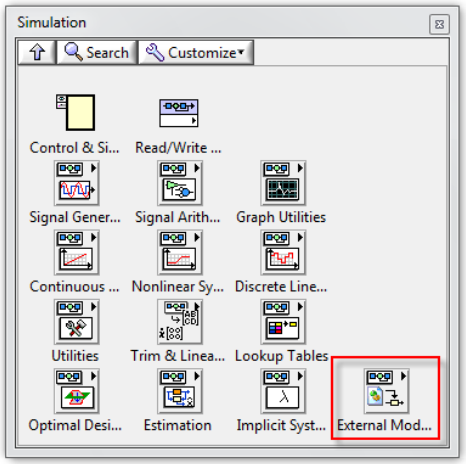
- Double click the HB/SC connector to change the name and wire it to the appropriate node in the circuit.

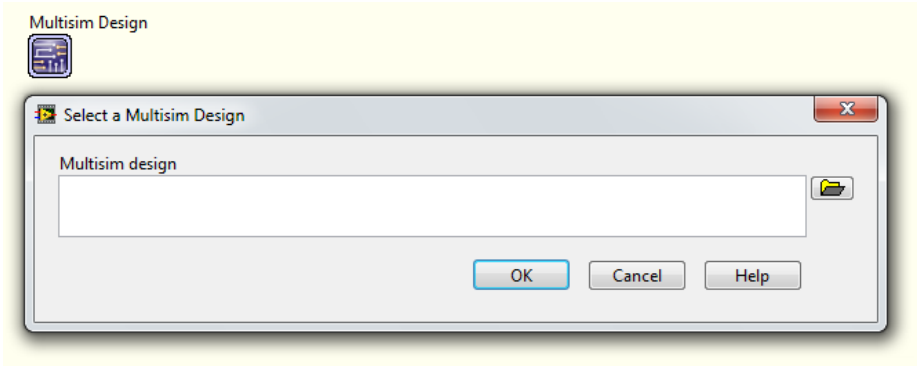



- In LabVIEW, open a new VI and go to the block diagram. Navigate to **Control Design & Simulation>Simulation>Control & Simulation Loop**.

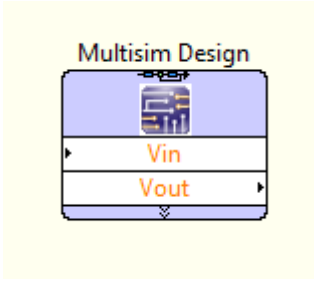


- Navigate to **Control Design & Simulation>Simulation>External Model Interface>Multisim** and a **Multisim Design** node into the **Control Design & Simulation** loop. Then select the name of the MultiSim Design (filename ending in **.ms12**) and click **OK**.




- The **Multisim Design** interface block now appears in your LabVIEW Control & Simulation Loop. You may need to drag down on the bottom of the block to expose named terminals as shown below. Each **HB/SC** connector in your Multisim schematic.



# TRIAC Light Dimmer

**Co-Simulation Concepts**

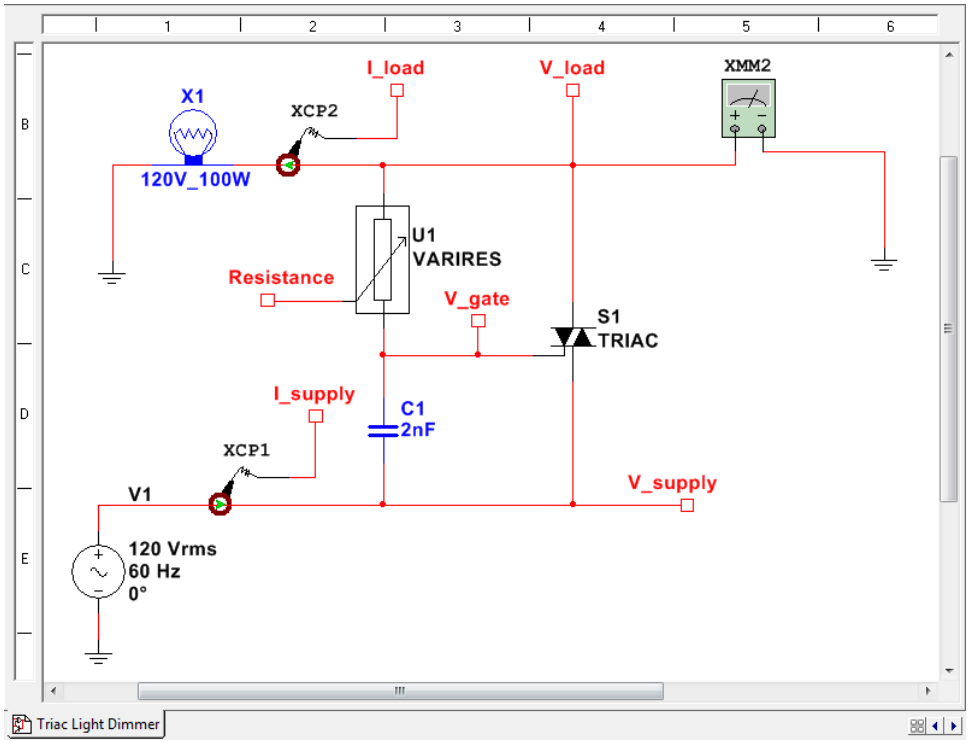
- Using a voltage controlled resistor component in Multisim that can be controlled by LabVIEW
- Using **Current Probes** in Multisim
    - Double-click the probe to configure scaling properties
- Using a fixed time-step solver in LabVIEW
    - Double-click the LabVIEW Control & Simulation Loop to change the simulation settings
    - How does changing the **Step Size (s)** value on the LabVIEW front panel affect the simulation results chart?
- Configuring the Icon Style of the Multisim Design block in LabVIEW

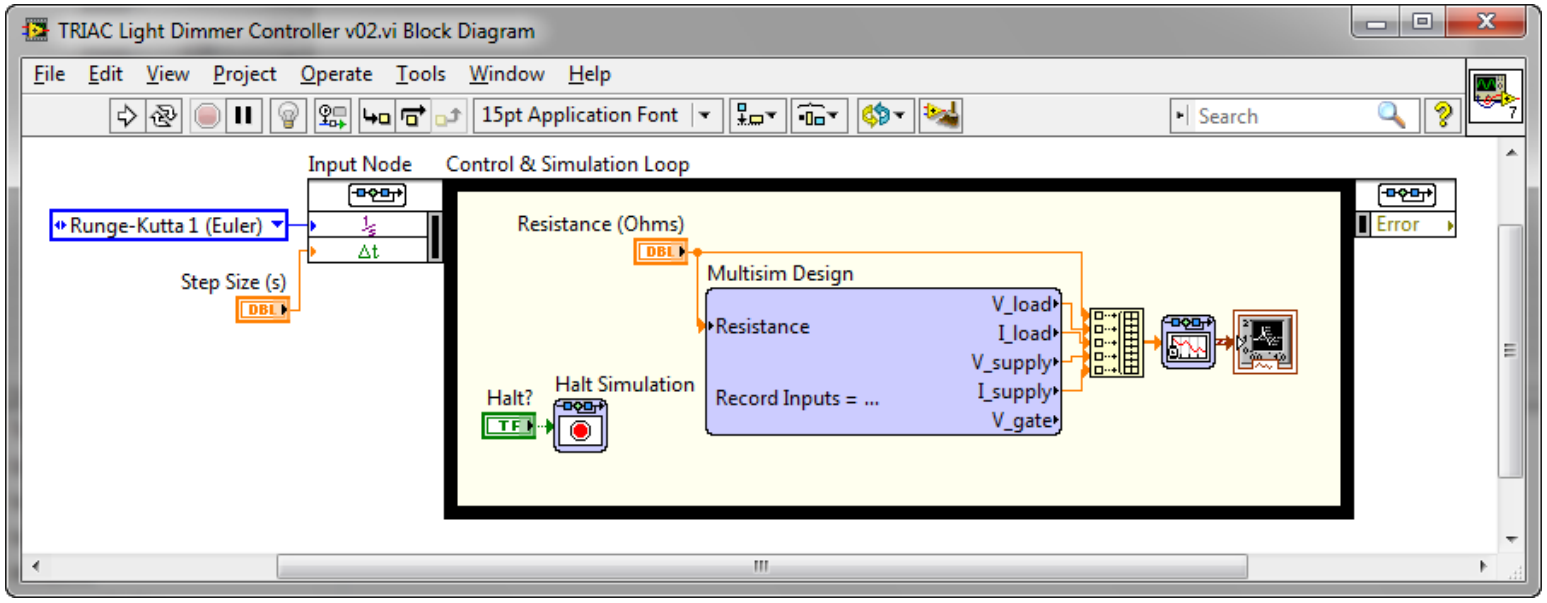
**Power Electronics Concepts**

- Fundamentals of TRIAC light dimmer circuits, a voltage driven switch in which the gate voltage is controlled by a variable resistor (learn more)

**Required Toolkits and Modules**

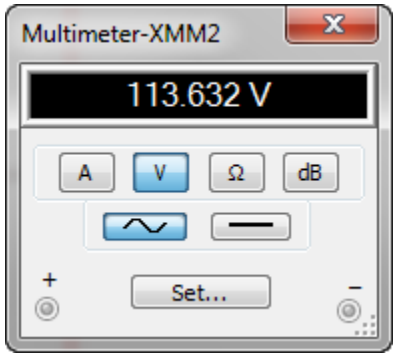
- NI LabVIEW 2011 Full or Professional (ni.com/labview)
- NI LabVIEW Control Design & Simulation Module 2011 (help files)
- NI Circuit Design Suite 12.0 Beta 0 (or higher)(ni.com/multisim)


I_load
V_load
XMM2
X1
XCP2
120V_100W
U1
VARIRES
Resistance
V_gate
S1
TRIAC
I_supply
C1
2nF
XCP1
V1
V_supply
120 Vrms
60 Hz
0°
Triac Light Dimmer

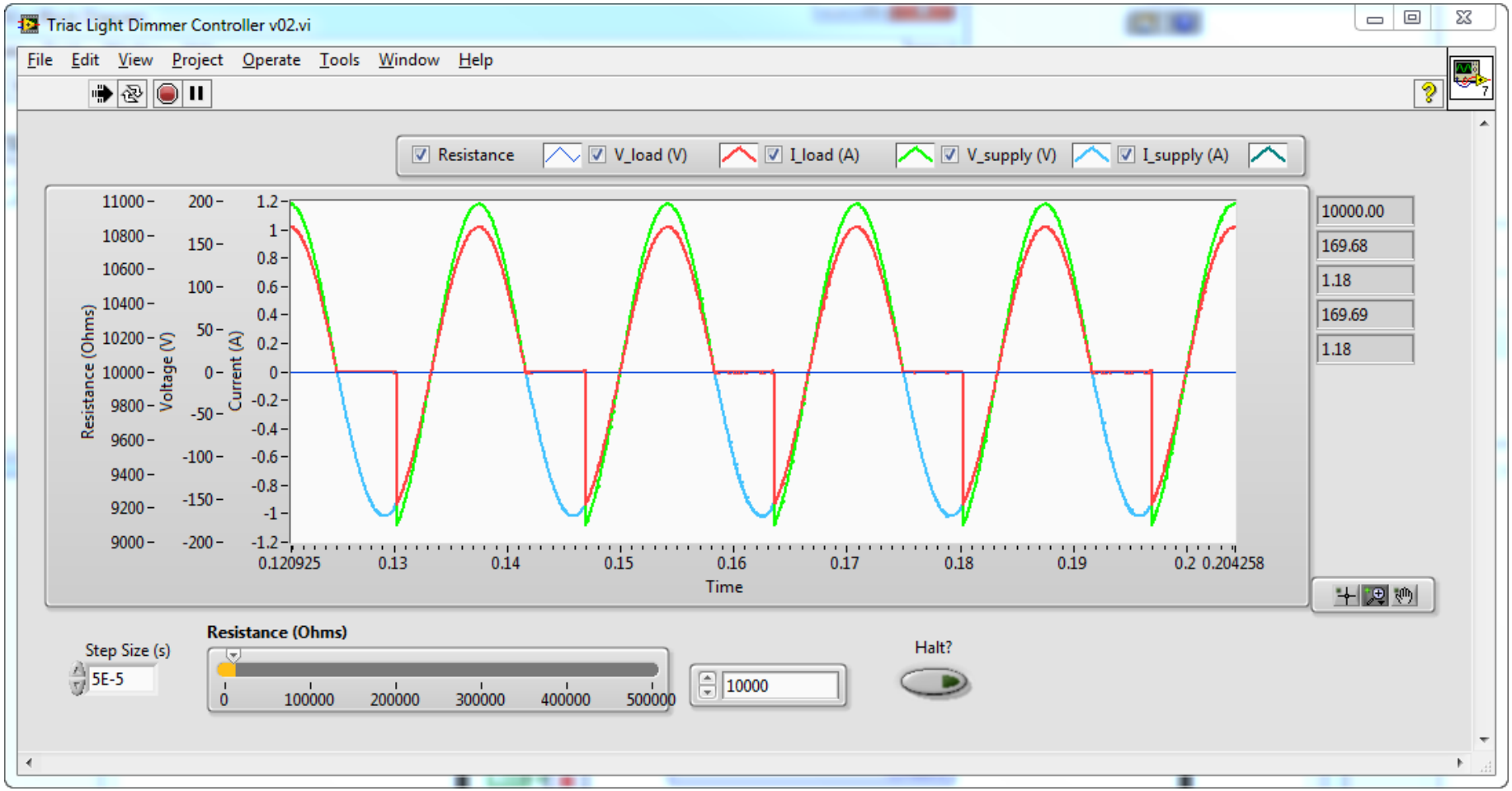

TRIAC Light Dimmer Controller v02.vi Block Diagram
File Edit View Project Operate Tools Window Help
15pt Application Font
Search
Input Node
Control & Simulation Loop
Runge-Kutta 1 (Euler)
Step Size (s)
Resistance (Ohms)
Multisim Design
Resistance
Record Inputs = ...
V_load
I_load
V_supply
I_supply
V_gate
Halt?
Halt Simulation
Error

**Release Notes**

- “Interactive mode simulations” are not supported in this release, therefore the **XMM2 Multimeter** cannot be used during co-simulation with LabVIEW. This feature is not yet committed for future releases since the roadmap is still in definition phase. Please let us know if it’s important for your applications you are able to use Multisim instruments like DMMs, oscilloscopes and spectrum analyzers during co-simulation.
    - You can typically find these instruments on the toolbar to the right of your Multisim schematic.

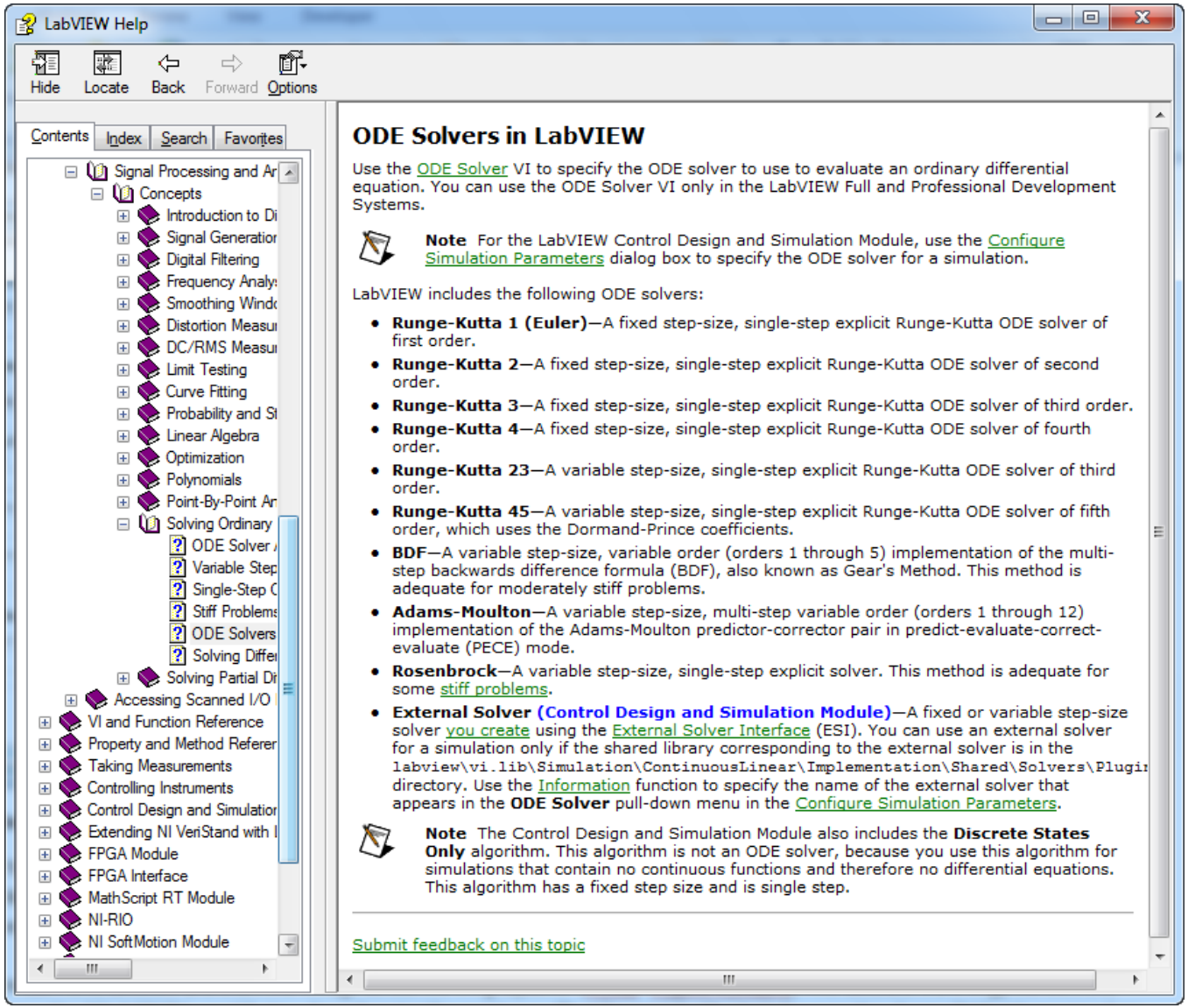


**Simulation Results**

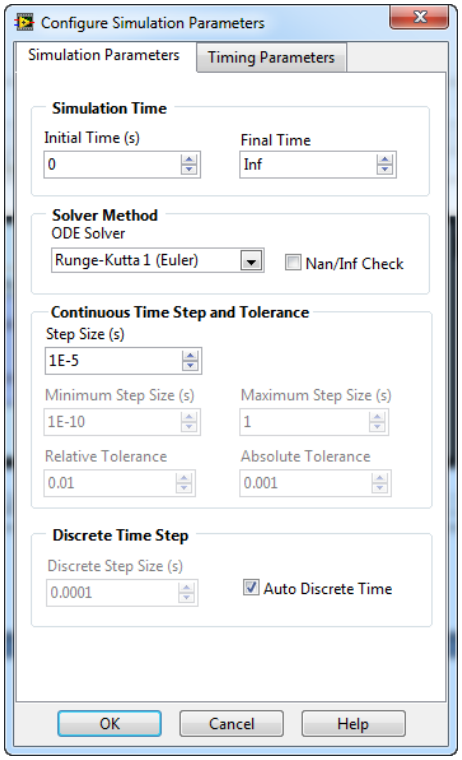


**Suggested Co-Simulation Exercises**

- Try changing the fixed step solver **Step Size (s)**. How does this affect the speed of the simulation? Try changing to a variable time step solver like **Runge-Kutta 45 (variable)**. Can you explain the difference between a fixed timestep and variable timestep solver? For more information, explore the LabVIEW help topic, **ODE Solvers in LabVIEW**.



- Explore the LabVIEW **Control & Simulation Loop** settings. To do this, navigate to the LabVIEW block diagram. Then, double-click the **Input Node** terminal in the upper left corner of the simulation loop to configure the simulation parameters. Notice that the solver is set to **Runge-Kutta 1 (Euler)**.
    - This is a discrete time (fixed time step) rather than continuous time (variable time step) solver. In this case, the fixed time solver in LabVIEW is set to a step size of 1e-5 s (100 kHz).
    - A fixed time step solver always uses the same time interval for each iteration of the simulation and does **not** perform error checking to verify that the simulation results satisfy accuracy constraints.



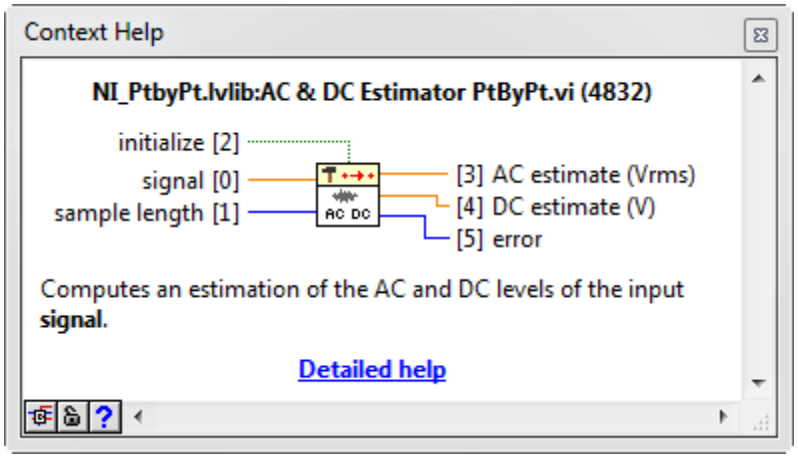

    - Guidelines for choosing a solver:
        - Multisim always uses a variable time step solver. Variable time step solvers automatically adjust their step size to satisfy your accuracy and min/max time step constraints.
            - To access the solver settings Multisim, navigate to **Simulate>Interactive Simulation Settings**. You can find more information about how to change these settings in the **Buck Converter (Closed Loop** PID Control) section.

- Using a variable time step solver is generally recommended for co-simulation. In particular, it is recommended if any of the following describe your LabVIEW application:
    - Contains continuous time simulation elements, such as transfer function or state-space blocks.
    - You have multiple LabVIEW FPGA subVIs that needs to be simulated at different discrete time rates. (To do this, right-click the subVI and go to subVI node setup. This is explained in more depth in the **Buck Converter (Closed Loop** PID Control) section.)
- Using a fixed step solver is sometimes okay for co-simulation. Fixed time step solvers typically execute faster than variable time step solvers. However, there is no (automatic) guarantee of accuracy. Using a discrete time solver may be okay if all of the following describe your LabVIEW application:
    - All of your LabVIEW FPGA code executes at a single discrete time rate (or integer multiples a single rate).
    - Contains continuous time simulation elements, such as transfer function or state-space blocks, but you have verified that the discrete step size is sufficiently small to produce accurate results.

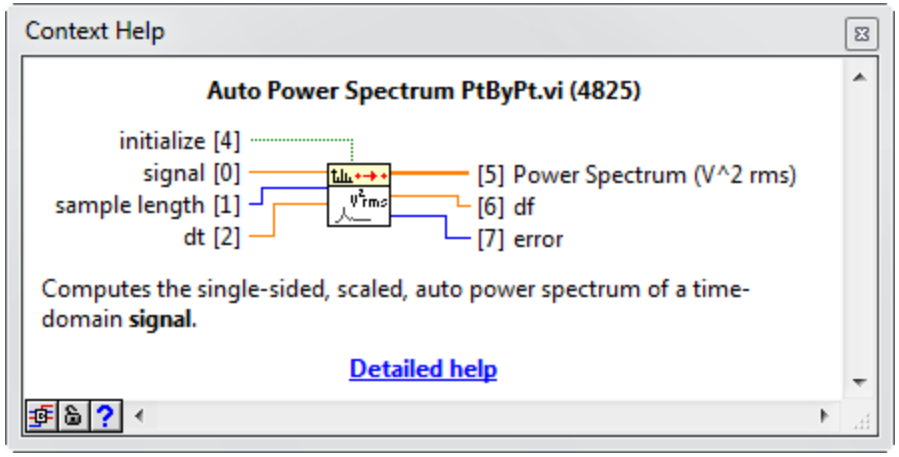
- Notes:
    - If you are using a discrete time solver other than **Runge-Kutta 1 (Euler),** you must set your LabVIEW FPGA subVI to discrete time mode if it contains state information (such as a feedback node or any type of memory element). That's because these solvers execute the LabVIEW simulation diagram more than once per iteration.
    - Care must be taken to ensure that the step size is sufficiently small to obtain accurate results if you are using a discrete time solver and your LabVIEW application contains dynamic, continuous time simulation blocks such as transfer functions, state-space models or differential equations with integrators.
        - Typically this is done by first executing the simulation in variable time step mode to obtain a benchmark. Then the simulation is executed using a fixed time step solver, and the results are compared to the benchmark simulation. If the fixed step size is too large, the discrete time results will not match the continuous time results.
            - Note: You can use the **Collector** function on the LabVIEW Simulation palette to save simulation results for comparison.
    - When using a variable step solver, you can use the **Zero Crossing** function in LabVIEW to give the solver a higher degree of accuracy in the timing of events such as the switching ON/OFF of a transistor gate drive signal. This is explained in the **Brushless DC Motor with Hall Sensors** section.

**Suggested Power Electronics Exercises**

- In LabVIEW, calculate and chart the voltage across the firing capacitor (V_cap=V_gate-V_supply). Also calculate the energy stored in the capacitor (W=0.5*C*V_cap^2). How does changing the resistance affect the speed with which the capacitor charges? What happens when the voltage at the gate of the TRIAC exceeds its trigger gate voltage (2.5 V)?
- Add LabVIEW code to calculate the AC RMS value of the load power (P_load=V_load*I_load) and supply power (P_supply=V_supply*I_supply). Next, calculate the energy efficiency of the dimmer circuit (E=P_load/P_supply).
    - Hint:
        - Use the **AC & DC Estimator PtByPt.vi** function to calculate the power spectrum. To find it, right-click on an empty area of the LabVIEW block diagram and navigate to **Signal Processing>Point by Point>Spectral Analysis PtByPt**.




    - Add LabVIEW code to calculate and graph the auto power spectrum of the load power. Can you reduce the magnitude of the harmonic energy in the frequency spectrum?
    - Hints:
        - Use the **Auto Power Spectrum PtByPt.vi** function to calculate the power spectrum.




- Try adding an inductor choke and interference capacitor to the circuit.

## Simple Chaotic Circuit

**Co-Simulation Concepts**

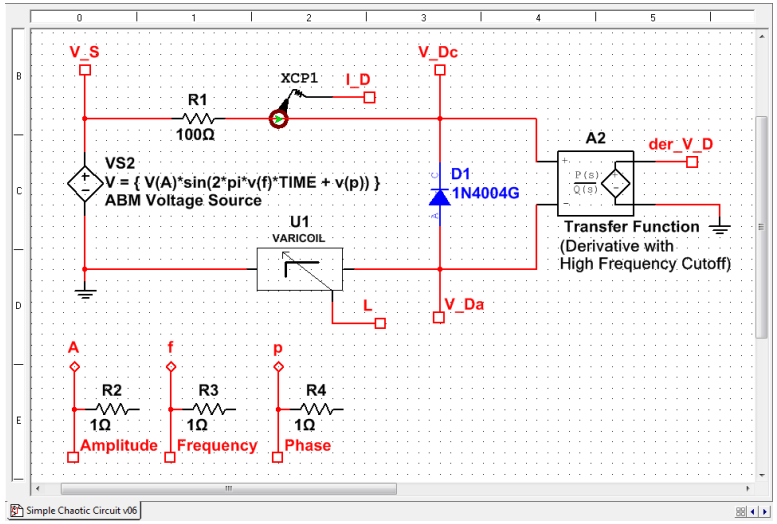
- Using a parameterized function generator voltage source in Multisim that is controlled by LabVIEW.
  - On page connectors are used to set the values of the function generator variables, while HB/SC connectors are used to interface to LabVIEW.
- Using a voltage controlled inductor component in Multisim that is controlled by LabVIEW.
- Using a transfer function component in Multisim to implement a derivative with high frequency cutoff frequency. Setting an appropriate value for the cutoff filter helps the Multisim solver converge.
  - The High Frequency Cutoff frequency, *f*, in Hz can be converted to rad/s for use in the transfer function as follows:
    - $\omega$ [rad/s]= $1/(2\pi f)$
  - Then the transfer function is:
    - $G = \frac{s}{\omega s+1}$

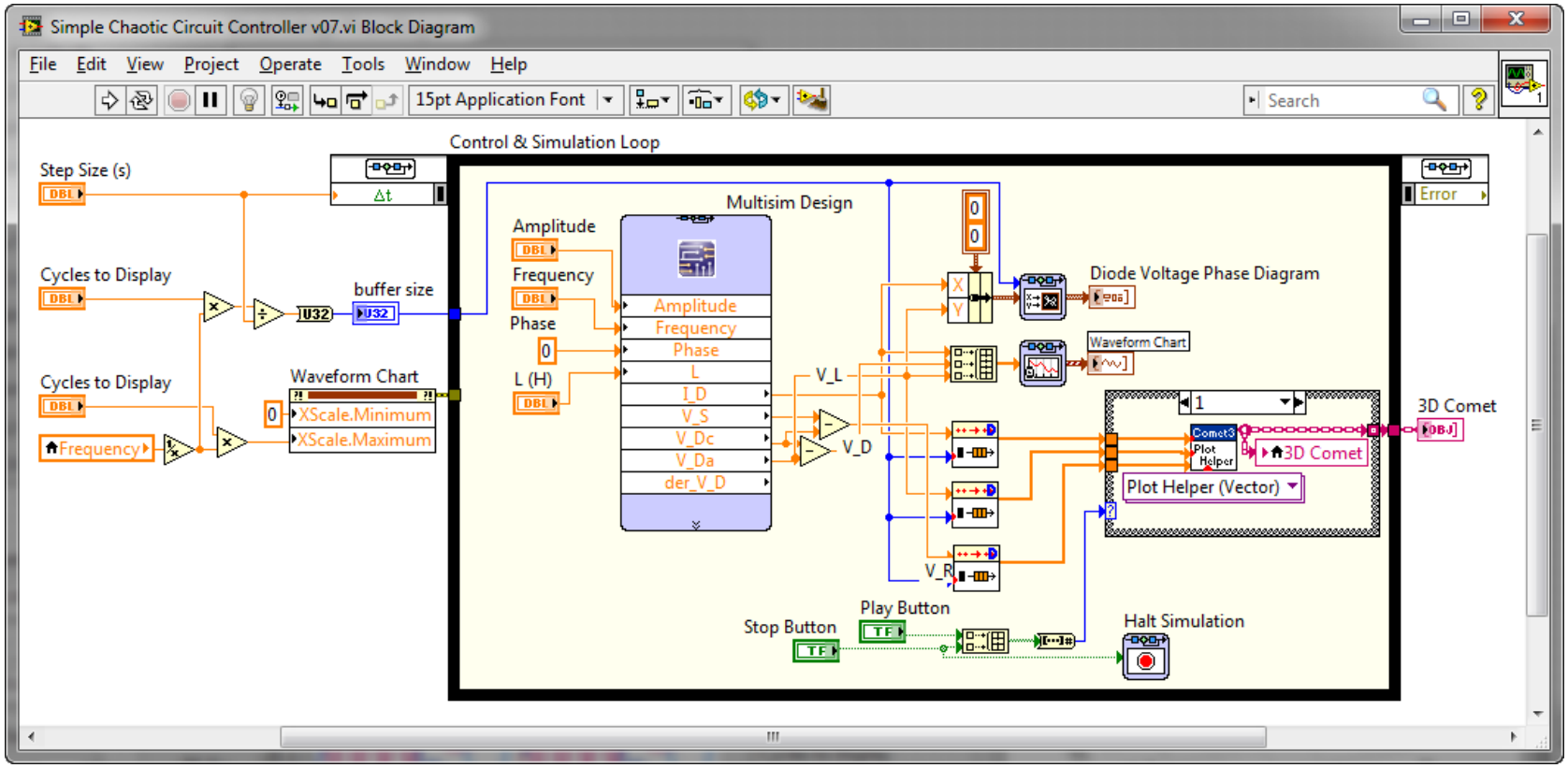
**Power Electronics Concepts**

- Fundamentals of chaos in switching power circuits, which in this case occurs when there is non-linear interaction between inductor energy storage components and passive switching devices such as diodes (learn more).
  - Chaos is aperiodic behavior in deterministic nonlinear dynamical systems that is highly sensitive to initial conditions.
  - Let this example serve as an important caution for designers of switch-mode power supplies; it is important to appreciate that is not difficult to produce chaotic circuit operation when non-linear transistors such as p-n junction devices and energy storage elements such as inductors and capacitors are involved.
    - In this case, the mechanism resulting in chaos has to do with the non-ideal nature of the p-n junction of the diode. The non-ideal diode continues to conduct for a short period of time after the forward current through the diode reaches zero. This phenomena is known as the reverse recovery time. Furthermore, the diode exhibits a non-linear capacitance with respect to voltage.
    - This circuit follows a period doubling path to chaos in which the system switches to a new behavior with twice the period of the original system during bifurcation (learn more).

**Required Toolkits and Modules**

- NI LabVIEW 2011 Full or Professional (ni.com/labview)
- NI LabVIEW Control Design & Simulation Module 2011 (help files)
- NI Circuit Design Suite 12.0 Beta 0 (or higher) (ni.com/multisim)


V_S
XCP1
I_D
V_Dc
R1
100Ω
A2
der_V_D
VS2
V = { V(A)*sin(2*pi*v(f)*TIME + v(p)) }
ABM Voltage Source
D1
1N4004G
U1
VARICOIL
Transfer Function
(Derivative with
High Frequency Cutoff)
L
V_Da
A
f
p
R2
1Ω
Amplitude
R3
1Ω
Frequency
R4
1Ω
Phase
Simple Chaotic Circuit v06

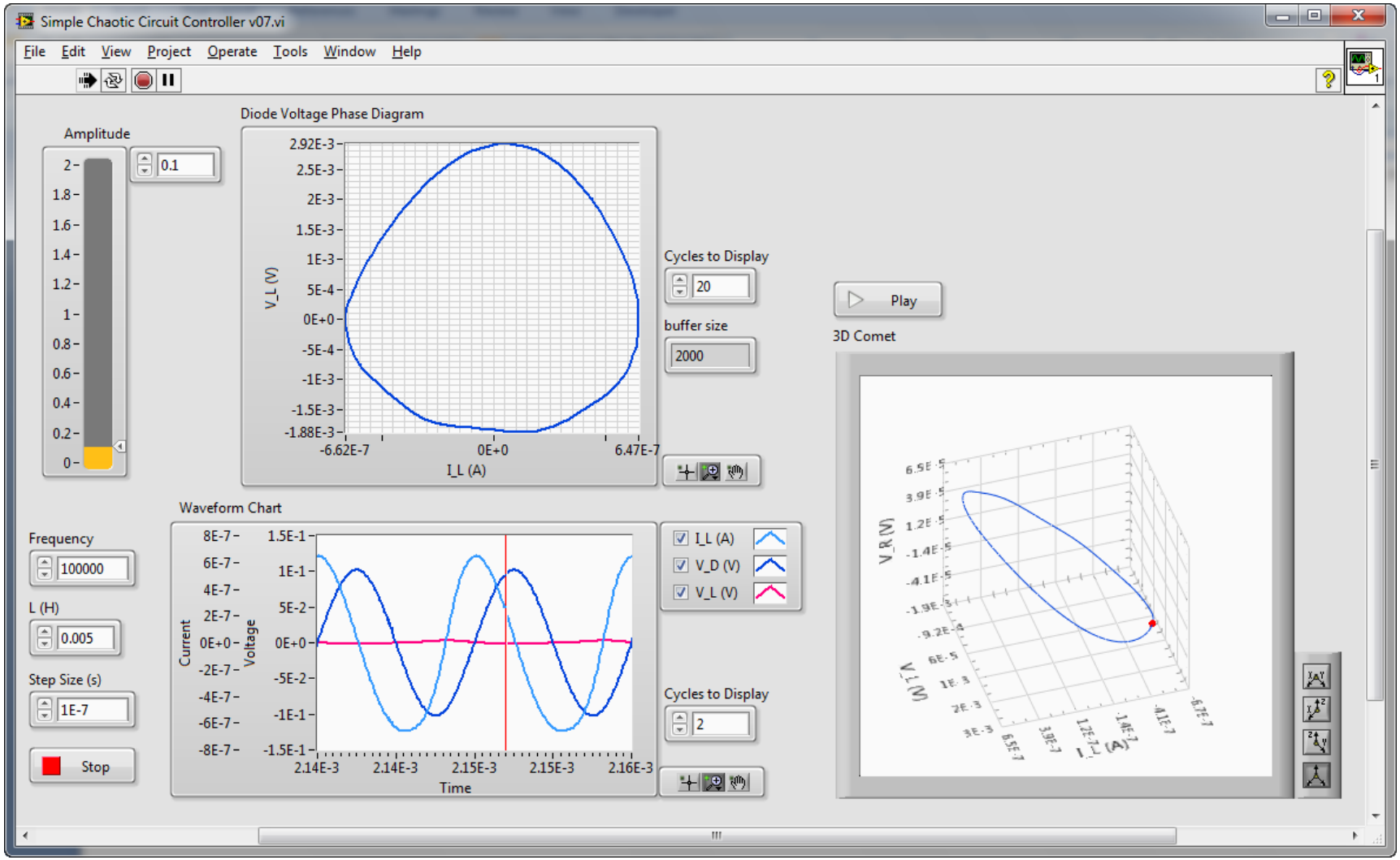

Simple Chaotic Circuit Controller v07.vi Block Diagram
File Edit View Project Operate Tools Window Help
15pt Application Font
Search
Control & Simulation Loop
Step Size (s)
Δt
Error
Cycles to Display
buffer size
U32
Cycles to Display
Waveform Chart
XScale.Minimum
XScale.Maximum
Frequency
Multisim Design
Amplitude
Frequency
Phase
L (H)
Amplitude
Frequency
Phase
L
I_D
V_S
V_Dc
V_Da
der_V_D
V_L
V_D
V_R
Diode Voltage Phase Diagram
Waveform Chart
3D Comet
Plot Helper
3D Comet
Plot Helper (Vector)
Play Button
Stop Button
Halt Simulation

**Known Issues**

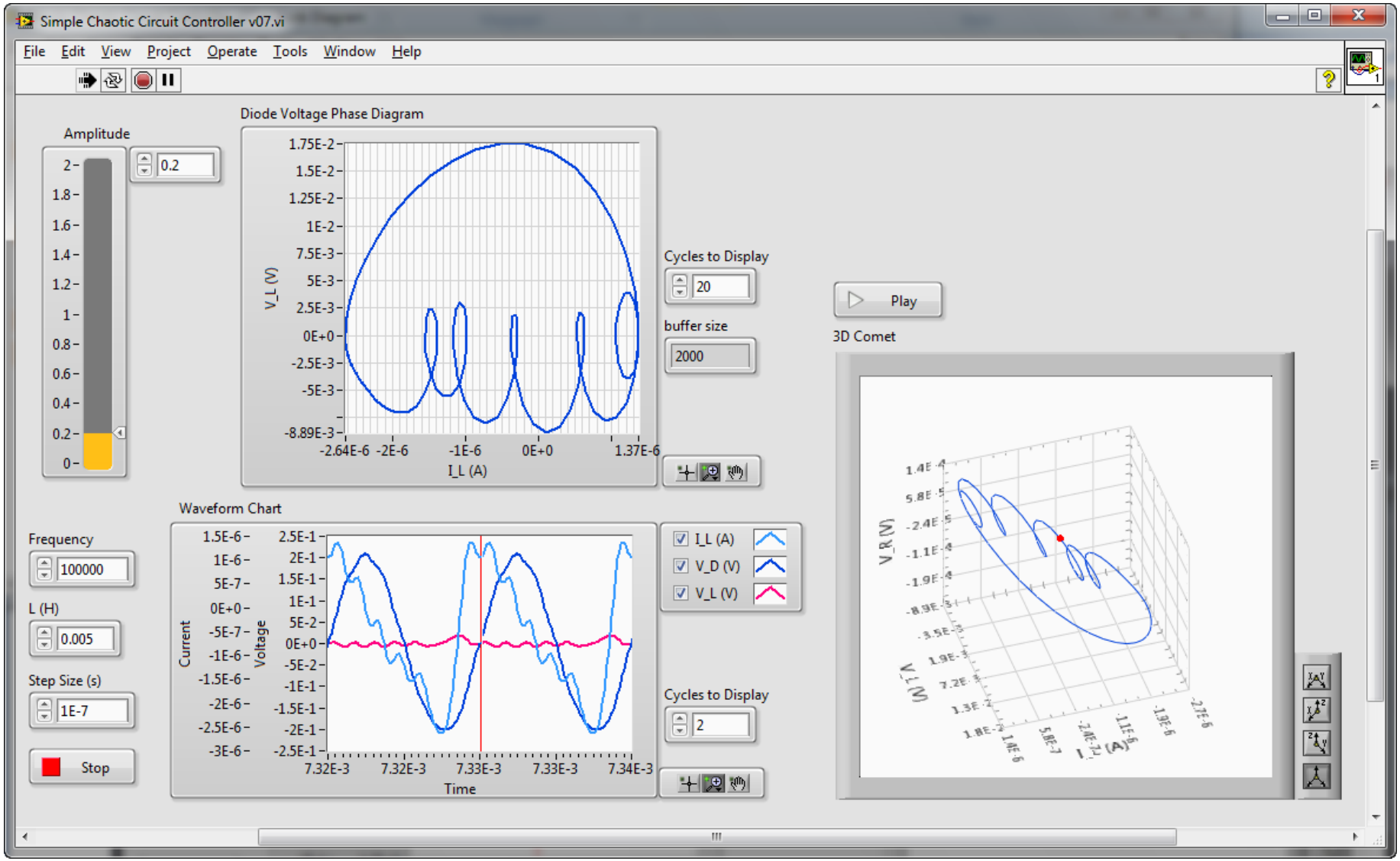
- Currently, resistors must be added between the on-page connectors that interface to the **ABM Voltage Source** parameters and the HB/SC connectors that interface to LabVIEW. In the beta release, a netlist error would otherwise occur. This issue will be resolved.

**Simulation Results**

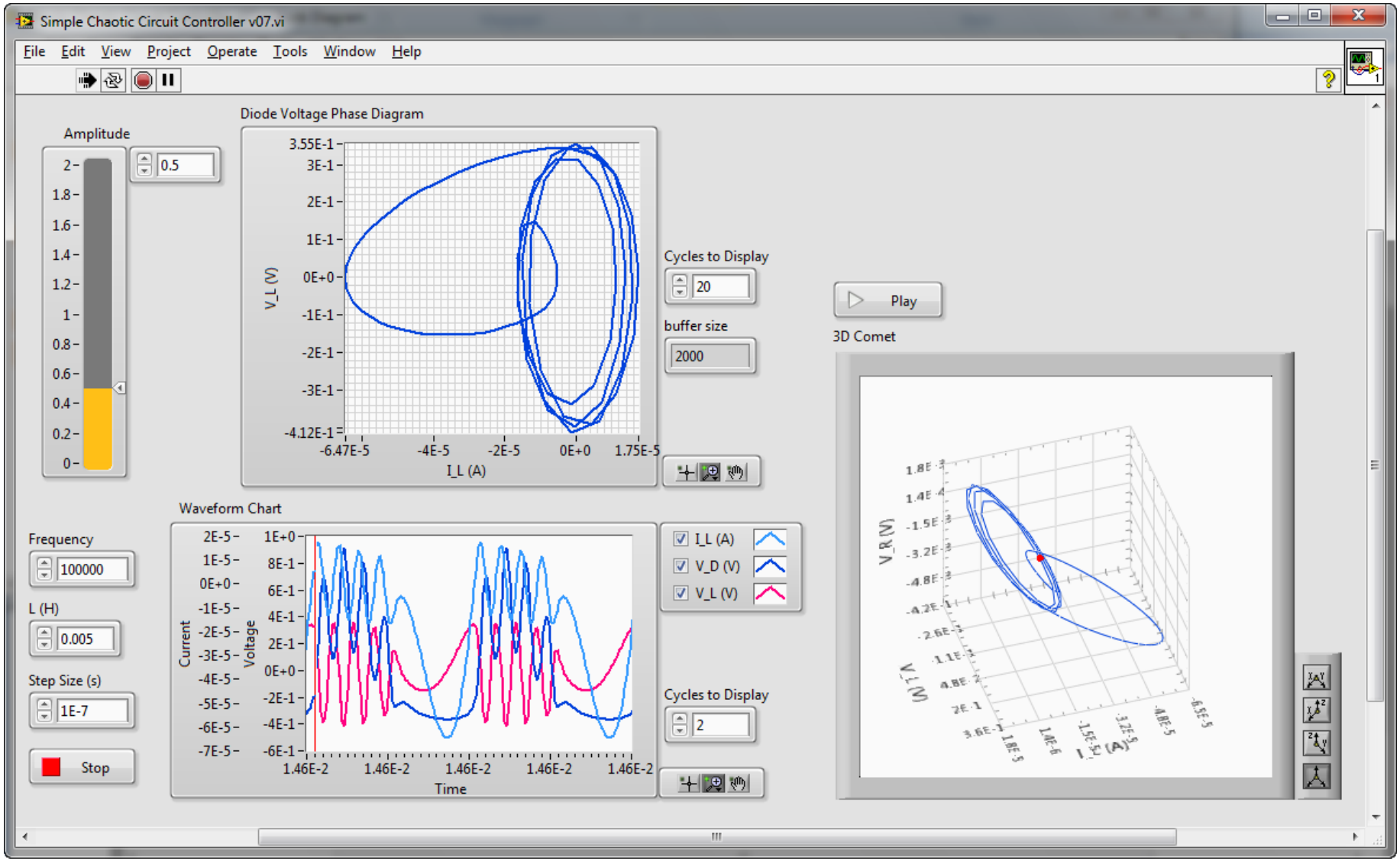
- At a sinusoidal voltage amplitude of 0.1 V, the circuit is periodic but non-linear.



- At a 0.2 V sinusoidal voltage amplitude, a Period-5 window appears.

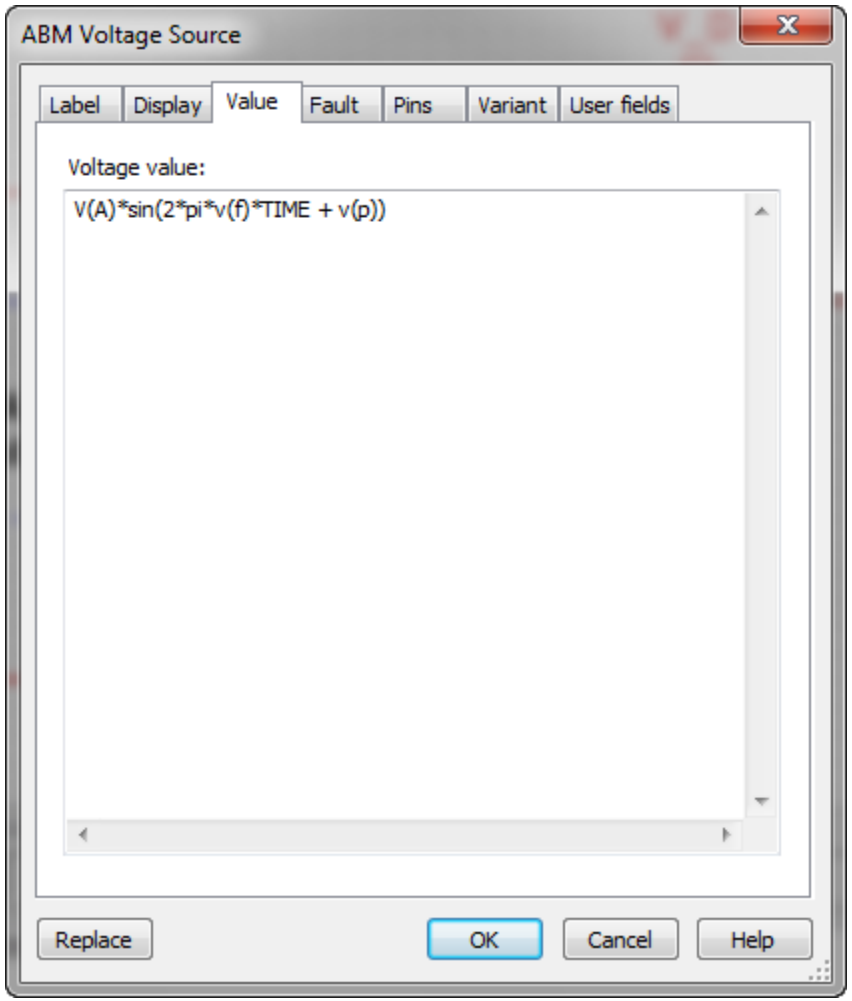


- A 0.5 V sinusoidal voltage amplitude produces an attractor.

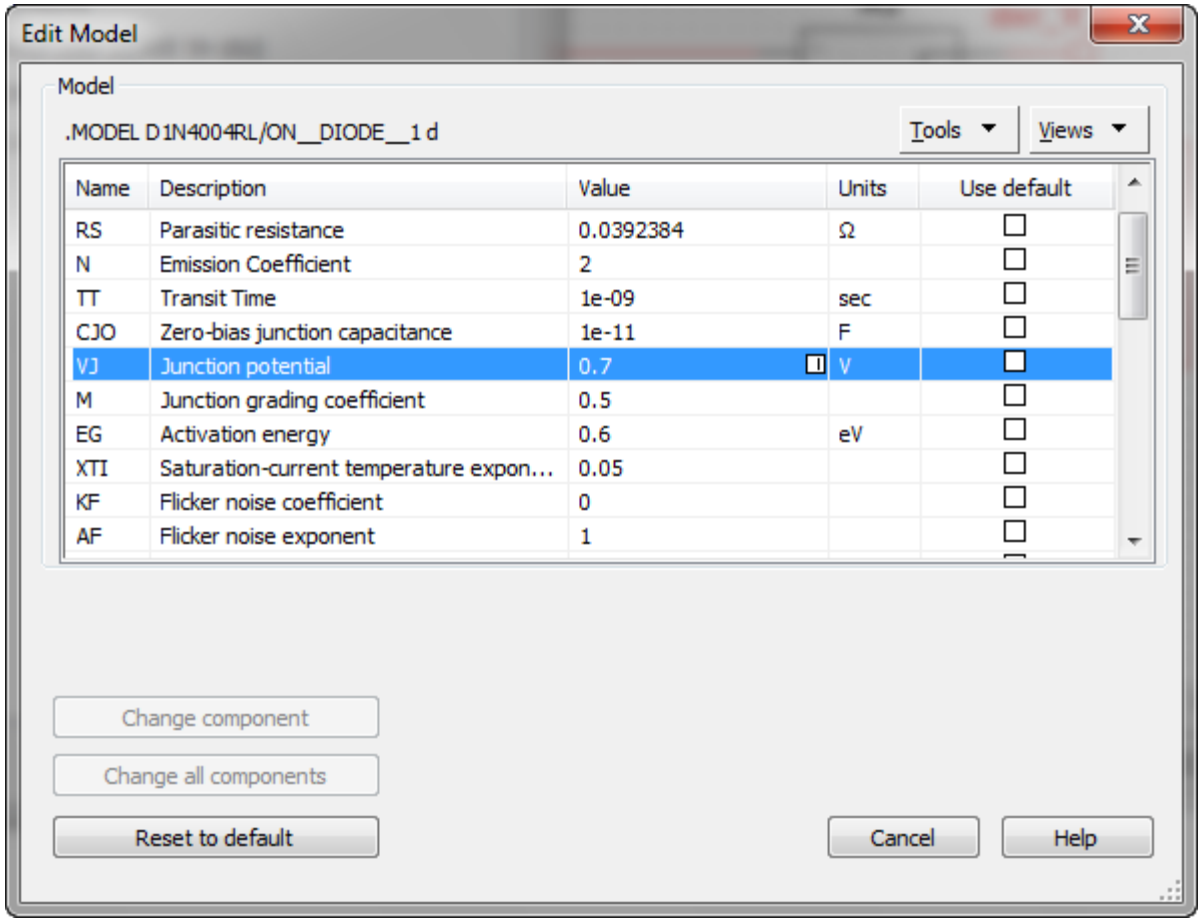


**Suggested Exercise**

- Try incrementing the sinusoidal voltage source Amplitude from 0.3 V to 1 V. How do the patterns in the phase diagram change depending on Amplitude? How does changing the frequency of the sinusoidal voltage source affect the results? How does changing the inductance affect the results?
- In Multisim, double-click on the **ABM Voltage Source** component and explore its configuration. This voltage source can be used as an arbitrary waveform generator based on a programmable mathematical function. The variables in the mathematical function can be controlled by LabVIEW.



- In Multisim, double-click on the **1N4004G** diode component and select **Edit Model**.

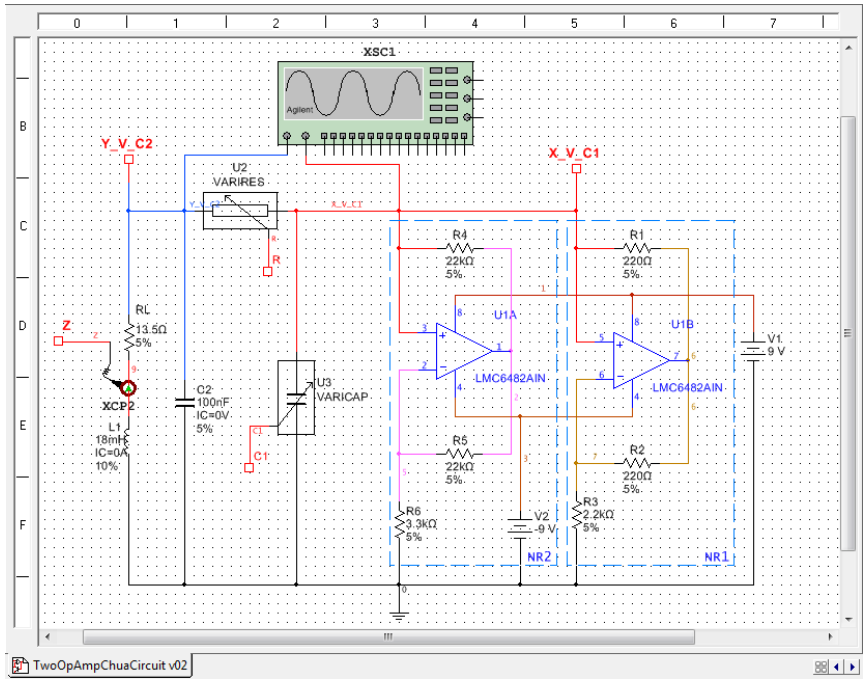


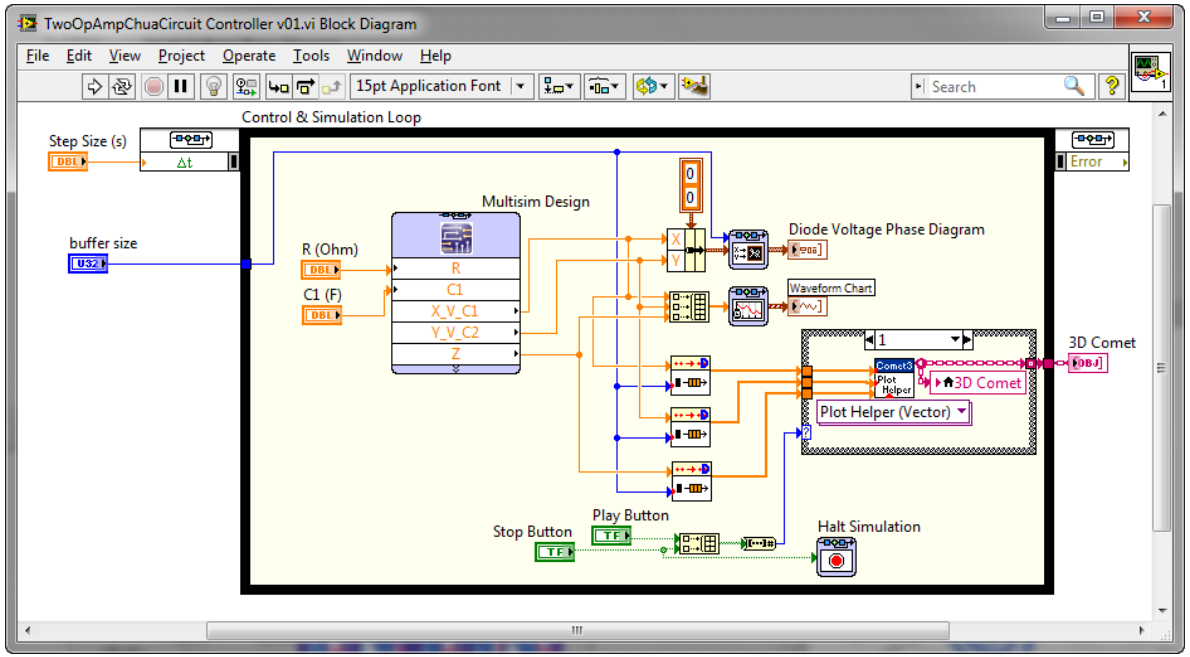
**Bonus Exercise**

- Explore the circuit designed by Dr. Leon Chua in 1983, known as Chua's circuit. It is a simple nonlinear circuit capable of producing strange attractors. In this case, LMC6482 op-amps used to create a non-linear resistance (learn more).
    - Try fixing the resistance, R, at 1.8 kOhms and varying the capacitance, C1. Can you produce Period 1, 2, 3 and 4 windows? Can you produce strange attractors?

**Community Challenge**

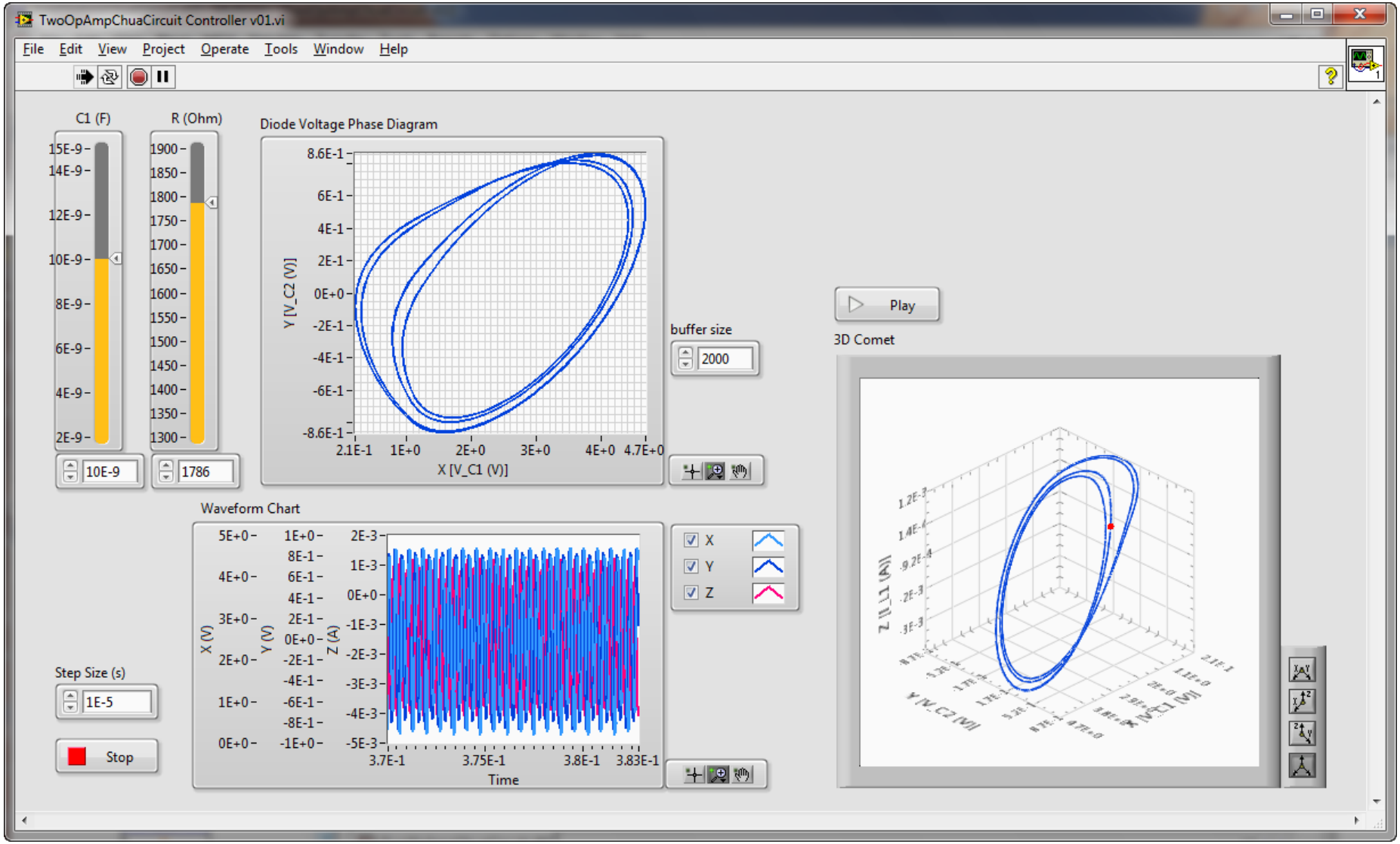
- The author would like to encourage you to enhance these samples by adding a variety of analytical techniques built in LabVIEW for analyzing chaotic circuits such as phase portraits, Poincare diagrams and bifurcation diagrams (learn more).


XSC1
Y_V_C2
X_V_C1
U2
VARIRES
R4
22kΩ
5%
R1
220Ω
5%
U1A
U1B
RL
13.5Ω
5%
V1
9 V
LMC6482AIN
XCP2
C2
100nF
IC=0V
5%
U3
VARICAP
L1
18mH
IC=0A
10%
R5
22kΩ
5%
R2
220Ω
5%
R6
3.3kΩ
5%
V2
-9 V
R3
2.2kΩ
5%
NR2
NR1
TwoOpAmpChuaCircuit v02

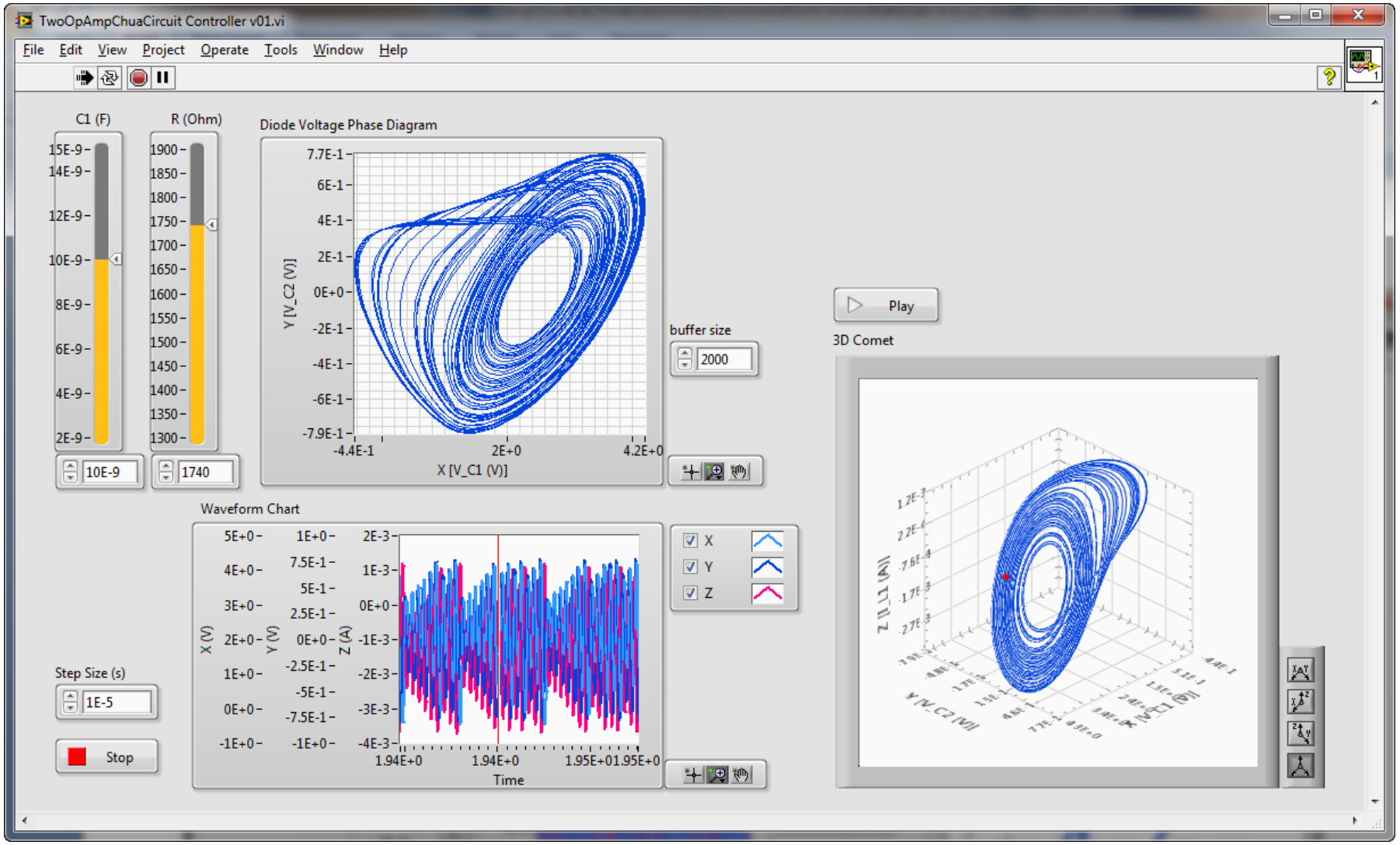

TwoOpAmpChuaCircuit Controller v01.vi Block Diagram
File Edit View Project Operate Tools Window Help
15pt Application Font
Search
Control & Simulation Loop
Step Size (s)
Δt
Error
buffer size
R (Ohm)
C1 (F)
Multisim Design
R
C1
X_V_C1
Y_V_C2
Z
Diode Voltage Phase Diagram
Waveform Chart
3D Comet
Plot Helper (Vector)
Stop Button
Play Button
Halt Simulation

**Simulation Results**

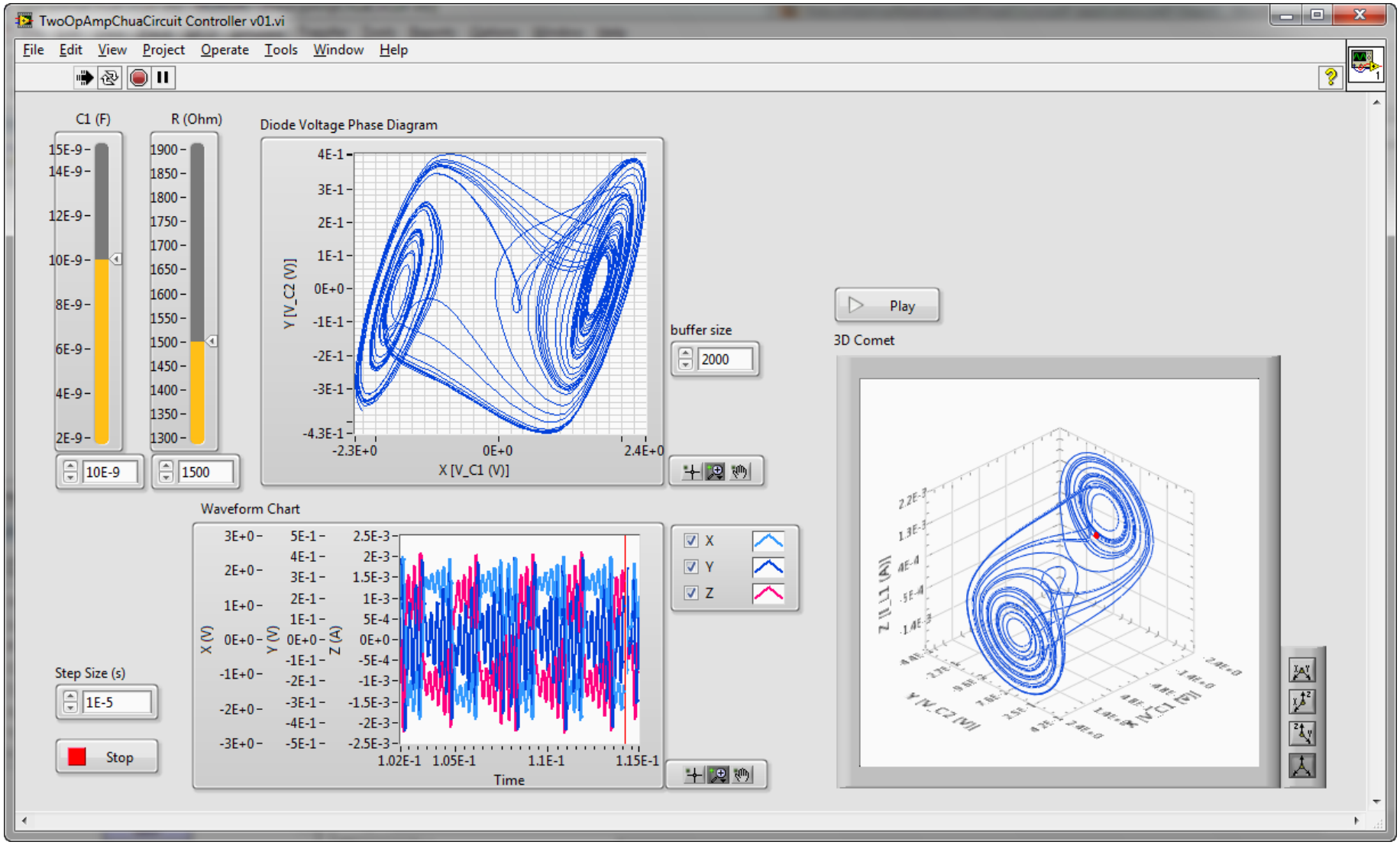
- Period-4 window with C1 = 10 nF, R = 1.786 kOhm.



- Rössler-type attractor with C1 = 10 nF, R = 1.74 kOhm.



- Double scroll attractor with C1 = 10 nF, R = 1.5 kOhm.



# Flyback Converter

**Co-Simulation Concepts**

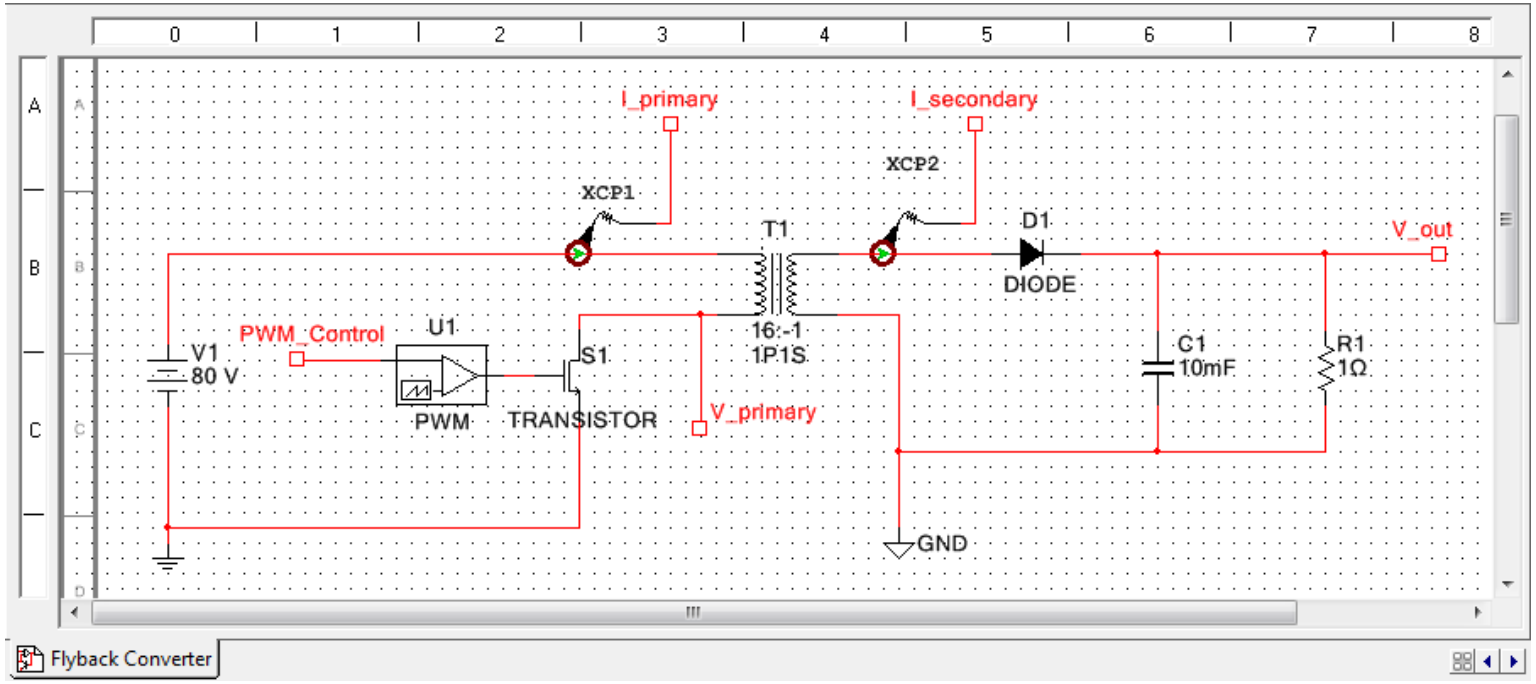
- Configuring the parameters of Multisim circuit components
  - Double-click the 1P1S transformer component to configure the transformer
- Using **Current Probes** in Multisim
  - Double-click the probe to configure scaling properties
- Using a voltage controlled pulse width modulation (PWM) generation block in Multisim

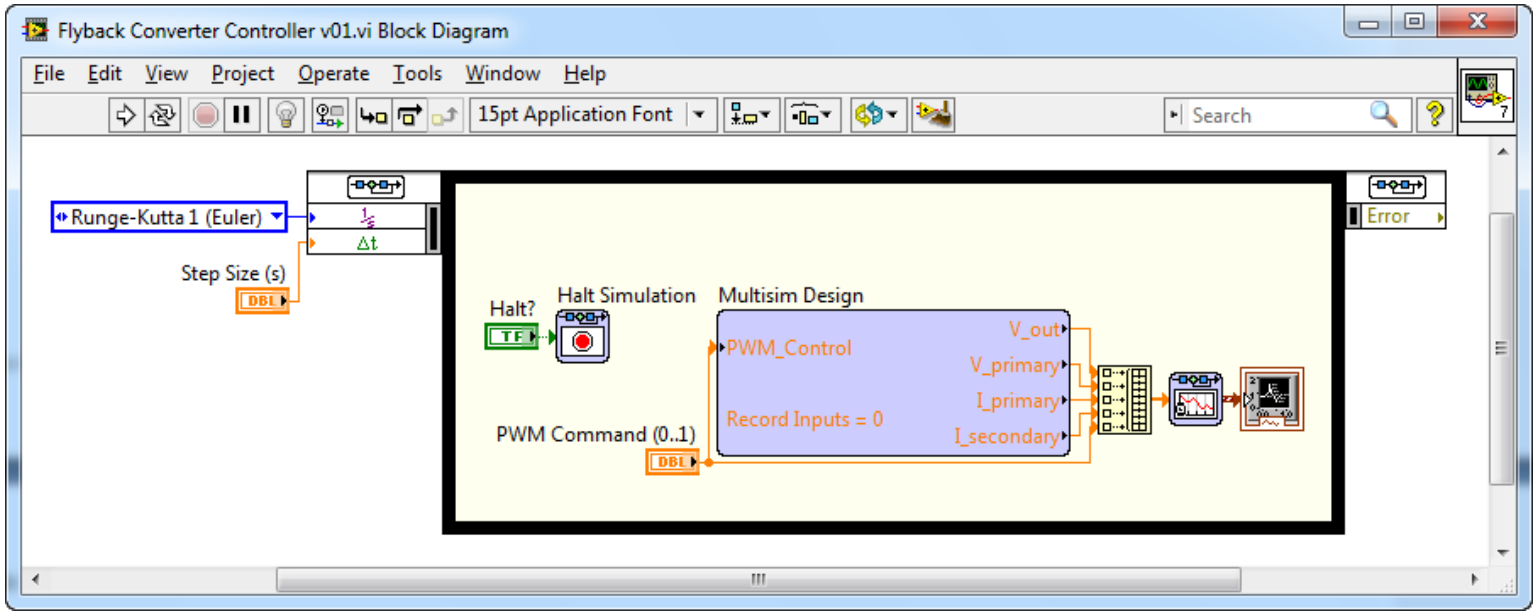
**Power Electronics Concepts**

- Fundamentals of flyback converter switched-mode power supply circuits, a buck-boost converter with galvanic isolation between the input and output (learn more)
- How energy is stored in the tranformer and then discharged into the load
  - Can you explain the relationship between the primary and secondary current signals?
- How the transformer ratio effects the input to output voltage ratio
  - Can you explain the relationship between the transformer ratio, PWM duty cycle command and the output voltage magnitude?

**Required Toolkits and Modules**

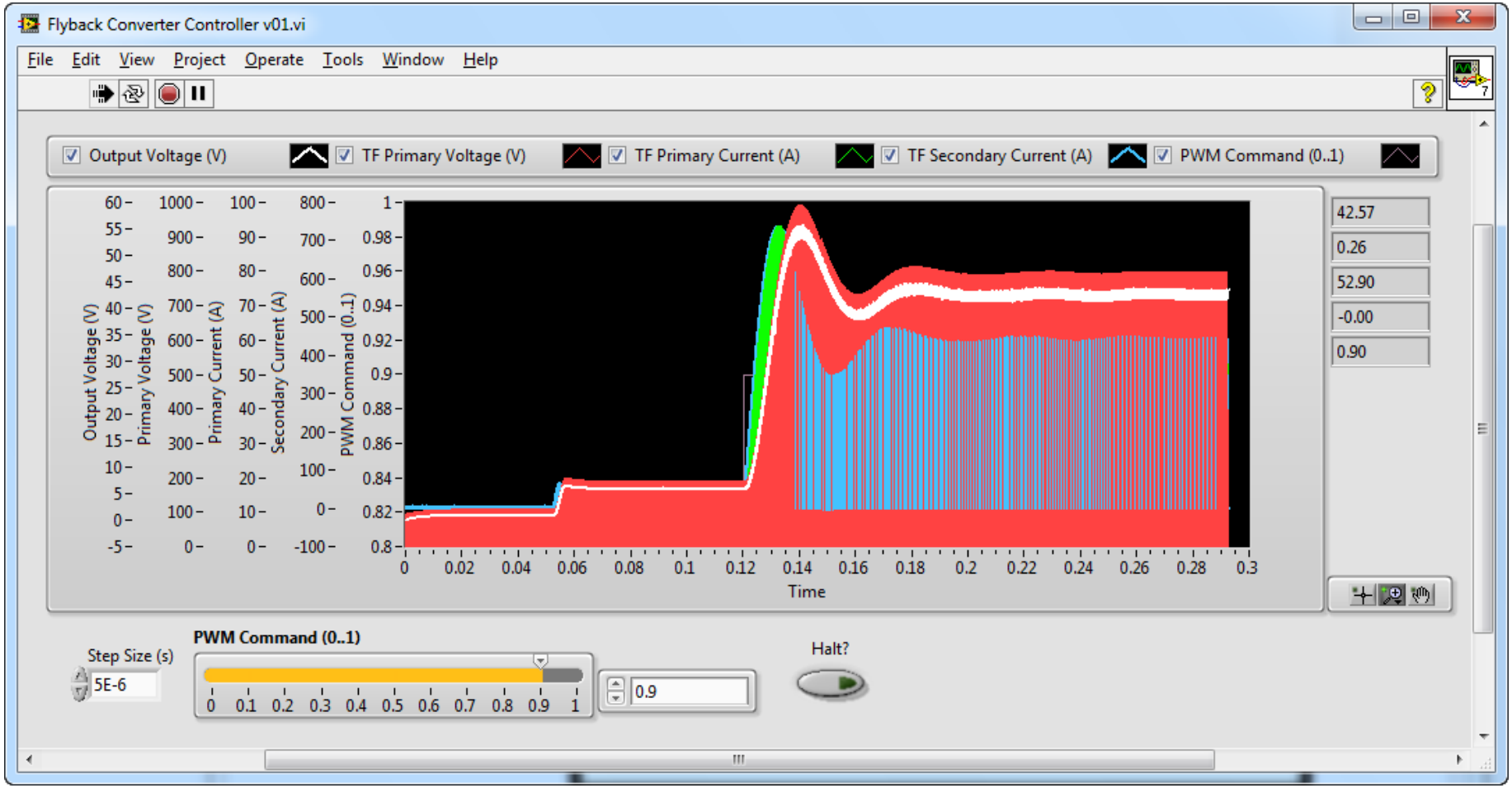
- NI LabVIEW 2011 Full or Professional (ni.com/labview)
- NI LabVIEW Control Design & Simulation Module 2011 (help files)
- NI Circuit Design Suite 12.0 Beta 0 (or higher)(ni.com/multisim)


I_primary
I_secondary
XCP1
XCP2
T1
D1
V_out
DIODE
PWM_Control
U1
V1
80 V
S1
16:-1
1P1S
C1
10mF
R1
1Ω
PWM
TRANSISTOR
V_primary
GND
Flyback Converter


Flyback Converter Controller v01.vi Block Diagram
File Edit View Project Operate Tools Window Help
15pt Application Font
Search
Runge-Kutta 1 (Euler)
Step Size (s)
Error
Halt?
Halt Simulation
Multisim Design
PWM_Control
Record Inputs = 0
V_out
V_primary
I_primary
I_secondary
PWM Command (0..1)

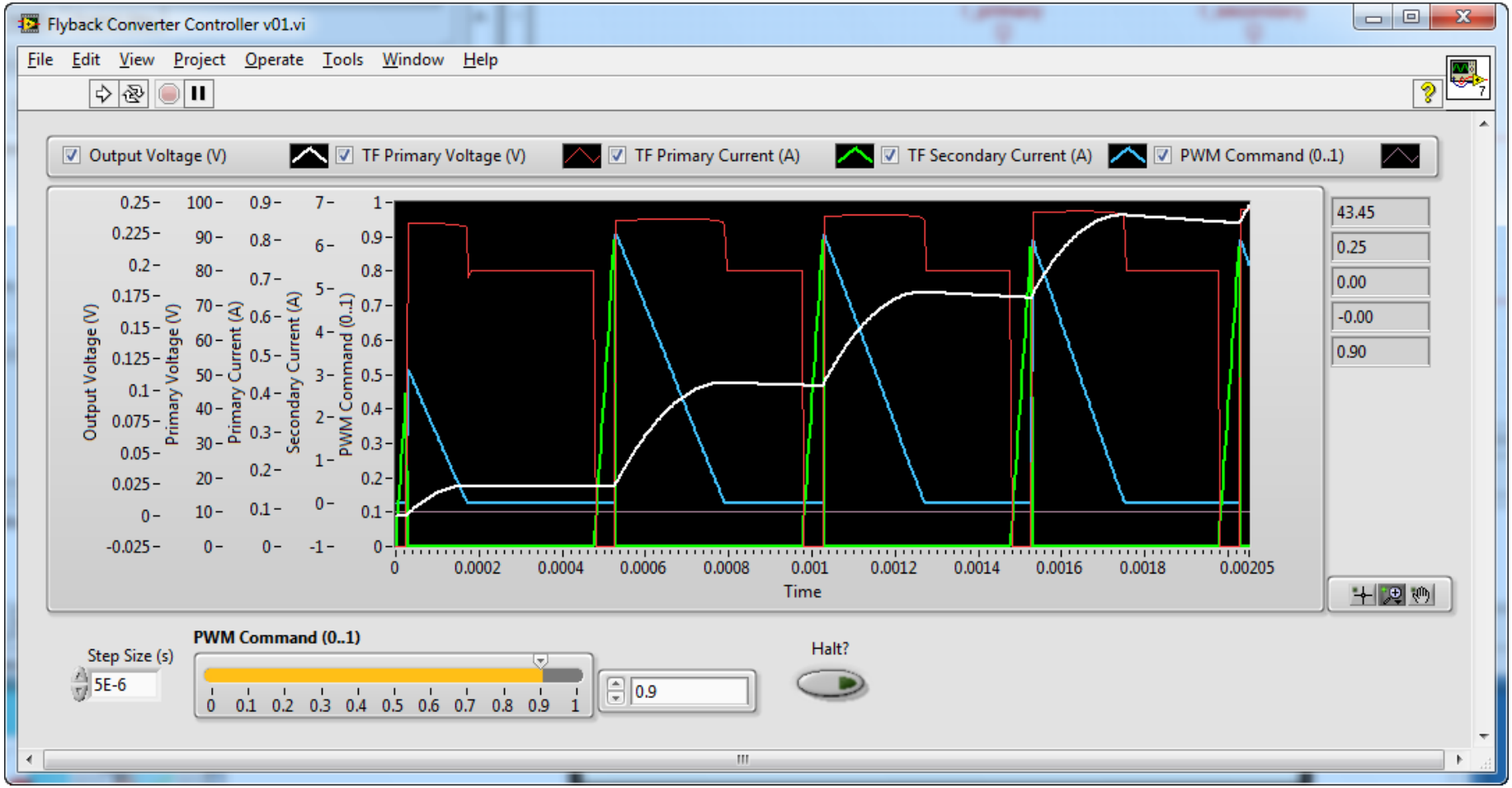
**Simulation Results**

- The PWM duty cycle command is changed from 0.1 to 0.5 to 0.9:



Initial transient response:

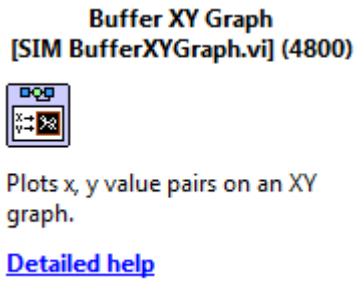
- The initial duty cycle is set to 10 percent. The PWM frequency is 2 kHz.
- Note the initial charging of the primary inductor, and energy transfer relationship between the primary and secondary transformer windings. In this case, the switch is open 90 percent of the time. What happens when it the switch closes, connecting the lower terminal of the primary inductor to ground?
    - As you can see, the pulse occurs every 0.0005 seconds, causing current to charge the primary transformer winding (green trace) and then transfer to the secondary winding (blue trace) when the switch is opened.



Steady state response at 10 percent duty cycle:

- What's the peak-to-peak amplitude of the Output Voltage (V) ripple? How would you modify the circuit design to reduce this ripple?
    - Can you add a LabVIEW subVI to measure the RMS ripple on the output voltage?
        - Hint: Think AC amplitude.
- What's the ratio between the peak-to-peak amplitudes of the primary and secondary currents?
    - How would changing the transformer ratio affect the relative amplitudes of the primary and secondary transformer currents?

- What's the relationship between the **PWM Command (0..1)** and the **Output Voltage (V)**?
    - Can you add a buffered XY graph to plot the relationship?
        - Hints:
            - Think DC amplitude.
            - On the Control Design & Simulation palette you will find a subVI named **Buffer XY Graph**:



# Boost Converter

**Co-Simulation Concepts**

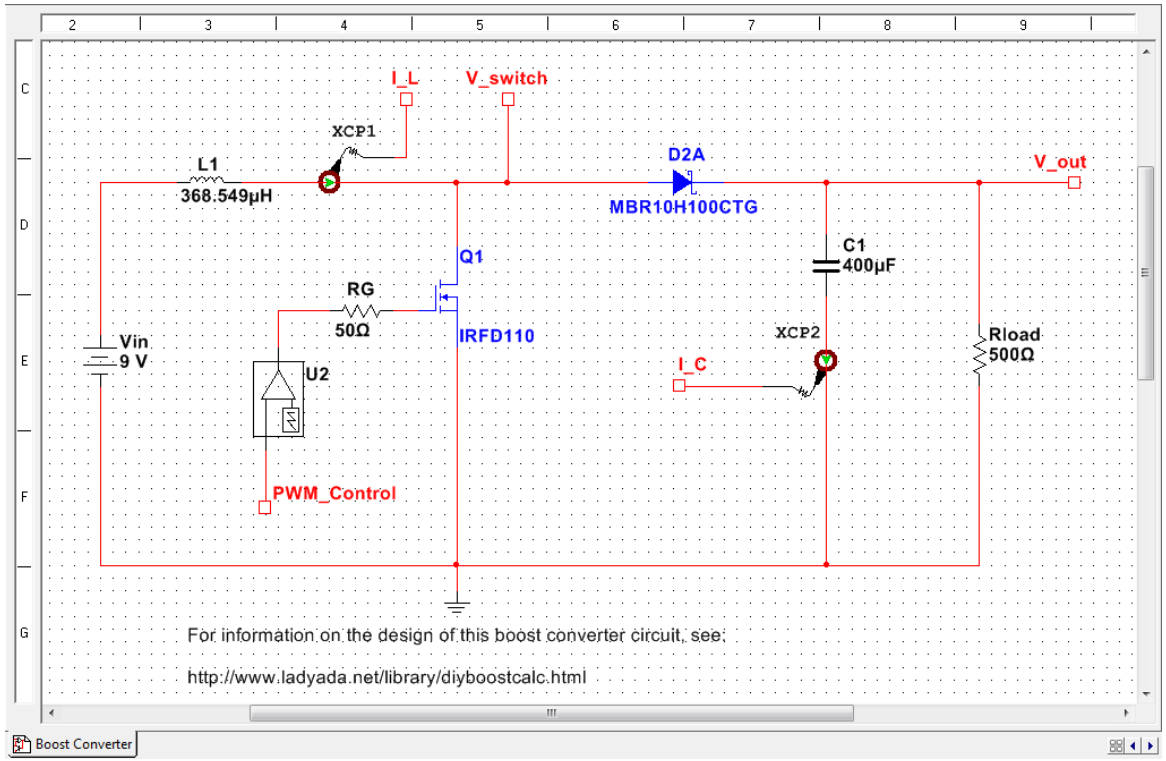
- Using a variable time step solver in LabVIEW
  - Double-click the LabVIEW Control & Simulation Loop to change the simulation settings
  - How does changing the **Minimum Step Size (s)** value on the LabVIEW front panel affect the simulation results chart?
- Using  vendor specific detailed part models in your Multisim circuit
  - Blue circuit symbols indicate detailed (rather than idealized) vendor defined part models
  - Examine the International Rectifier N-Channel Power MOSFET, Q1, and ON Semiconductor rectifier diode, D1, in the circuit
    - Download the datasheets for these components and examine their properties. Are they an appropriate choice for this circuit?

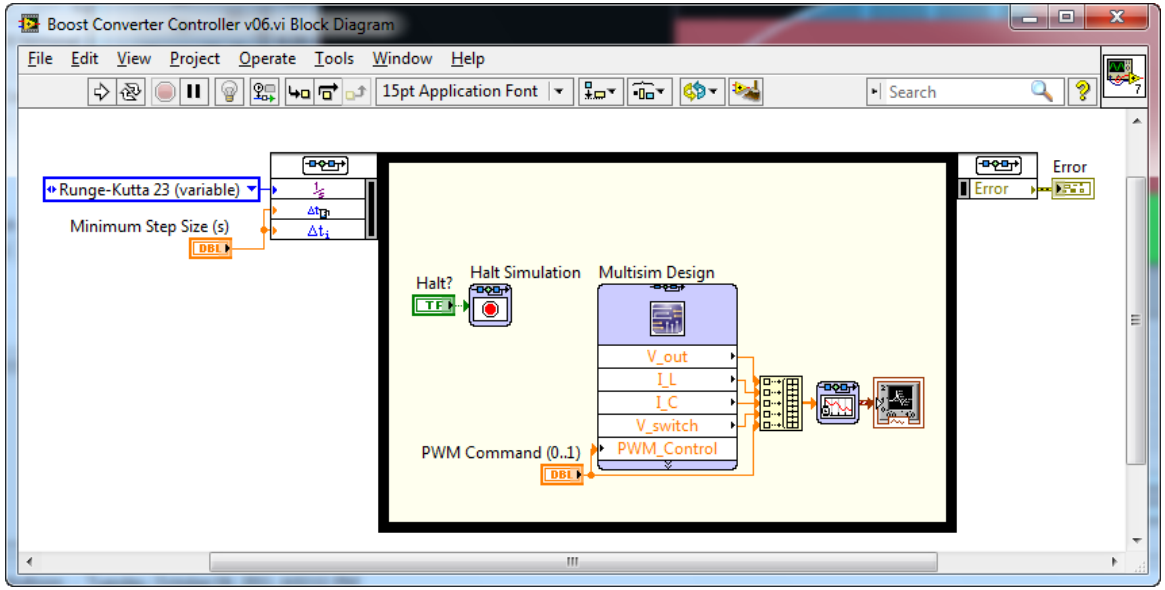
**Power Electronics Concepts**

- Fundamentals of boost converter switched-mode power supply circuits, a step-up DC to DC converter (learn more)
- How source voltage energy is stored in the inductor and then discharged into the load
  - Can you explain the relationship between the switch voltage (**V_switch**) and the inductor current (**I_L**) signals?
  - Can you use LabVIEW to calculate and chart the power flowing into the inductor (P=I*V)?
- Understanding how output voltage ripple is effected by the output capacitor value
  - To control the capacitance from LabVIEW, replace the load resistor with a voltage controlled variable capacitor (**Basso/VARICAP**).
  - Can you measure the AC RMS value of the output voltage ripple using a LabVIEW analysis VI?
- Continuous versus discontinuous conduction modes of operation
  - To control the load resistance from LabVIEW, replace it with a voltage controlled variable resistor (**Basso/VARIRES**).
  - Can you determine the conditions in which the circuit transitions from continuous to discontinuous conduction modes?

**Required Toolkits and Modules**

- NI LabVIEW 2011 (ni.com/labview)
- NI LabVIEW Control Design & Simulation Module 2011 (help files)
- NI Circuit Design Suite 12.0 Beta 0 (or higher)(ni.com/multisim)


I_L
V_switch
XCP1
L1
368.549µH
D2A
MBR10H100CTG
V_out
Q1
C1
400µF
RG
50Ω
IRFD110
Vin
9 V
U2
XCP2
I_C
Rload
500Ω
PWM_Control
For information on the design of this boost converter circuit, see:
http://www.ladyada.net/library/diyboostcalc.html
Boost Converter


Boost Converter Controller v06.vi Block Diagram
File Edit View Project Operate Tools Window Help
15pt Application Font
Search
Runge-Kutta 23 (variable)
Minimum Step Size (s)
DBL
Error
Error
Halt?
TF
Halt Simulation
Multisim Design
V_out
I_L
I_C
V_switch
PWM_Control
PWM Command (0..1)
DBL

Online design tool calculations (http://www.ladyada.net/library/diyboostcalc.html):

## The calculator

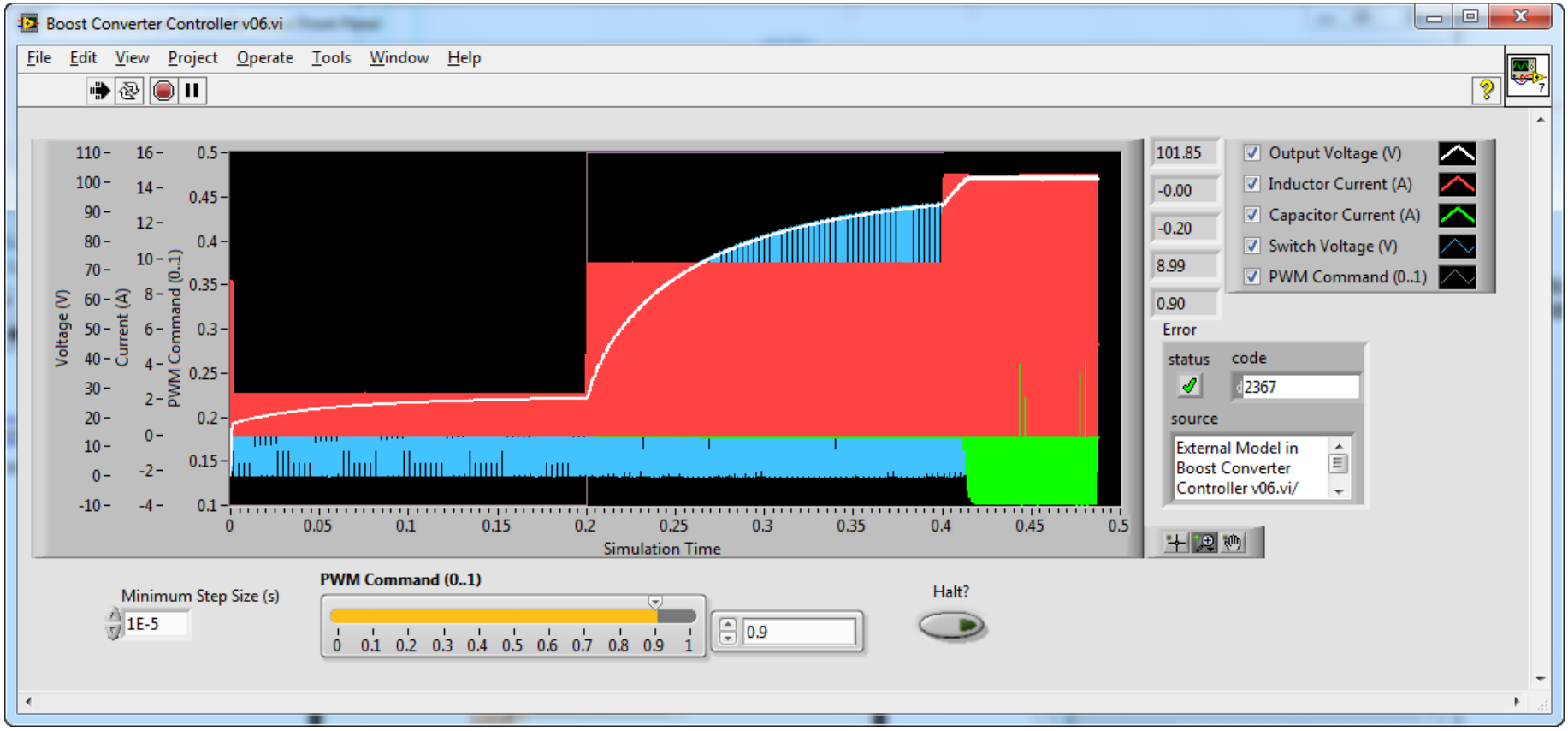
| | | |
|---|---|---|
| **Frequency** | 5000 Hz | This is the boost converter frequency. For microcontrollers its often the CPU clock / 256 |
| **Min Vin** | 9 V | The lowest expected input voltage |
| **Max Vin** | 9 V | The highest expected input voltage |
| **Min Vout** | 9 V | The lowest desired output voltage |
| **Max Vout** | 100 V | The highest desired output voltage |
| **Iout** | 0.20 Amps | Output current draw |
| **Vripple** | 0.1 V | Maximum allowable voltage ripple |

| | | |
|---|---|---|
| **Min. Duty Cycle** | **Dmin = 1 - (Vimax/Vomin)** | 0 % |
| **Max. duty cycle** | **Dmax = 1 - (Vimin/Vomax)** | 91 % |
| **Min. Inductor size** | **L > D * Vin * (1-D) / (freq * 2 * Iout )** | 368.54 uH |
| **Peak inductor current** | **Ipk = (Vinmax * D)/(f * L)** | 4.4444 A |
| **Minimum capacitor** | **Cap > Iout / (Vripple * freq)** | 400 uF |
| **Minimum Schottky diode** | Vbreakdown >= Voutmax & Idiode >=Ipk | 100 V<br>4.44 A |

Don't forget that Duty cycle is the amount of time the switch is **off** / output is low. Be sure to measure the output voltages before connecting up anything important!

**Simulation Results**

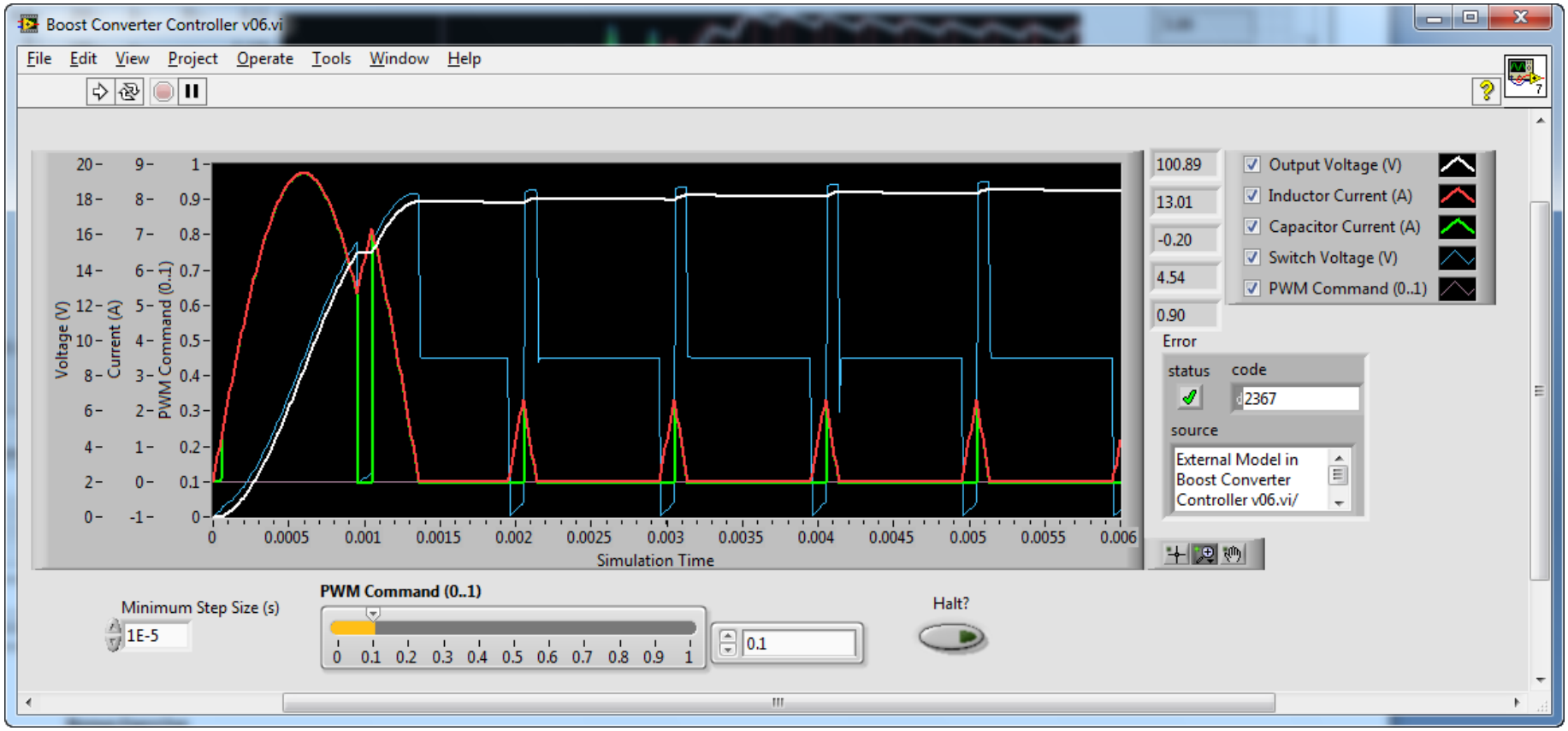
- PWM Command is changed from 0.1 to 0.5 to 0.9



Initial transients:

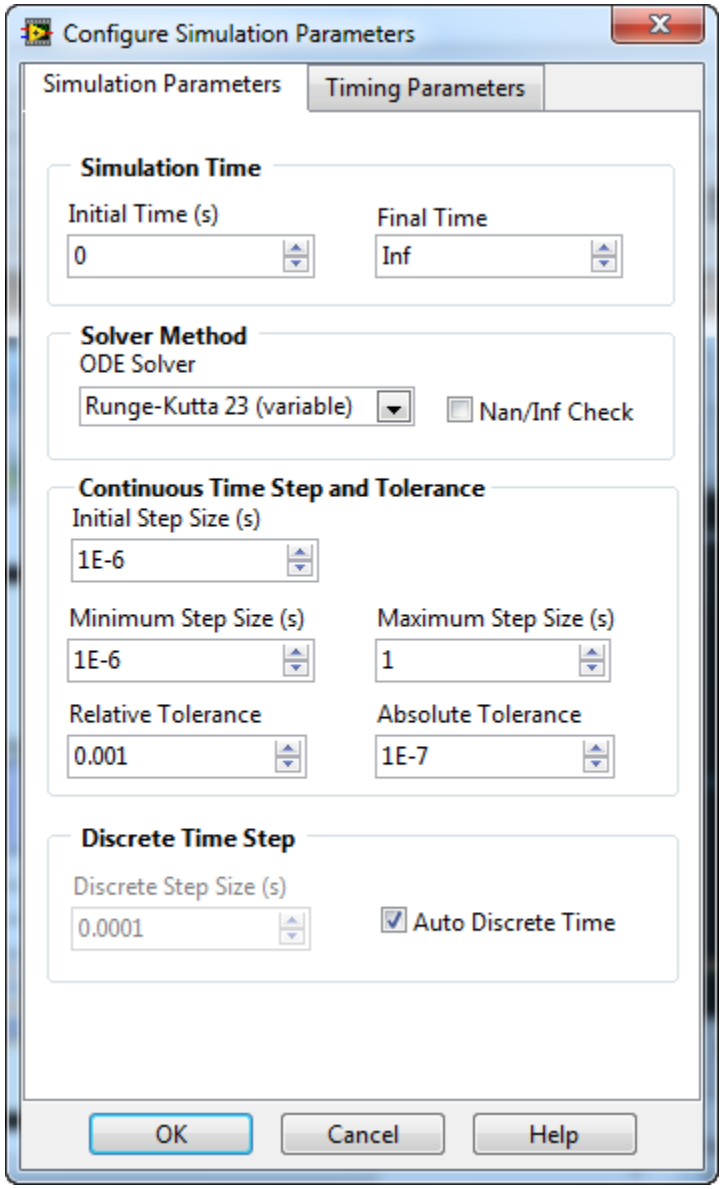
- PWM Command is 0.1:

1. Charging of the inductor and capacitor
2. Boosting the energy of the inductor
3. Transfer of energy from the inductor
4. After inductor current decays to zero and the diode opens, the switch voltage Is equal to the supply voltage, Vin
    - Is the circuit operating in continuous or discontinuous conduction mode? How does changing the load resistance affect the response?



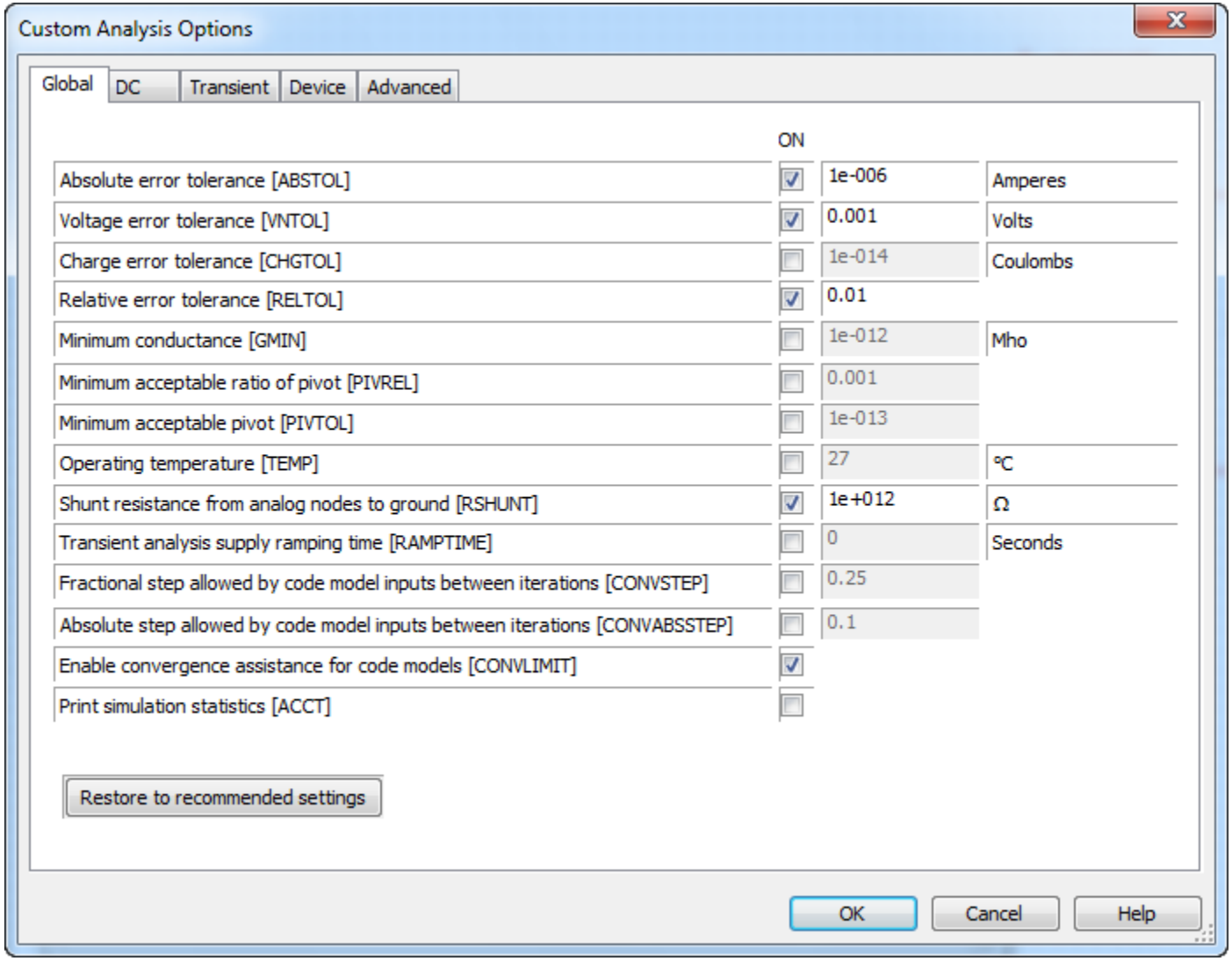
Configuring simulation settings in LabVIEW:

- On the LabVIEW block diagram, double-click the upper left corner of Control & Simulation Loop to configure the simulation parameters

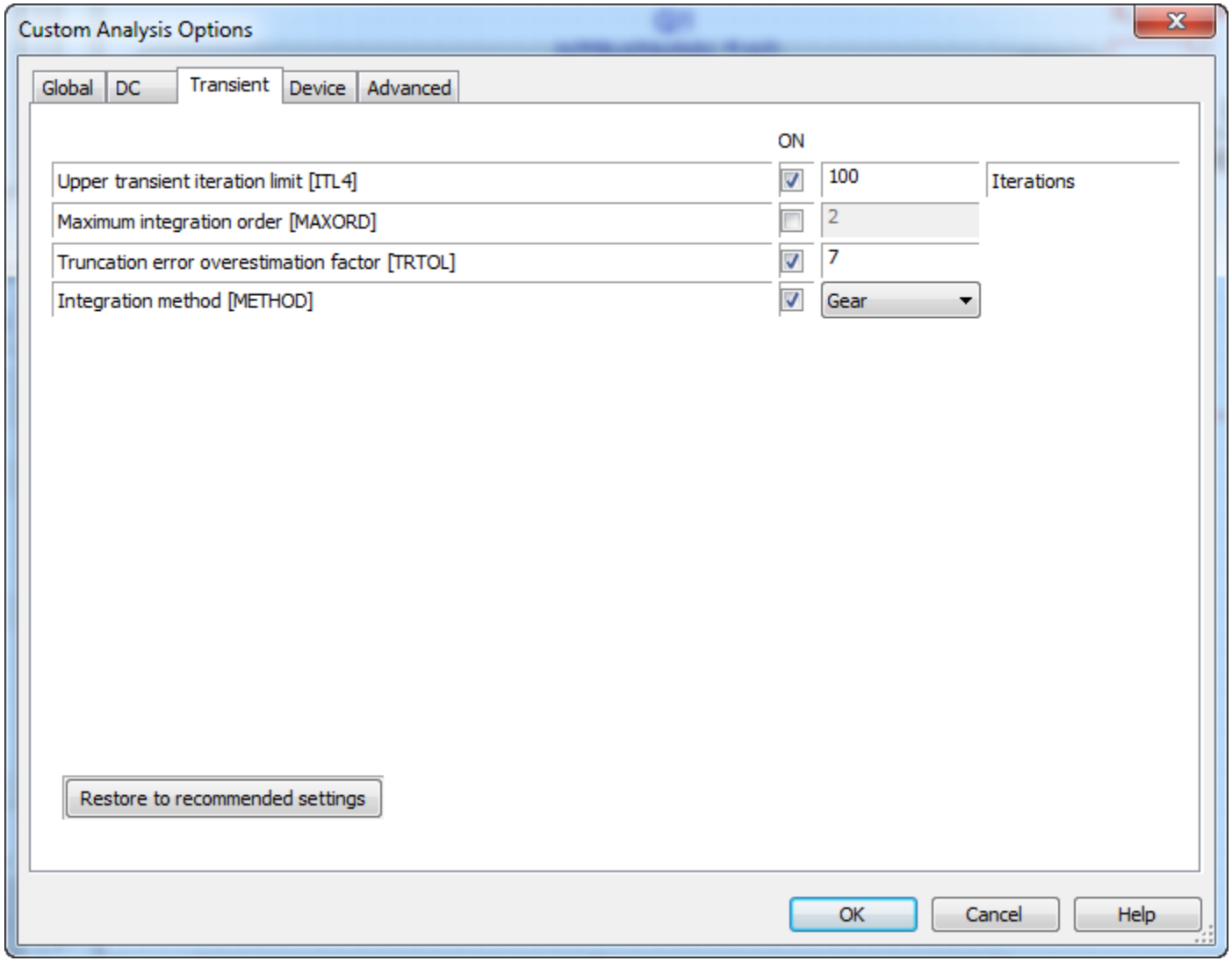


Configuring the simulation settings in Multisim:

- In Multisim, navigate to **Simulate>Interactive Simulation Settings**.
- Navigate to **Analysis Options>Use custom settings>Customize…**
- Configure the **Custom Analysis Options**
    - Switch-mode power supply simulations often use looser absolute, voltage and relative error tolerance settings to aid convergence



- The gear integration method is also recommended for switch-mode power simulations




- Note: In Multisim you can save and load simulation setting configuration files via the **Simulate** menu

**Bonus Exercise**

- To understanding the impact of different diode models in Multisim, try replacing the diode with one not rated to handle the current, such as D1N5822_ONSEMI. What is the circuit behavior if the diode is in breakdown?

# Buck Converter (Closed Loop PID Control)

**Co-Simulation Concepts**

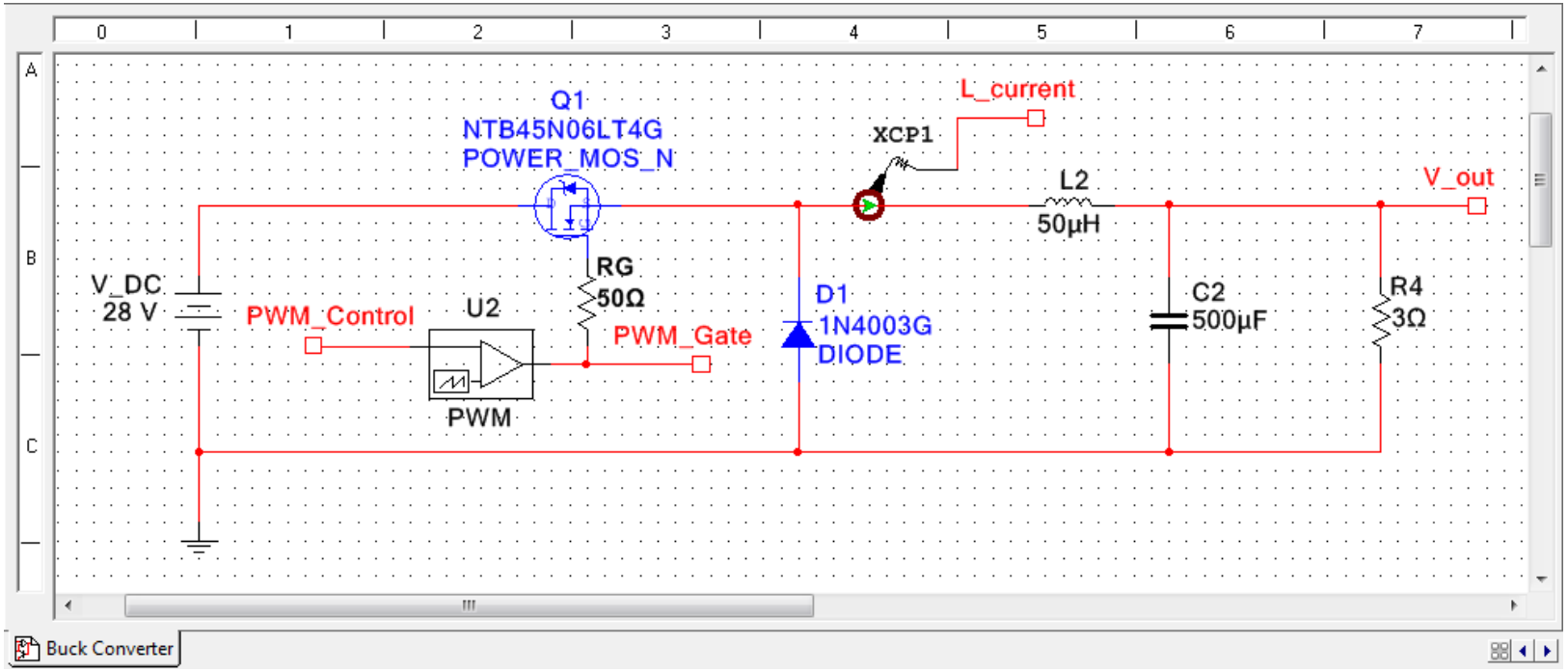
- Adding LabVIEW FPGA subVIs to your LabVIEW Control Design & Simulation loop
  - The discrete time settings on the LabVIEW subVIs enable simulations to replicate the precise timing behavior of execution in a field programmable gate array (FPGA) or real-time processor.
- Using **Memory Blocks** in LabVIEW to create feedback paths for closed loop control
  - Configure the discrete time settings to simulate analog to digital converter (ADC) sampling behavior.

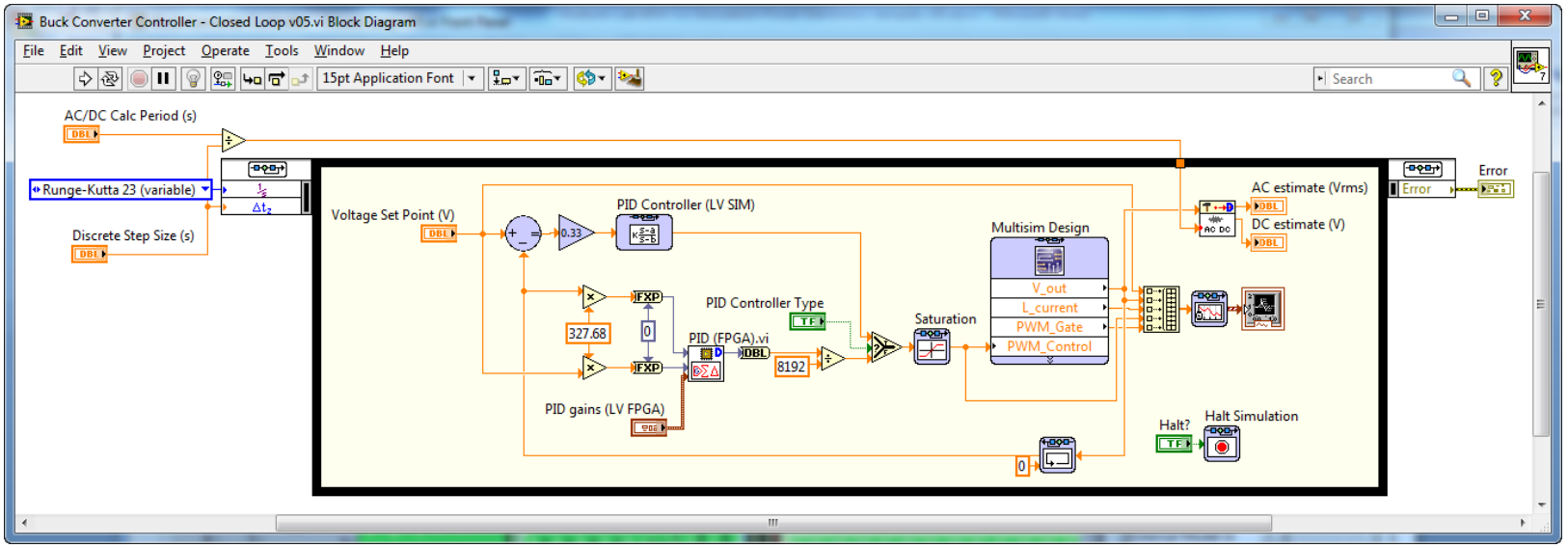
**Power Electronics Concepts**

- Fundamentals of buck converter circuits, a step-down DC to DC converter switched-mode power supply (learn more)
- How source voltage energy is stored in the inductor and then discharged into the load
  - Can you explain the relationship between the gate command (**PWM_Gate**) and the inductor current (**L_current**) signals?
  - Can you calculate and chart the energy stored in the inductor in LabVIEW (E=0.5L*I_L^2)?
- Understanding how output voltage ripple is effected by the output capacitor
  - To control the capacitance from LabVIEW, try replacing the load resistor with a voltage controlled variable capacitor (**Basso/VARICAP**).
  - Can you measure the AC coupled RMS value of the output voltage ripple using a LabVIEW analysis VI?
- Continuous versus discontinuous conduction modes of operation
  - To control the load resistance from LabVIEW, replace it with a voltage controlled variable resistor (**Basso/VARIRES**).
  - Can you determine the conditions in which the circuit transitions from continuous to discontinuous conduction modes?

**Required Toolkits or Modules**

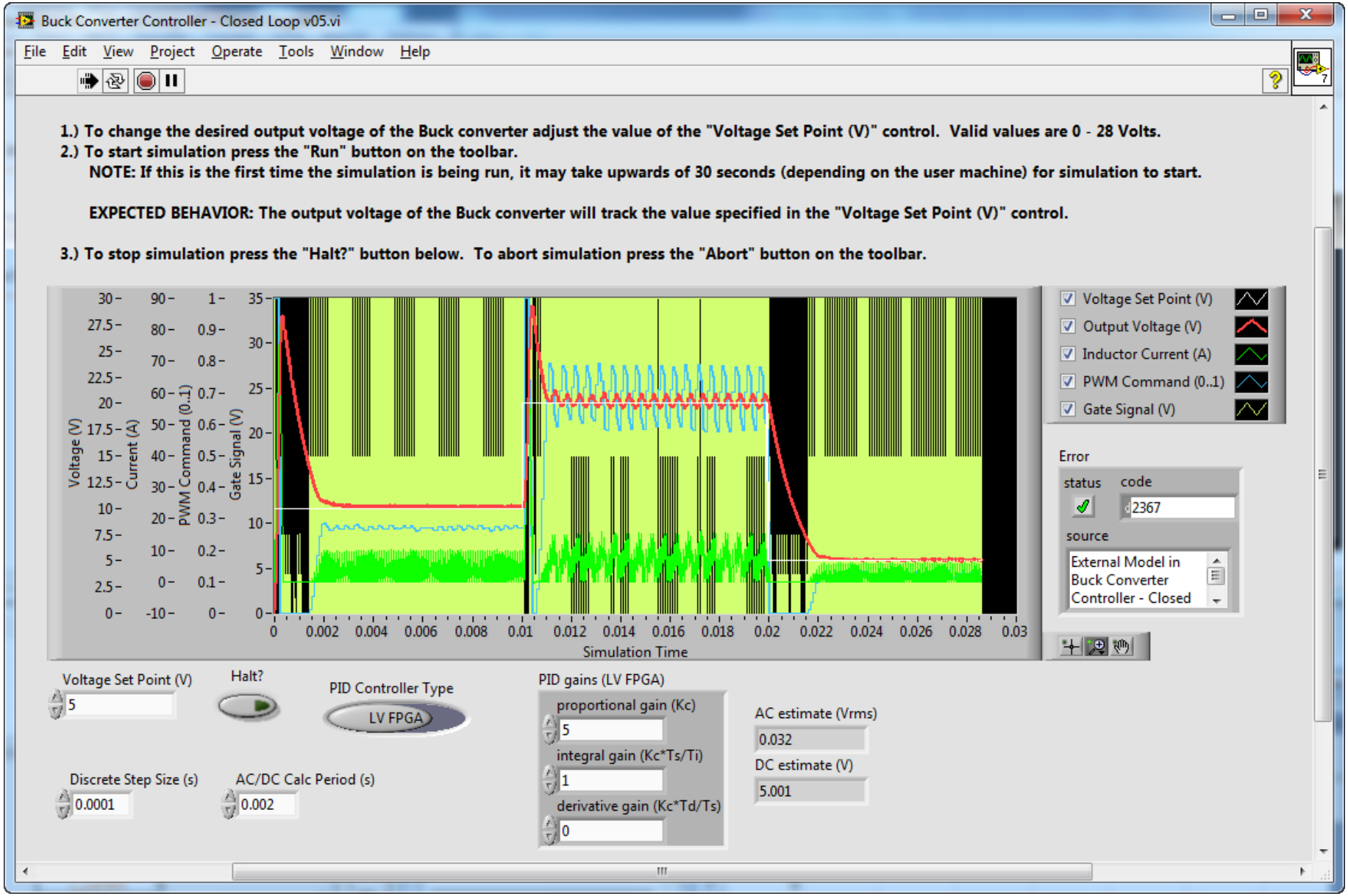
- NI LabVIEW 2011 (ni.com/labview)
- NI LabVIEW FPGA Module 2011 (ni.com/fpga)
- NI LabVIEW Control Design & Simulation Module 2011 (help files)
- NI Circuit Design Suite 12.0 Beta 0 (or higher)(ni.com/multisim)


Q1
NTB45N06LT4G
POWER_MOS_N
XCP1
L_current
L2
50µH
V_out
V_DC
28 V
PWM_Control
U2
RG
50Ω
PWM_Gate
D1
1N4003G
DIODE
C2
500µF
R4
3Ω
PWM
Buck Converter


Buck Converter Controller - Closed Loop v05.vi Block Diagram
File Edit View Project Operate Tools Window Help
15pt Application Font
Search
AC/DC Calc Period (s)
Runge-Kutta 23 (variable)
Discrete Step Size (s)
Voltage Set Point (V)
PID Controller (LV SIM)
0.33
PID Controller Type
PID (FPGA).vi
327.68
8192
PID gains (LV FPGA)
Saturation
Multisim Design
V_out
L_current
PWM_Gate
PWM_Control
AC estimate (Vrms)
DC estimate (V)
Error
Halt?
Halt Simulation

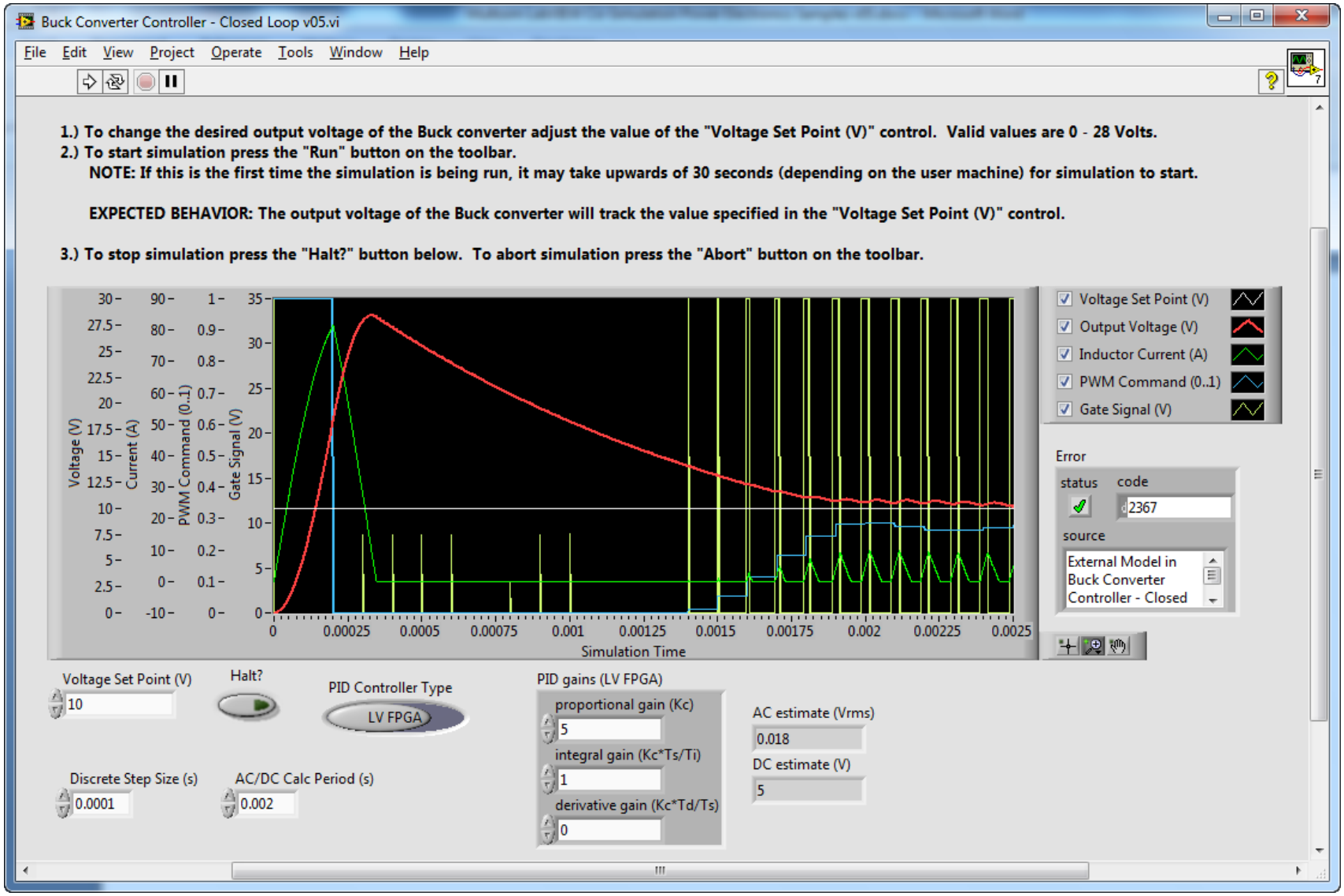
**Simulation Results**

- Closed loop control response when changing the **Voltage Set Point (V)** from 10 to 20 to 5 V:



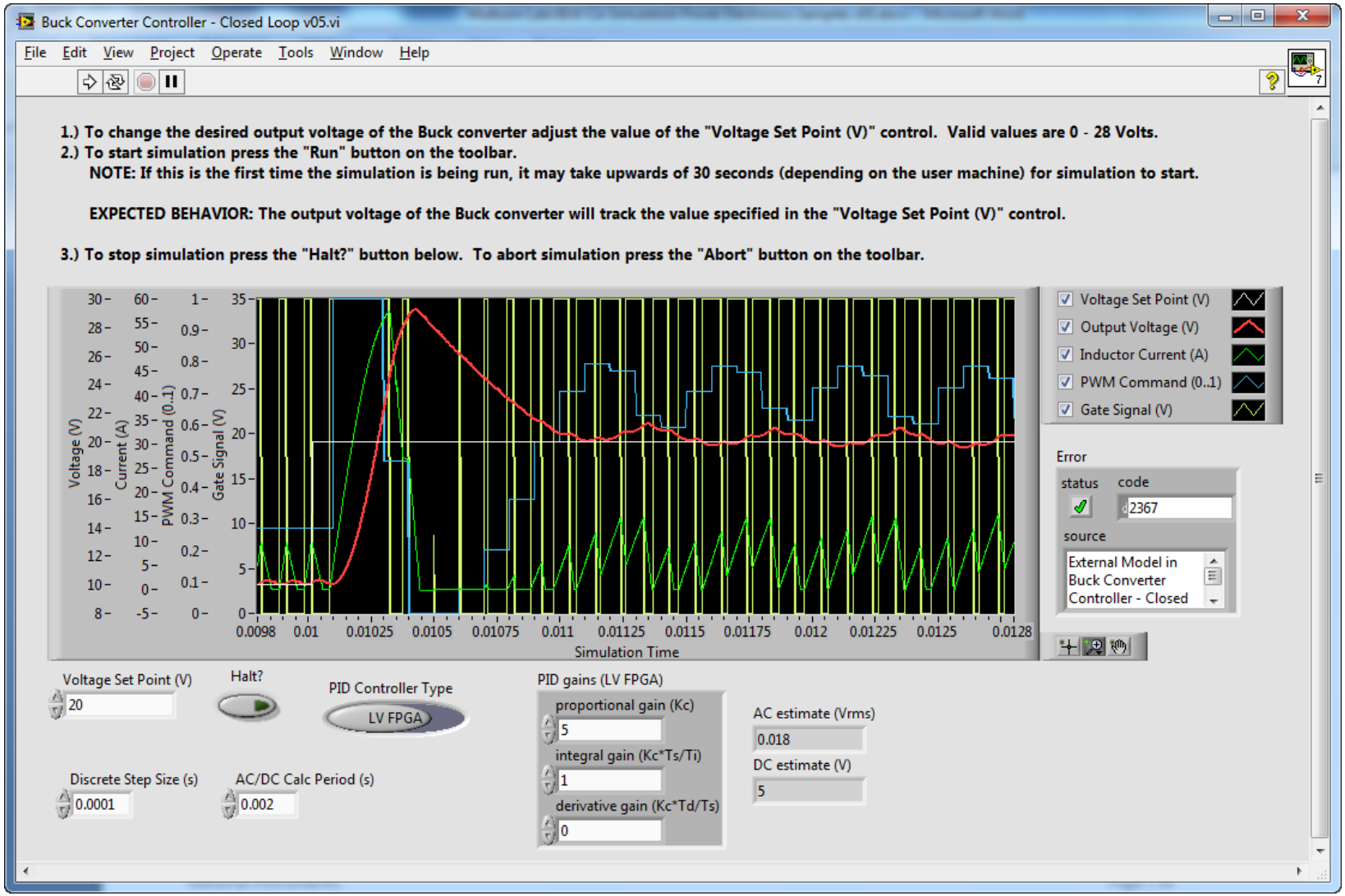
Initial transient response:

- **Voltage Set Point (V)** = 10 V
    - What is the percent overshoot, steady state error and AC output voltage ripple for the closed loop system?
        - Can you improve the tuning of the PID control loop?



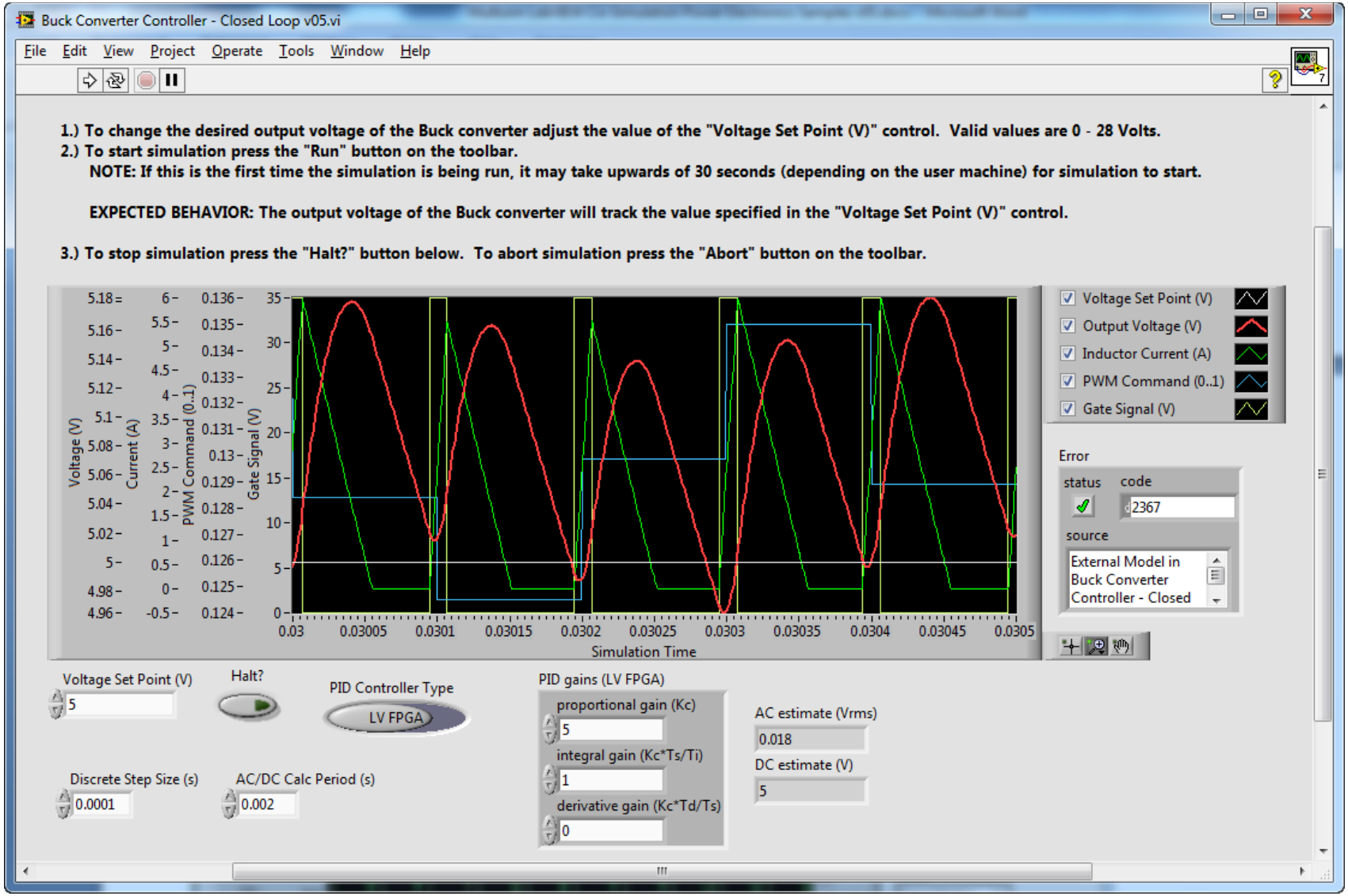
Set point change response:

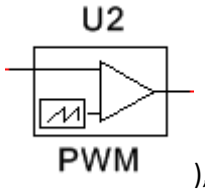
- **Voltage Set Point (V)** changed from 10 V to 20 V
    - What's the primary frequency of oscillations of the output voltage waveform?
        - How could you modify the circuit design to change the oscillation frequency and reduce the output voltage ripple?



Steady-state response:

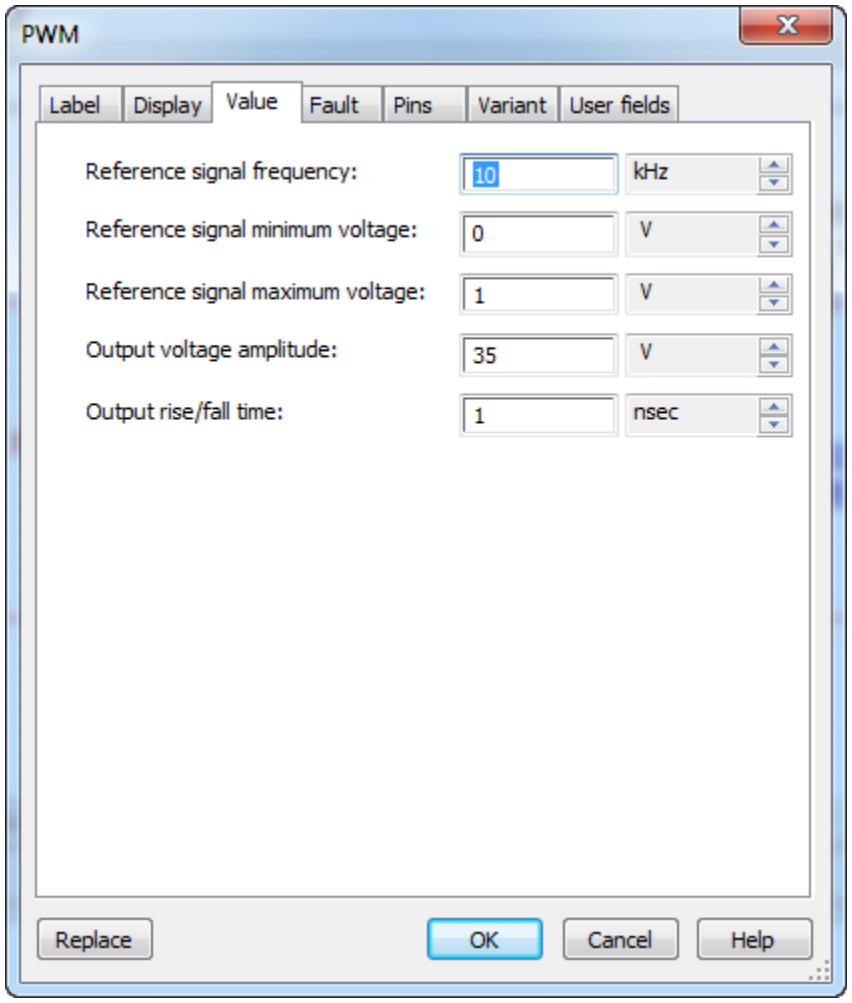
- **Voltage Set Point (V)** is 5 V
- 50 microseconds of time shown below. As you can see, the PWM frequency is 10 kHz.
- Can you explain the relationship between **Gate Signal (V)** and the **Inductor Current (A)**?
    - Is the circuit operating in continuous or discontinuous conduction mode? How does changing the load resistance affect the response?



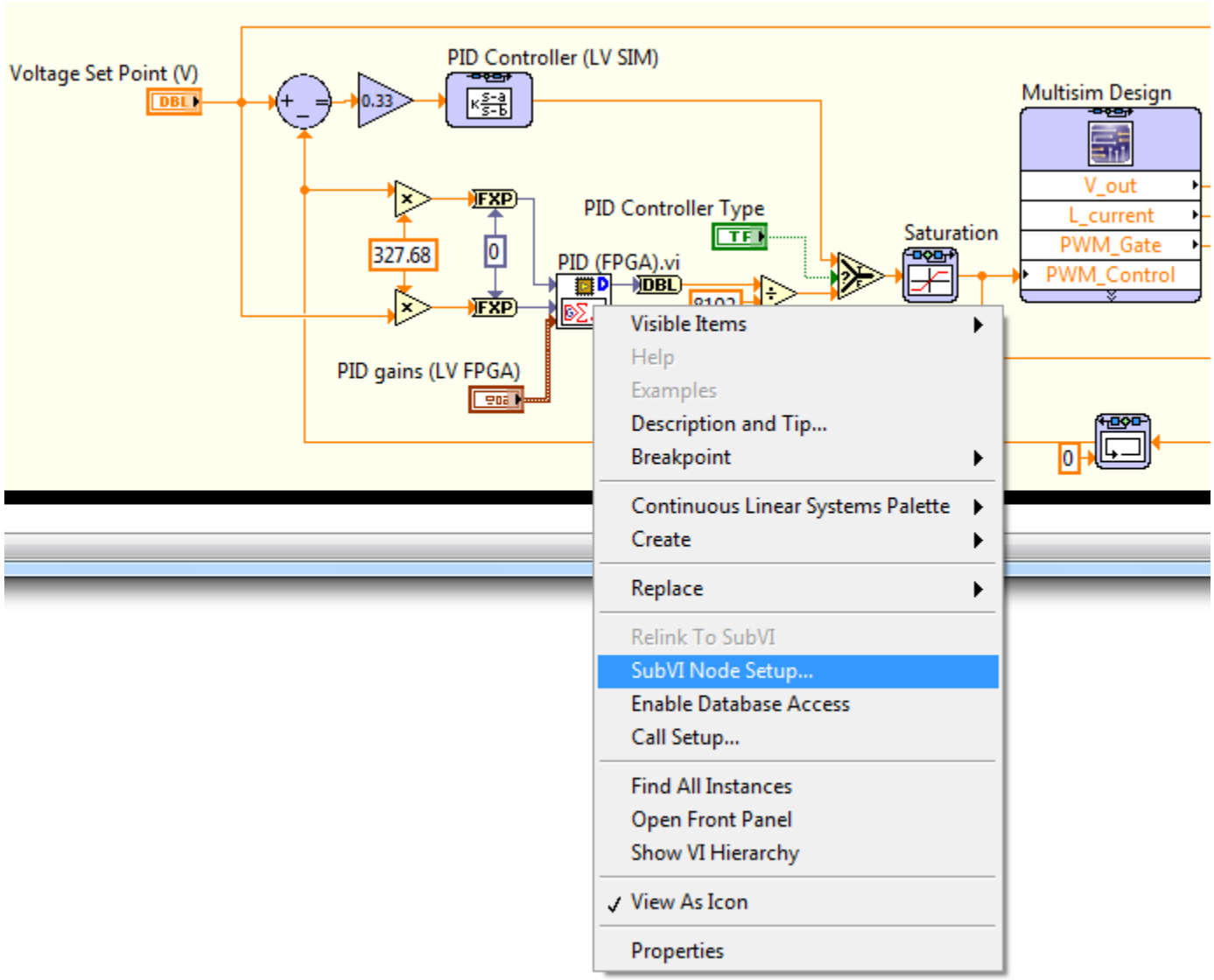



To change the settings for the PWM generation component ( ), double click it in Multisim.

- In this case, we are generating a 10 kHz PWM signal with 35 V amplitude and 1 nsec rise/fall time:

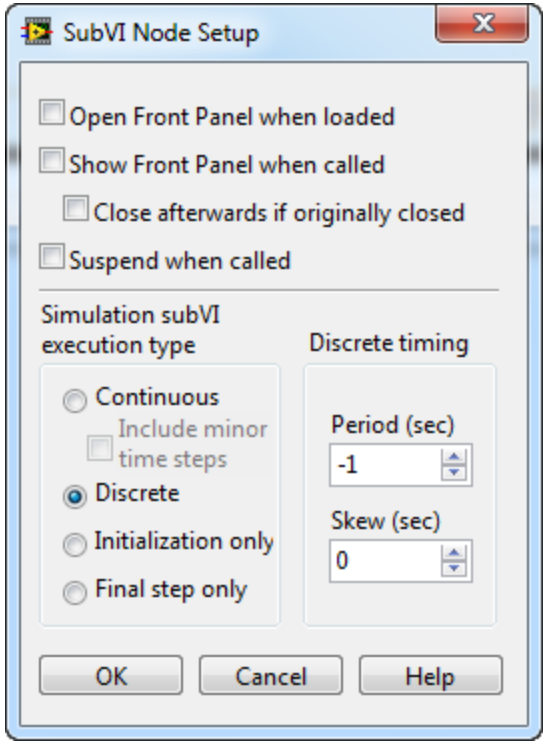


To change the LabVIEW FPGA execution time settings,right-click on the subVI in LabVIEW and navigate to **SubVI Node Setup…** as shown below:




If you set the **Discrete timing Period (sec)** to -1, the subVI will execute at the **Discrete Step Size (s)** rate set in the simulation parameters or terminal.

- Alternately, you can set the **Discrete timing Period (sec)** to a specific value, such as 25e-9 sec to simulate code executed within a single cycle timed loop (SCTL) in LabVIEW FPGA with the FPGA base clock set to 40 MHz.

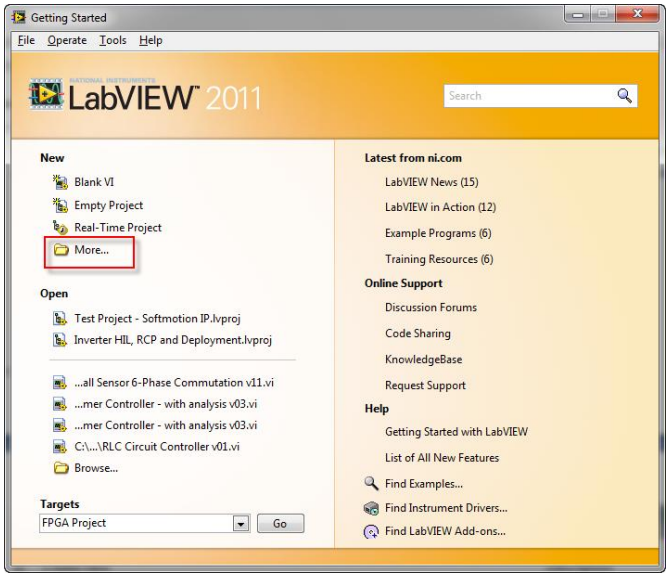



**Known Issues**

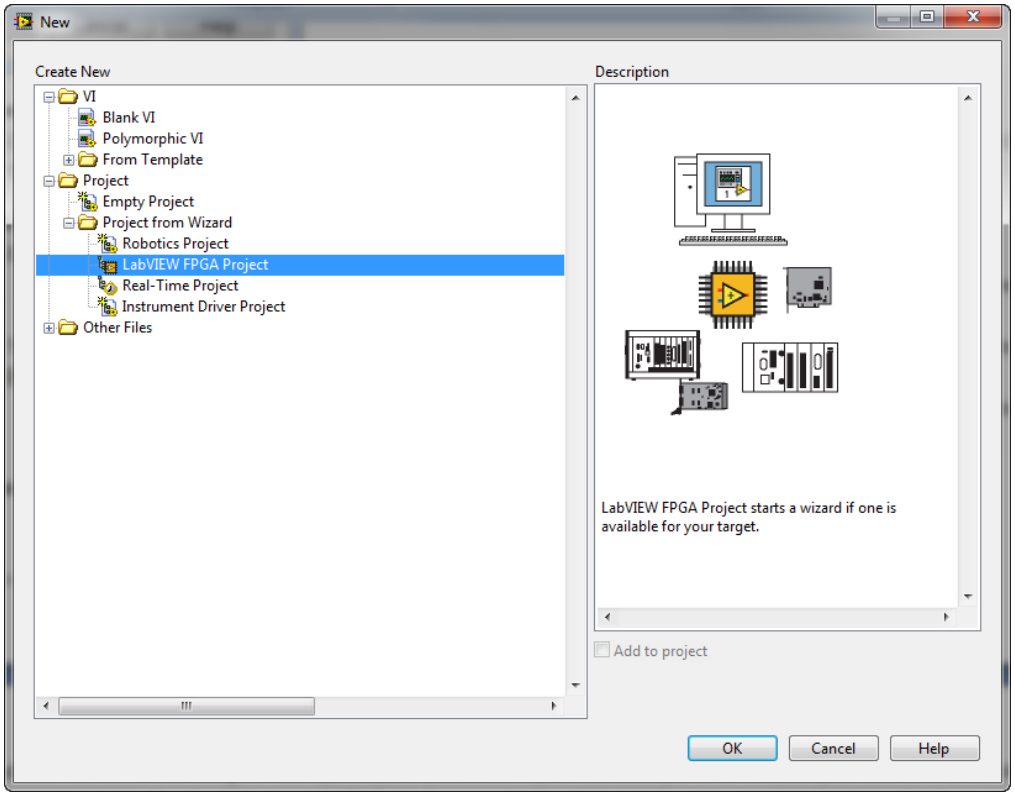
- Currently, some LabVIEW FPGA blocks must be converted to subVIs before executing inside the LabVIEW Control Design & Simulation loop. To do this, right click the block and select **Convert to SubVI**.

**Suggested Exercises**

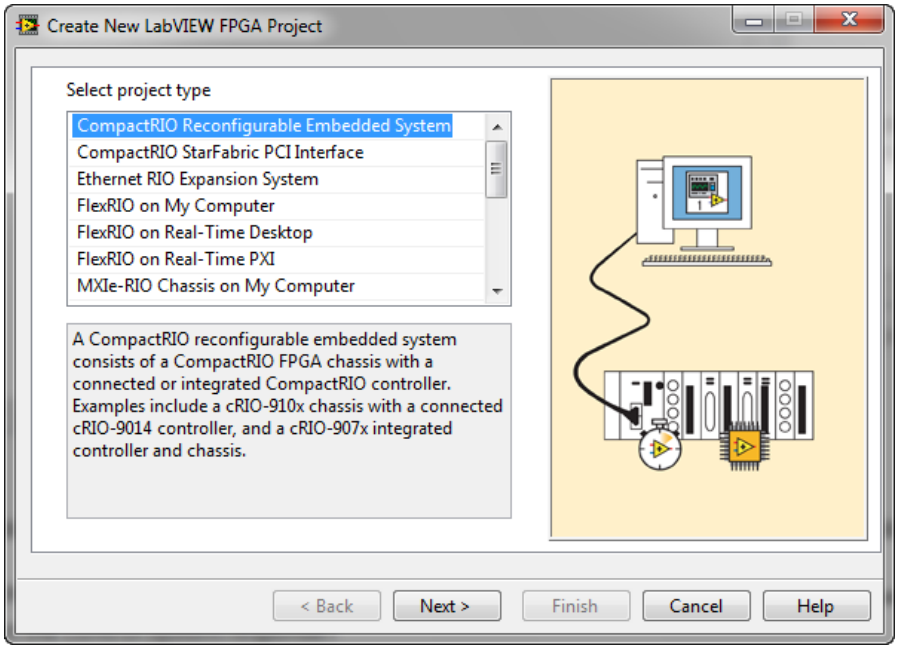
- The LabVIEW FPGA based PID control block is set to execute at 100 kHz (dT = 1e-5 s). How does changing the control loop rate of the **PID (FPGA).vi** affect the control system response?
    - Hint: Right-click on a subVI to set the discrete timing information. This enables you to specify the execution timing behavior for your LabVIEW FPGA subVIs.
- Explore the LabVIEW FPGA palette and try adding another IP block to your simulation.
    - Hint: To add a LabVIEW FPGA subVI to your desktop Control Design and Simulation loop, you'll need to create a LabVIEW Project containing an FPGA target. To do this, close all open LabVIEW VIs and from the LabVIEW splash screen, click **More…**



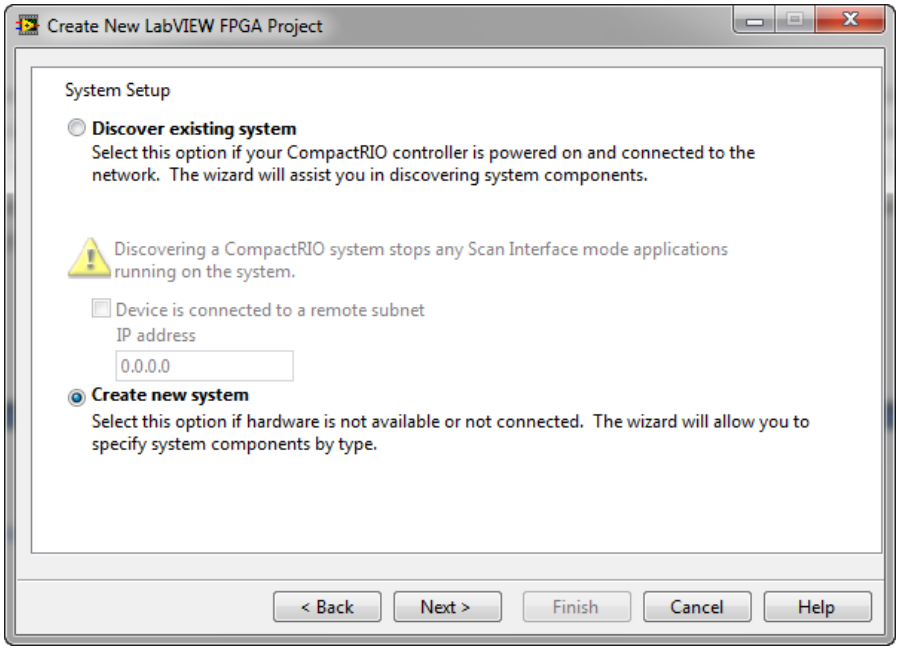

- Next, select **LabVIEW FPGA Project**.



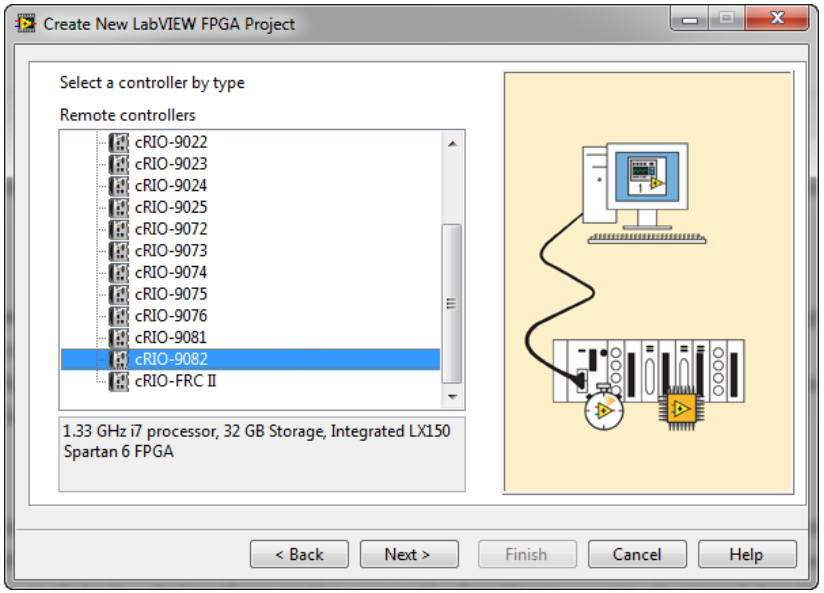
- Select **CompactRIO Reconfigurable Embedded System** and then click **Next**.



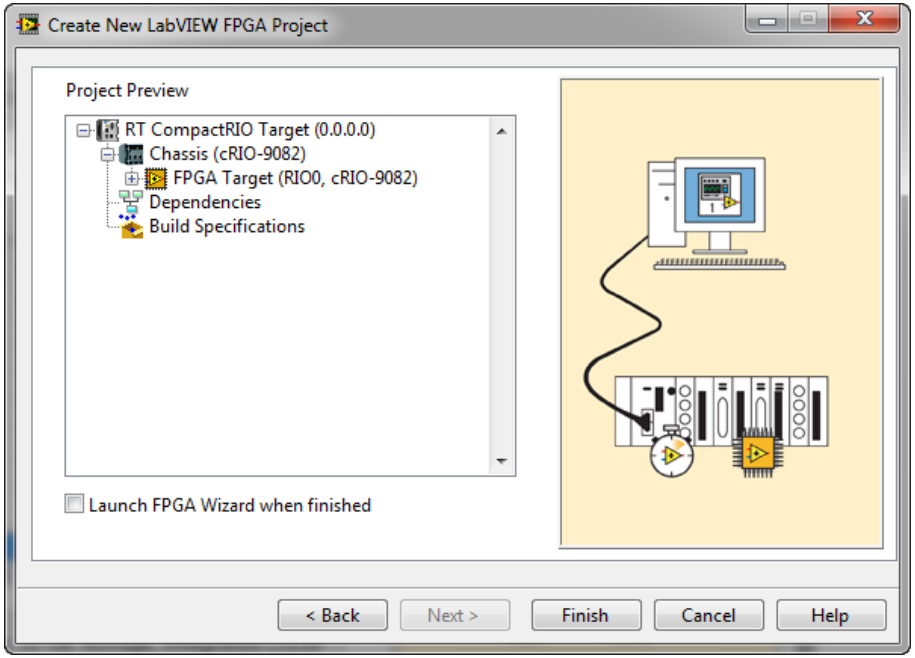

- If your CompactRIO system is not current connected to the network, select **Create New System**, then click **Next**.



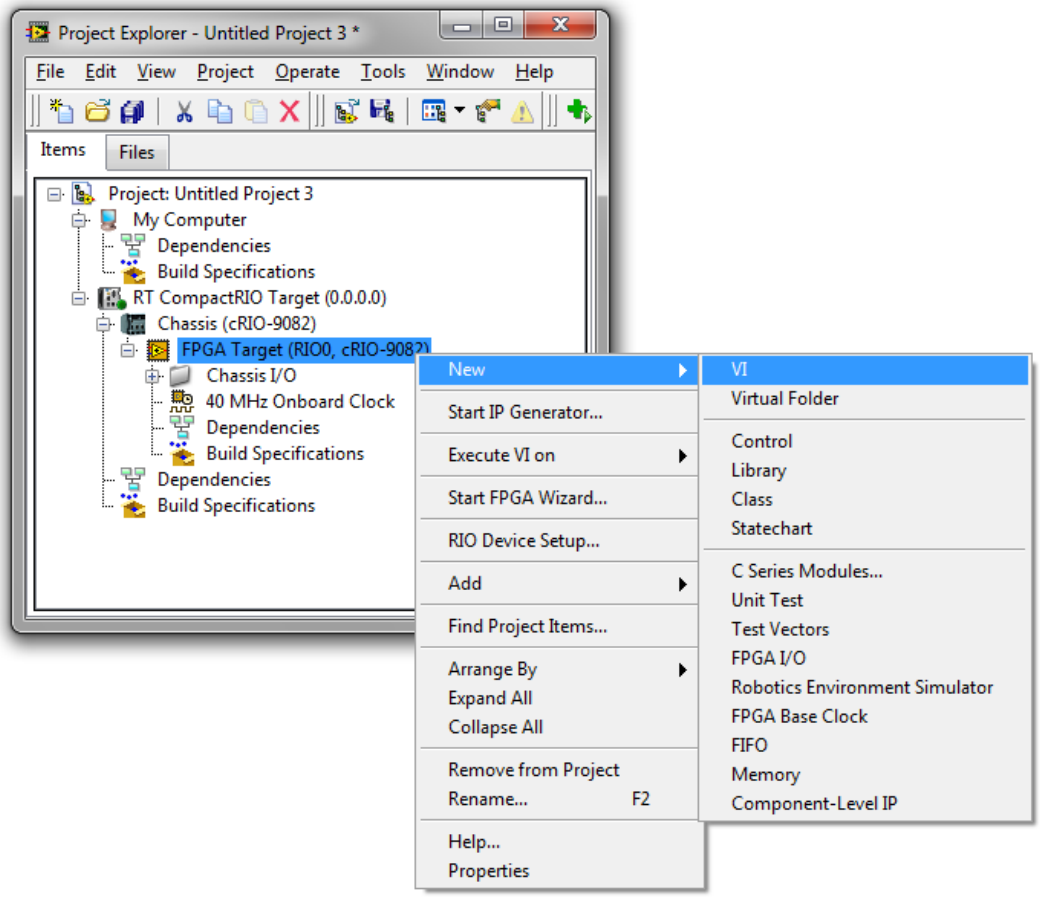
- Select the name of a CompactRIO system, such as **cRIO-9082** and click **Next**. The NI cRIO-9082 is a high performance Multicore processor based CompactRIO system and is ideal for the rapid control prototyping of power electronics systems.



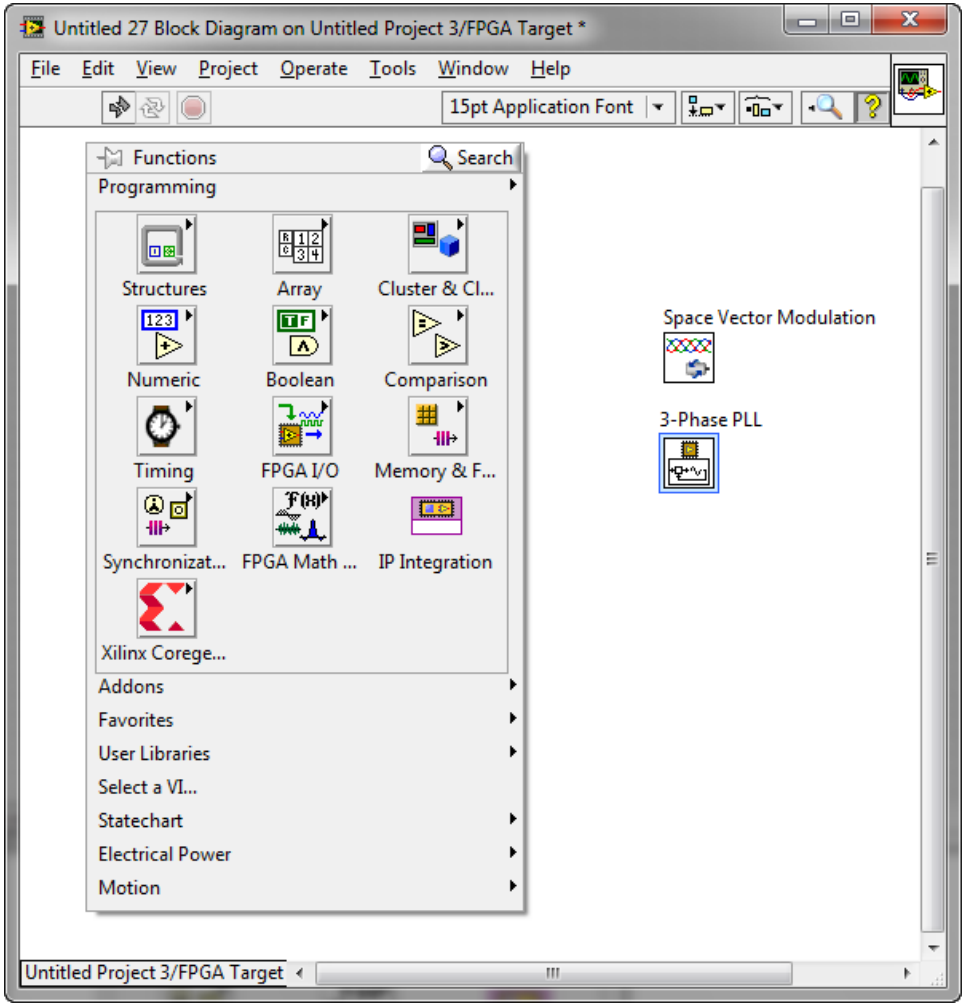

- Add modules to the chassis if desired, or skip this step and click **Next**. Finally, click **Finish** to create your LabVIEW Project.



- In the Project Explorer, right click on the **FPGA Target** and select **New>VI**.



- Right-click on an empty area of the block diagram and explore the **LabVIEW FPGA** functions palette.
    - If you have NI SoftMotion 2011 f1 or higher installed, be sure to look at the IP blocks in the NI power electronics IP library under the **Motion** sub-palette. This includes new officially supported LabVIEW FPGA IP blocks for space vector field oriented control.
    - Also, be sure to take a look at the 3-Phase PLL IP block located under the sub-pallete **FPGA Math & Analysis>Control**.




    - You can either drop the LabVIEW FPGA subVI directly onto the block diagram of your LabVIEW/Multisim simulation loop, or copy and paste them from your LabVIEW FPGA application.
        - Try adding a LabVIEW FPGA analysis or digital signal processing function of your choice to your **Buck Converter Controller** LabVIEW co-simulation application.

**Bonus Exercise**

- In addition to the two existing control options, can you add an option for a simple FPGA-based hysteresis controller? A hysteresis controller works by simply turning the switch on or off depending on whether the output voltage is below or above the desired value. To prevent damaging the transistor due to excessively high switching frequencies, use a relay function to create a hysteresis band around the setpoint.
- Can you implement a synchronous rectification control scheme by replacing the diode with an active switch device and writing a LabVIEW FPGA IP block to control it from LabVIEW?

## 3-Phase Single Level Inverter

**Co-Simulation Concepts**

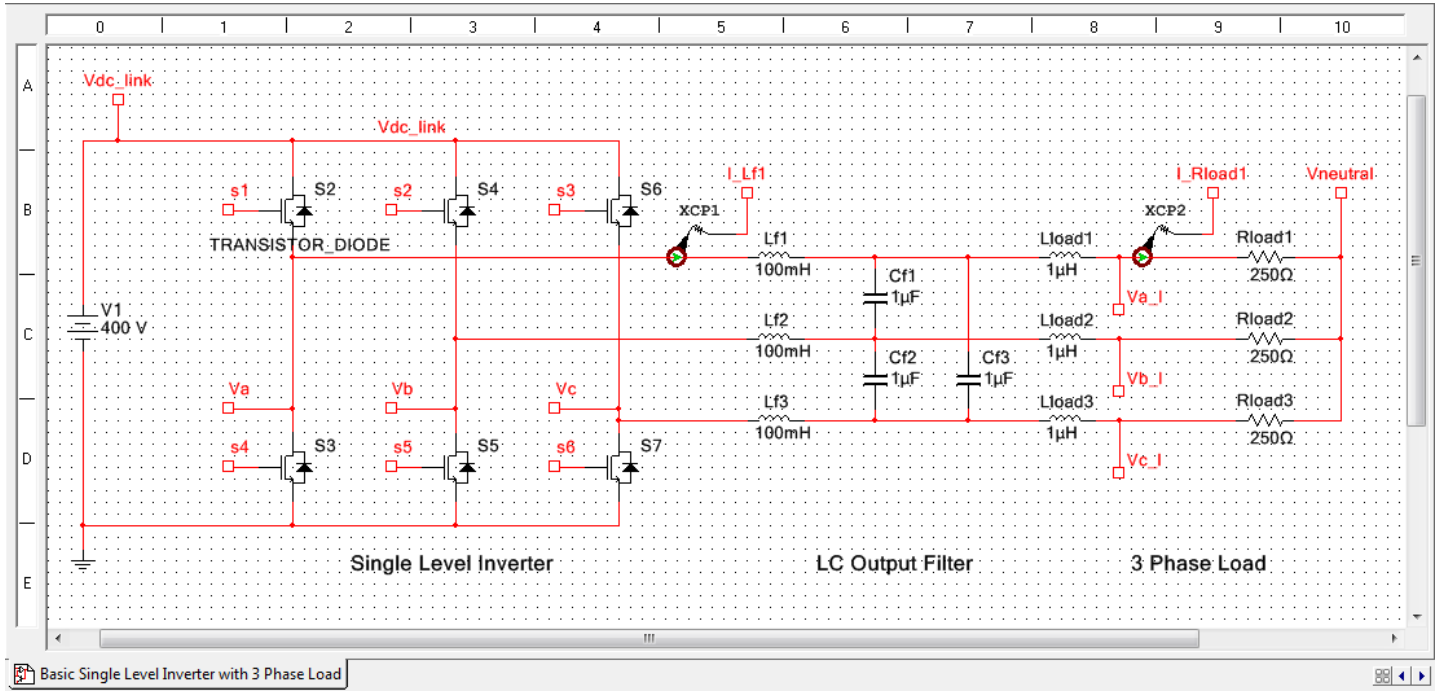
- Simulating the timing behavior of LabVIEW FPGA subVIs executing at different rates
    - In this application, the load voltage RMS calculation is executed in the data acquisition loop for an NI 9225 3-Channel, 300 Vrms analog input module executing at 50,000 samples/second, while the control loop executes at 500 kHz (or 80 ticks of the 40 MHz FPGA clock).
    - To create this example, LabVIEW FPGA code from an NI Single-Board RIO application was copied into the LabVIEW Control & Simulation loop. Then the LabVIEW FPGAs subVIs were set to their appropriate discrete time rates by right-clicking and navigating to **SubVI Node Setup** as explained in previous section.
        - See screenshots below of the actual Single-Board RIO deployment code
        - Watch a video demonstration of a real-time HIL simulation of this inverter (link)
- Examining the startup behavior of three phase power circuits and LabVIEW FPGA control algorithms
    - How does the PID loop respond before the first RMS calculation is available? Can you improve the control algorithm response during the first 60 Hz cycle?

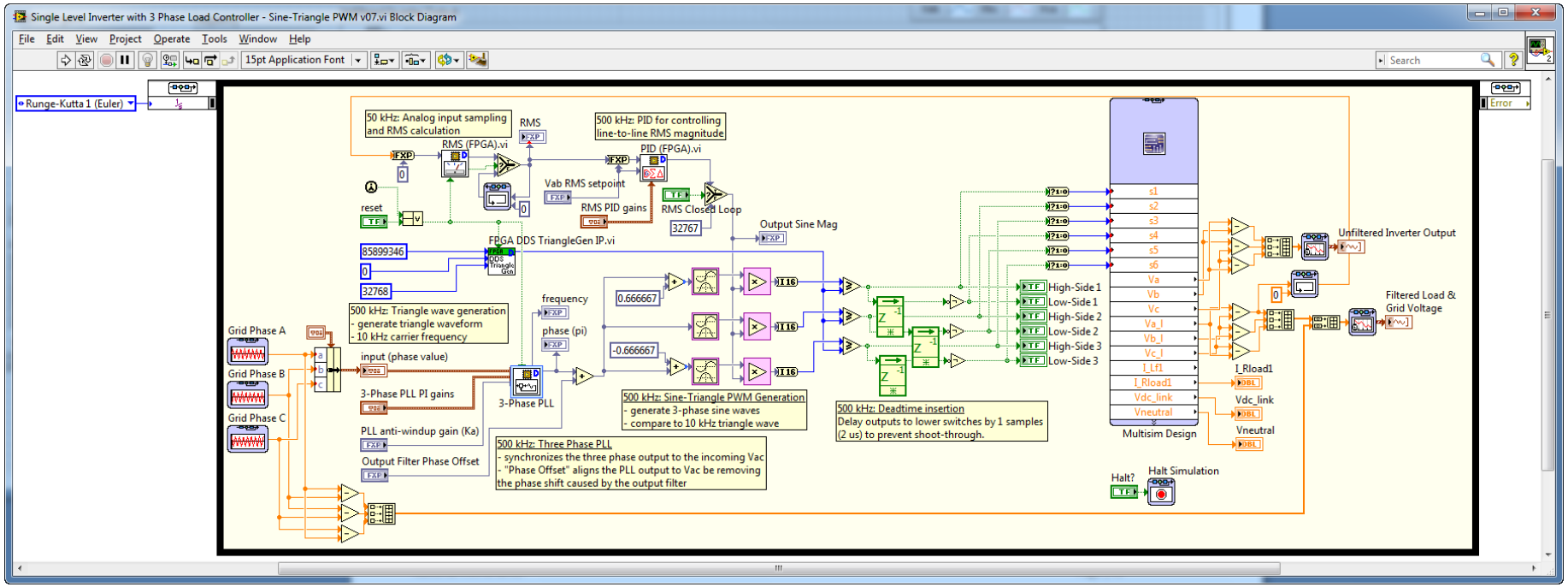
**Power Electronics Concepts**

- Fundamentals of single level three phase inverter circuits, a common DC to AC inverter topology used a range of applications such as solar inverters and variable-frequency motor/generator applications (learn more)
- Sine-triangle pulse width modulation (PWM) generation (learn more)
- 3-phase phase lock loop (PLL) algorithms for synchronizing generated power with the grid (learn more)
- How to compensate for the phase delay caused by the inverter output filter inductors and capacitors
    - Can you calculate the phase delay using Multisim AC analysis?
- Dead-time insertion for preventing shoot through currents that could potentially damage power transistor switches if the upper and lower switches are closed simultaneously

**Required Toolkits or Modules**

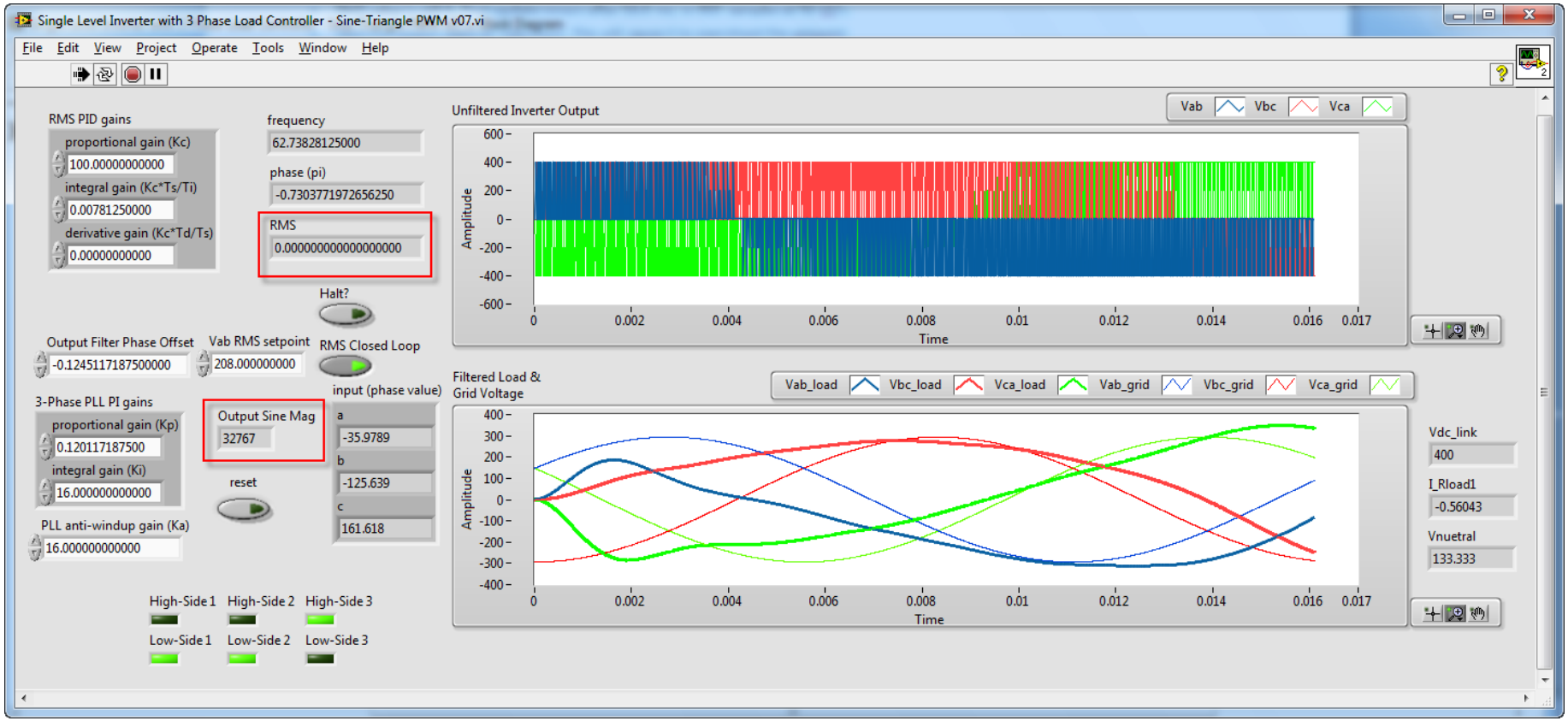
- NI LabVIEW 2011 (ni.com/labview)
- NI LabVIEW FPGA Module 2011 (ni.com/fpga)
- NI LabVIEW Control Design & Simulation Module 2011 (help files)
- NI Circuit Design Suite 12.0 Beta 0 (or higher) (ni.com/multisim)
- NI CompactRIO Waveform Reference Library (download here)


Vdc_link
Vdc_link
s1
S2
s2
S4
s3
S6
I_Lf1
XCP1
I_Rload1
XCP2
Vneutral
TRANSISTOR_DIODE
Lf1
100mH
Cf1
1µF
Lload1
1µH
Rload1
250Ω
Va_I
V1
400 V
Lf2
100mH
Cf2
1µF
Cf3
1µF
Lload2
1µH
Rload2
250Ω
Vb_I
Va
Vb
Vc
Lf3
100mH
Lload3
1µH
Rload3
250Ω
Vc_I
s4
S3
s5
S5
s6
S7
Single Level Inverter
LC Output Filter
3 Phase Load
Basic Single Level Inverter with 3 Phase Load


Single Level Inverter with 3 Phase Load Controller - Sine-Triangle PWM v07.vi Block Diagram
File Edit View Project Operate Tools Window Help
15pt Application Font
Search
Runge-Kutta 1 (Euler)
Error
50 kHz: Analog input sampling and RMS calculation
RMS (FPGA).vi
RMS
500 kHz: PID for controlling line-to-line RMS magnitude
PID (FPGA).vi
Vab RMS setpoint
RMS PID gains
RMS Closed Loop
reset
32767
Output Sine Mag
FPGA DDS TriangleGen IP.vi
85899346
0
32768
500 kHz: Triangle wave generation
- generate triangle waveform
- 10 kHz carrier frequency
frequency
phase (pi)
0.666667
-0.666667
Grid Phase A
Grid Phase B
Grid Phase C
input (phase value)
3-Phase PLL PI gains
3-Phase PLL
PLL anti-windup gain (Ka)
Output Filter Phase Offset
500 kHz: Three Phase PLL
- synchronizes the three phase output to the incoming Vac
- "Phase Offset" aligns the PLL output to Vac be removing the phase shift caused by the output filter
500 kHz: Sine-Triangle PWM Generation
- generate 3-phase sine waves
- compare to 10 kHz triangle wave
500 kHz: Deadtime insertion
Delay outputs to lower switches by 1 samples (2 us) to prevent shoot-through.
High-Side 1
Low-Side 1
High-Side 2
Low-Side 2
High-Side 3
Low-Side 3
s1
s2
s3
s4
s5
s6
Va
Vb
Vc
Va_I
Vb_I
Vc_I
I_Lf1
I_Rload1
Vdc_link
Vneutral
Multisim Design
Unfiltered Inverter Output
Filtered Load & Grid Voltage
I_Rload1
Vdc_link
Vneutral
Halt?
Halt Simulation

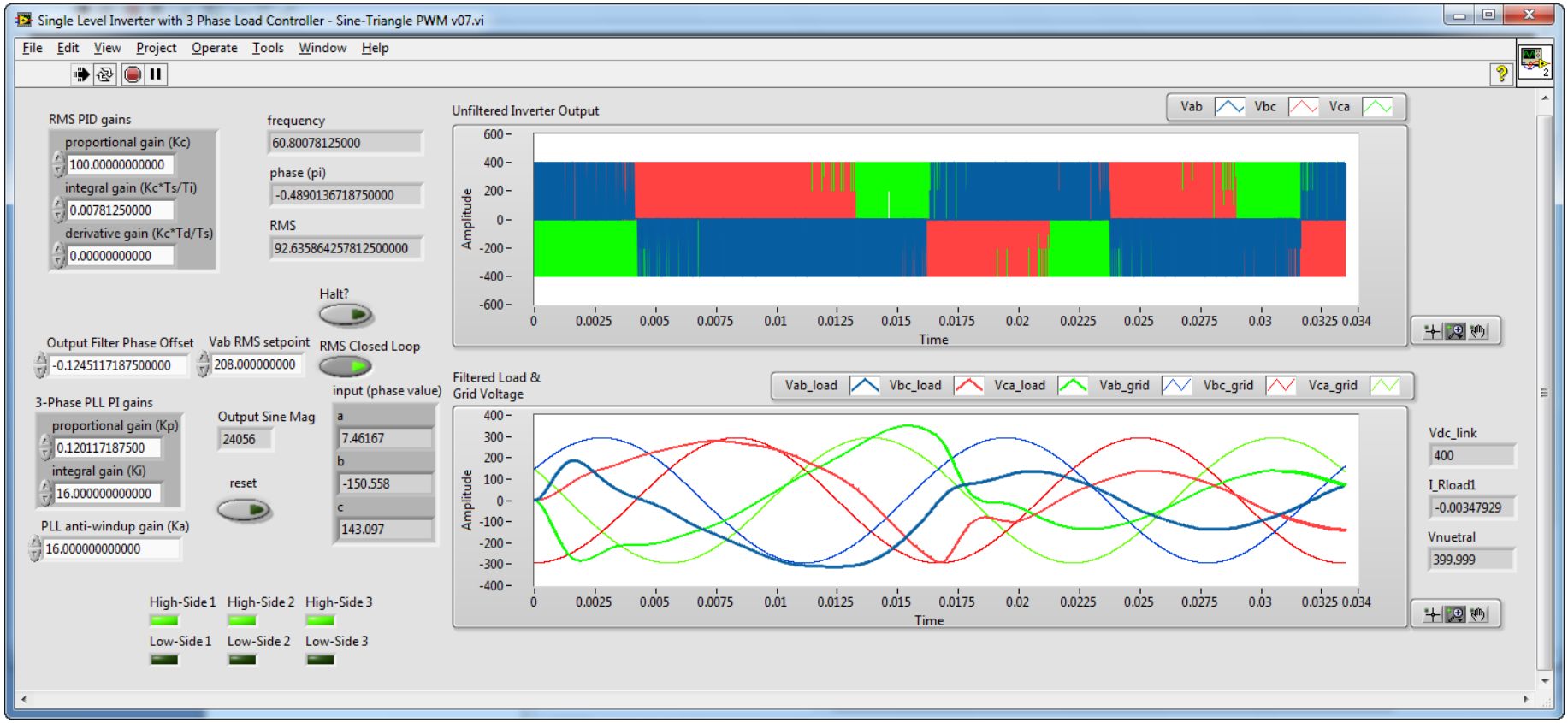
Startup behavior- first cycle:

- RMS value is still 0. First update occurs after 16.6 ms, or 833 samples at 50 kS/s.
- RMS PID output railed at maximum. This will cause it to overshoot the setpoint.



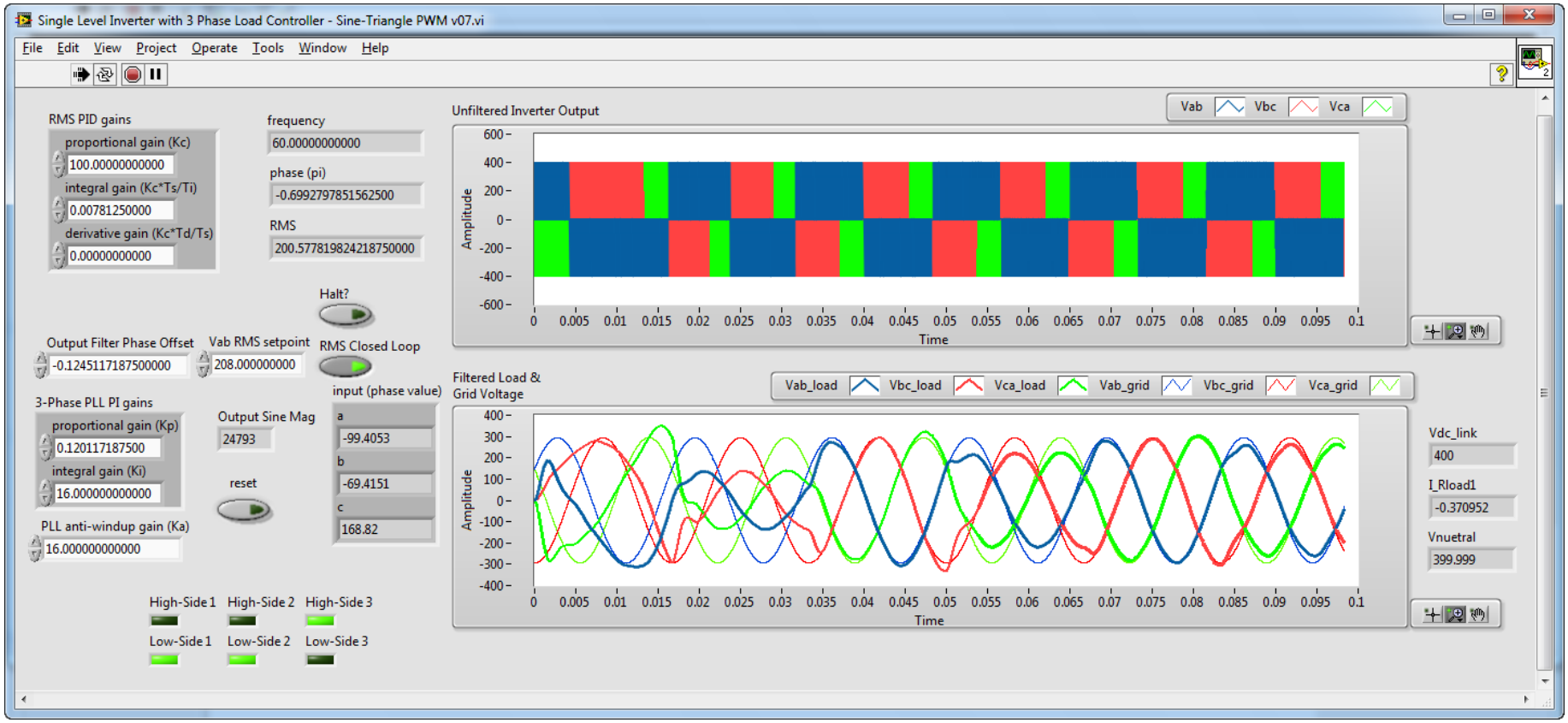
Startup behavior- second cycle:

- First RMS calculation is available.
- RMS PID output begins to drop.
- PLL frequency nearly stable. PLL frequency reaches the correct value of 60 Hz at around 40 ms.



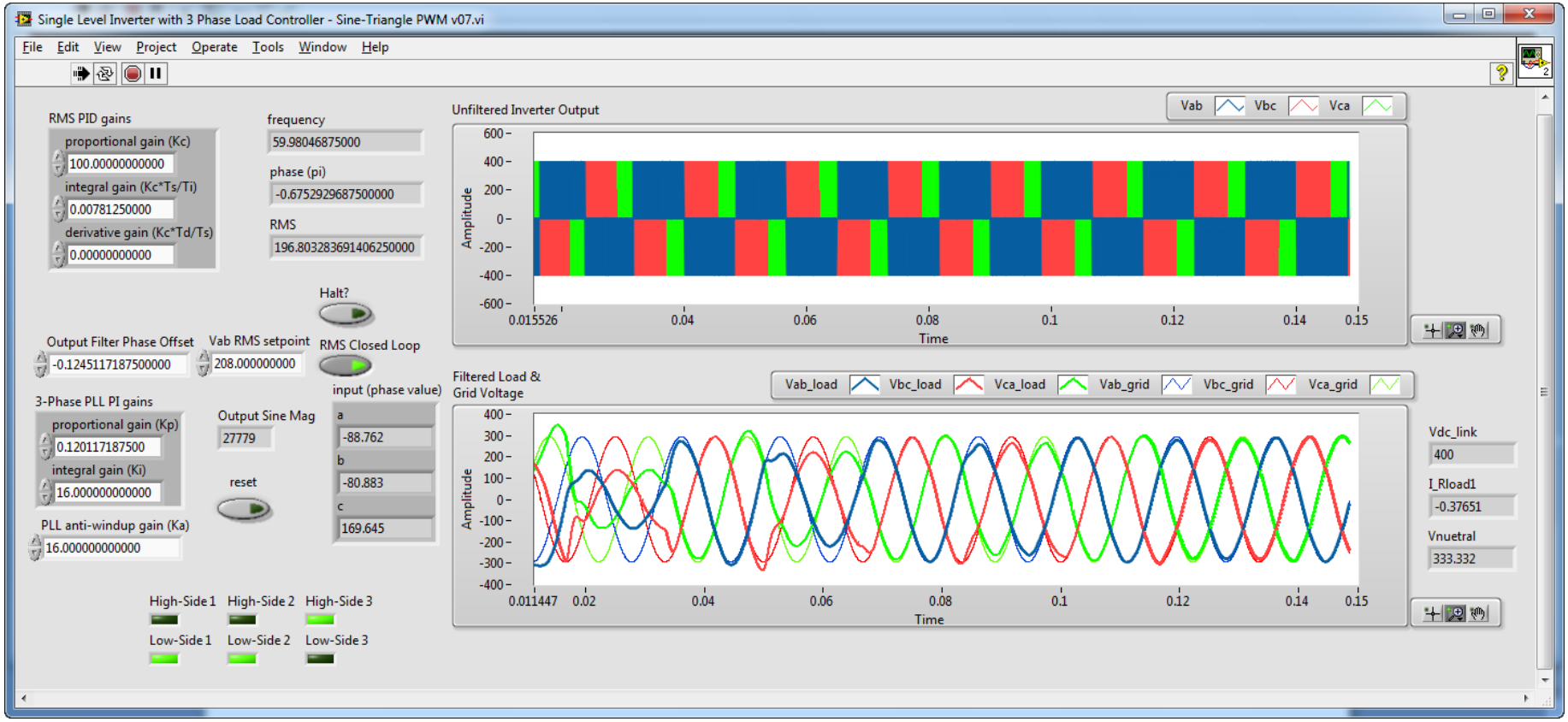
Startup behavior- sixth cycle:

- RMS PID stabilizing.
- PLL frequency stable.
- Sinusoidal waveform distortion decreasing.



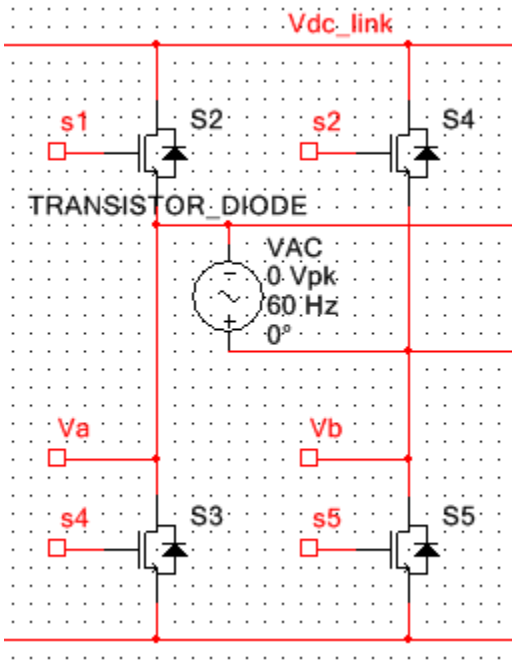
Steady state behavior- eighth cycle:

- System reaches steady state.



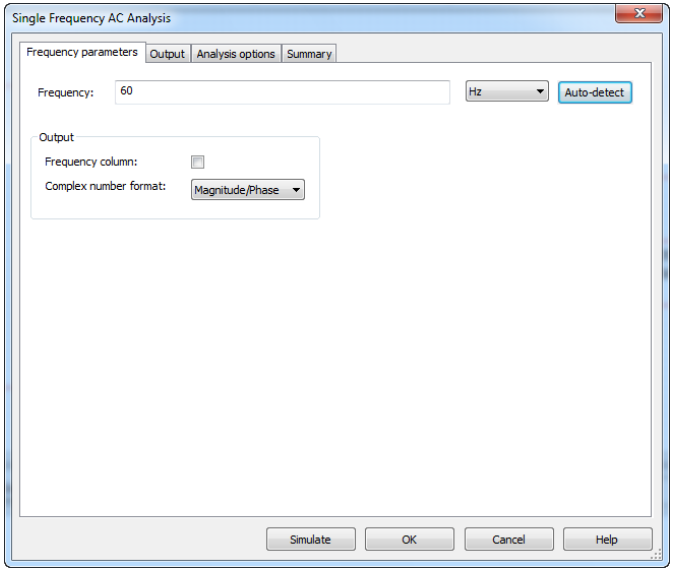
**Suggested Exercises**

- Add LabVIEW code to calculate and graph the auto power spectrum of the load power waveforms. How does changing the output filter design effect the magnitude of the harmonic energy in the frequency spectrum? How would implementing a different PWM generation schemes affect the harmonic energy generated by the inverter?
- The filtered inverter output waveform has a phase offset compared to the grid voltage, due to the output filter inductance and capacitance. Because the 3-Phase PLL uses the three-phase grid voltage as its input, it produces a phase signal that would be aligned with the grid voltage if there were no switching and filter delays in the inverter. This phase delay can be subtracted in the LabVIEW control application by adjusting the **Phase Offset** coefficient.
    - Can you use Multisim to perform **Single frequency AC analysis** to calculate the phase shift caused by the inverter output filter?
        - Hints:
            - In Multisim, place a sinusoidal 60 Hz voltage source near the inverter switches between the Vb and Va lines as shown. The negative terminal is connected to Va.

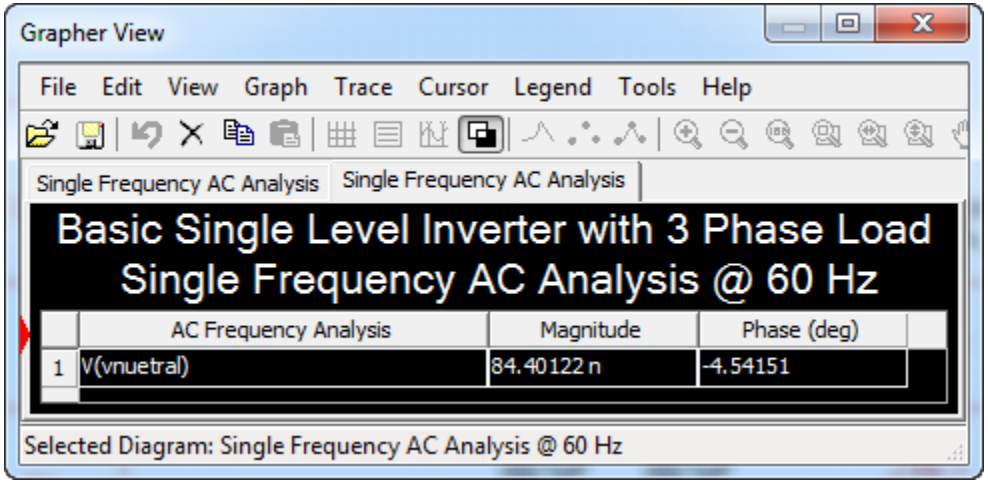



            - Note: You will need to disable this voltage source before running the co-simulation again. Otherwise Vab will be driven by the sinusoidal source, overriding your switching commands. To disable the component without deleting it from the circuit, follow the instructions in the section below for creating a circuit variant.

- Navigate to **Simulate>Analyses>Single frequency AC analysis**. Set the Frequency to 60 Hz, configure the analysis for **Magnitude/Phase** mode to calculate the phase shift in degrees and navigate to the **Output** tab to add an output for **V(vneutral).**



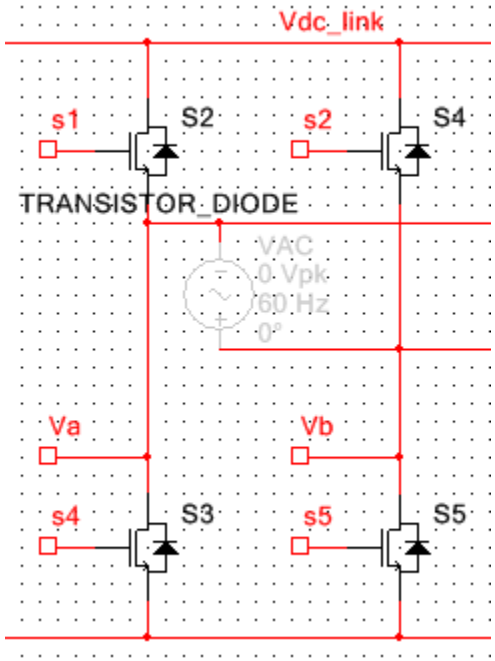

- The analysis results are shown:
  - Since the sinusoidal source was wired in reverse, the actual phase shift is +4.54151 degrees. Since there are two output filters in the analysis path, you can divide that by two to calculate the actual phase shift between the transistor diode outputs and the loads, or +2.2708 degrees.



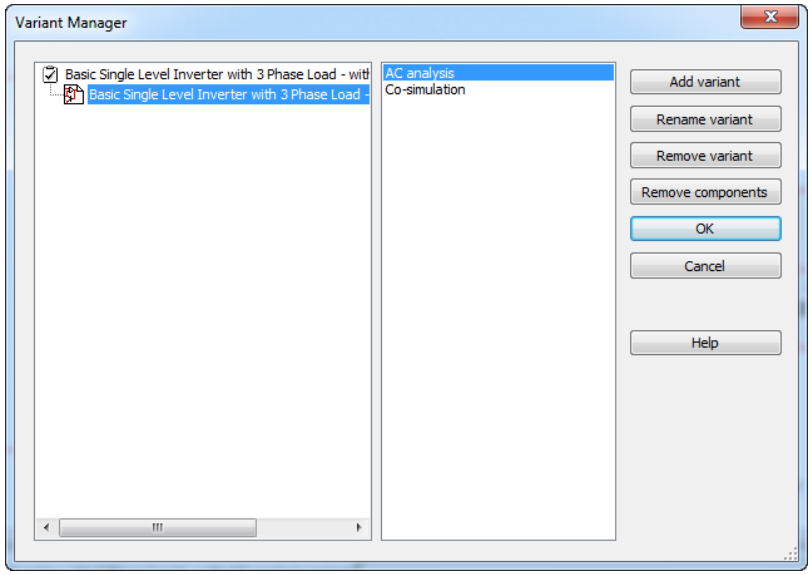
- In LabVIEW FPGA, you enter the **Output Filter Phase Offset** as a fixed point number in units of pi*radians. The phase offset for LabVIEW FPGA is therefore calculated as follows:
    - Phase Offset = (-4.54149 deg)/2 *pi/180*pi = -0.12451.

**Creating Circuit Variants**

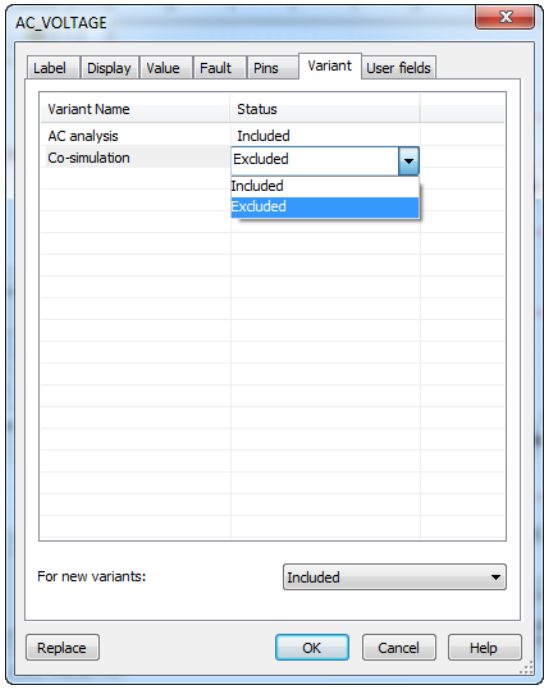
- You can create a circuit variant in Multisim to create a different version of the schematic for tasks such as AC analysis and easily switch between variants. For example, you can disable the AC voltage source when the circuit is in the variant you create for co-simulation.
- When the component is disabled in a circuit variant, it appears grayed out as shown below:



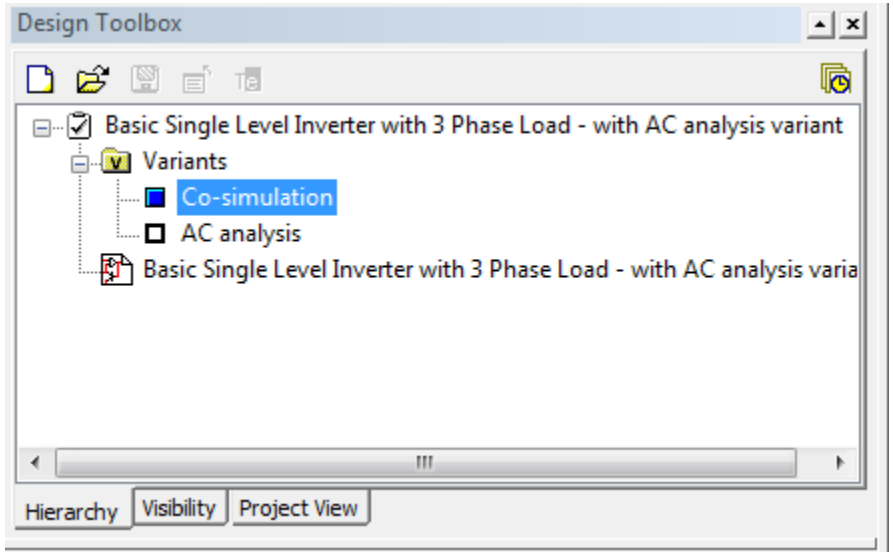

- To do this, navigate to **Tools>Variant Manager**. Then rename the default variant “Co-simulation”. Next add a new variant and name it “AC analysis”, as shown below.




- In the circuit, double click the **AC_VOLTAGE** component to configure its properties. Then navigate to the Variant tab and set the component to be excluded in the **Co-simulation** variant.



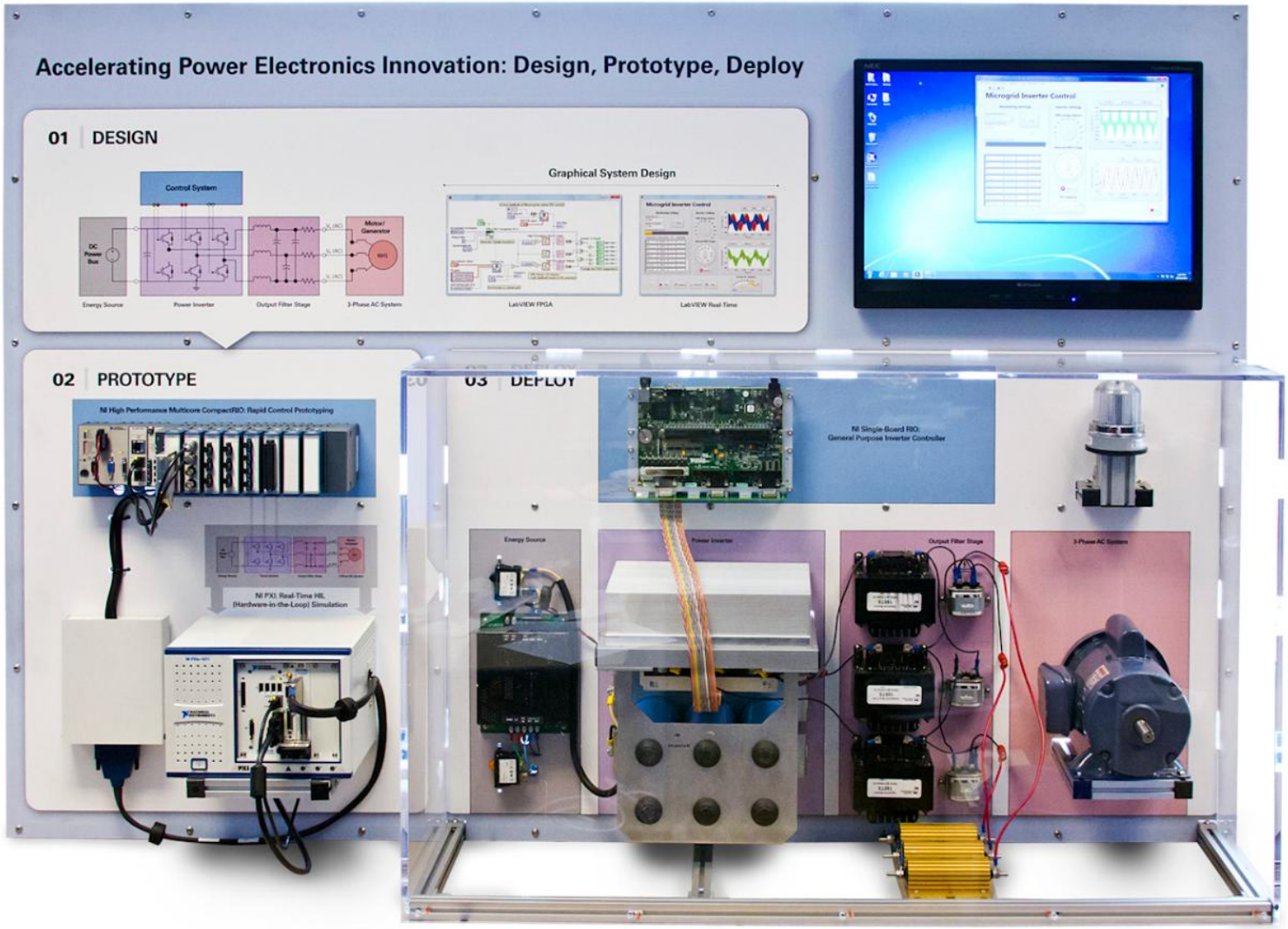
- Finally, double-click to select the active circuit variant in the Design Toolbox window to the left of the schematic. You can now quickly disable the **AC_VOLTAGE** component when executing co-simulation runs.



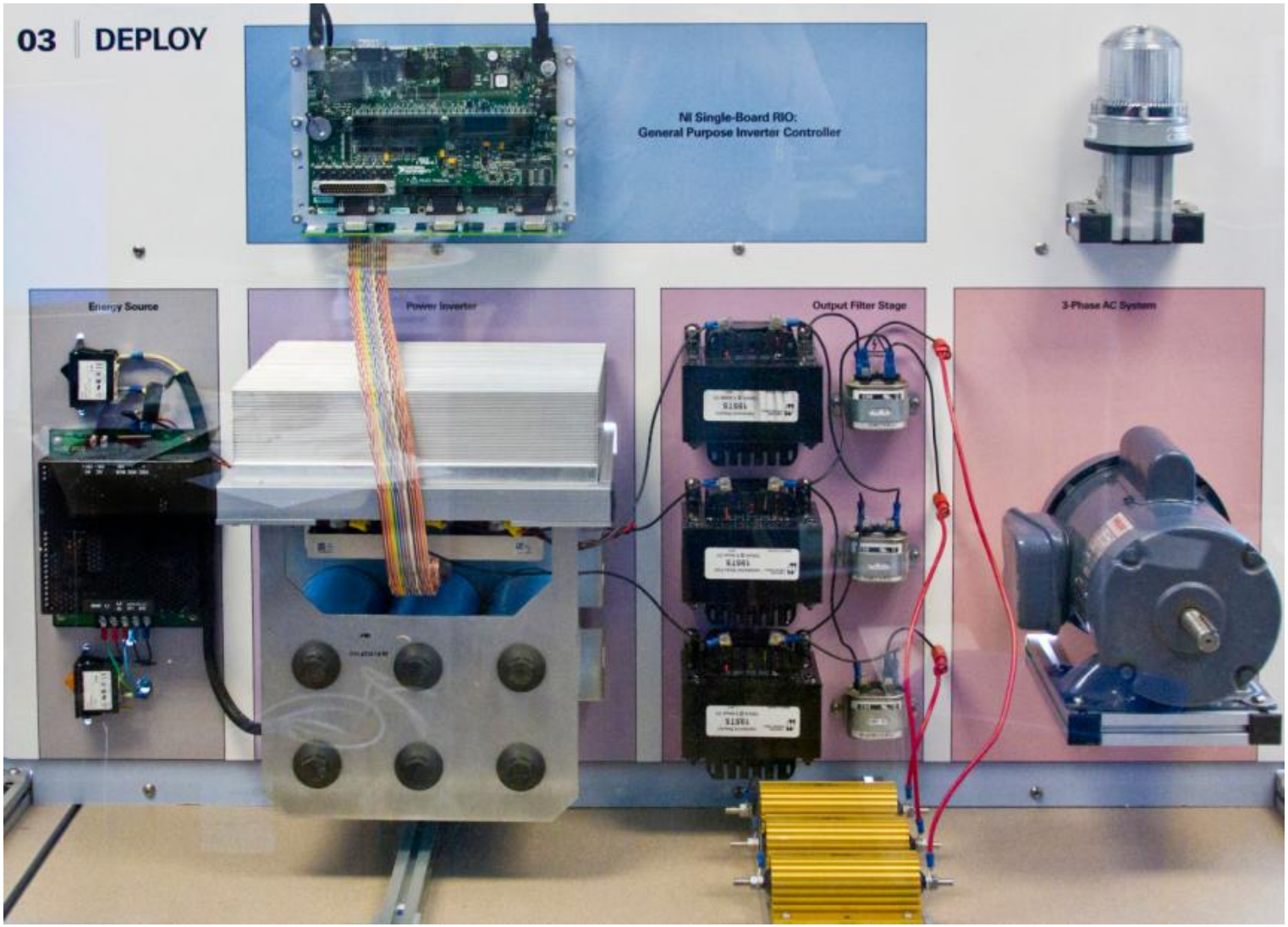

- Note: You can also use this feature to include waveform generators components to control the circuit when performing transient analysis independent of LabVIEW.

**Inverter Control Deployment Project**

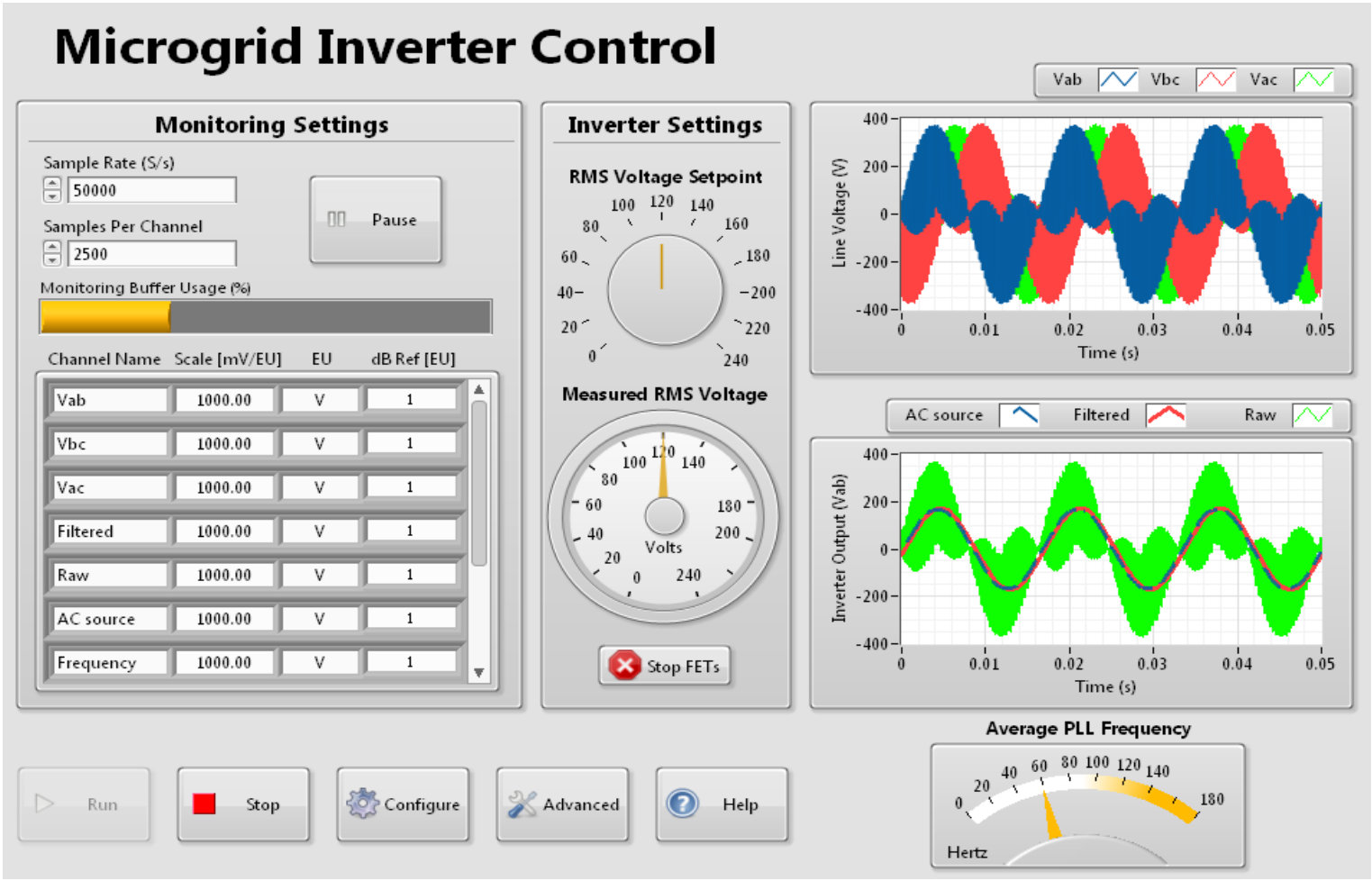
- The photos below show the inverter control proof of concept system that demonstrates: 1. Desktop co-simulation with NI Multisim and LabVIEW FPGA for control system development and debugging (top), 2. Rapid control prototyping with NI High Performance Multicore CompactRIO and real-time hardware-in-the-loop (HIL) simulation of inverters and motor drives with NI PXI and FlexRIO FPGAs, and 3. Commercial deployment with NI Single-Board RIO.



- The photo below show details of the physical inverter control system, controlled by NI Single-Board RIO (green PCB with multicolor ribbon cable). The inverter phase-locks to single-phase 120 VAC power from the grid. In this case the energy source is a 300 VDC power supply that connects to the large 3900 uF DC link capacitors (blue) of the SEMIKRON SKiiP®3 six-pack inverter (rated for 360 kW, 1200 V, 300 A) with the SKiiP®3 Intelligent Power Module (IPM) gate driver circuit (white with multicolor ribbon cable). The 3-leg single level inverter is connected to 100 mH output filter inductors (black), 1 uF capacitors (silver), and 250 Ohm power resistors (gold). The 3-phase motor/generator (grey) is not connected.



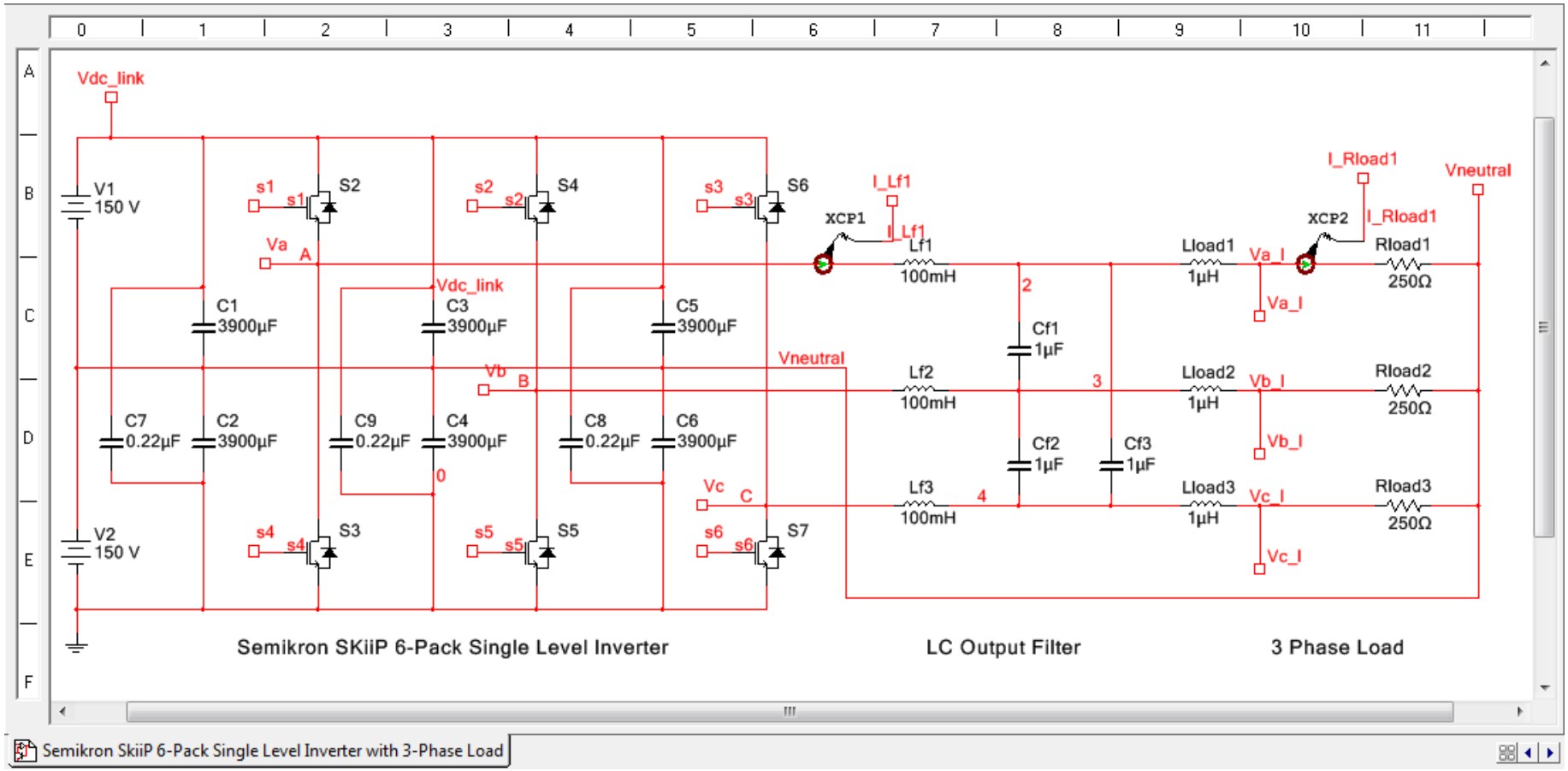
- Front panel of the LabVIEW Real-Time application while the physical inverter is running, showing unfiltered PWM outputs and filtered power signals.



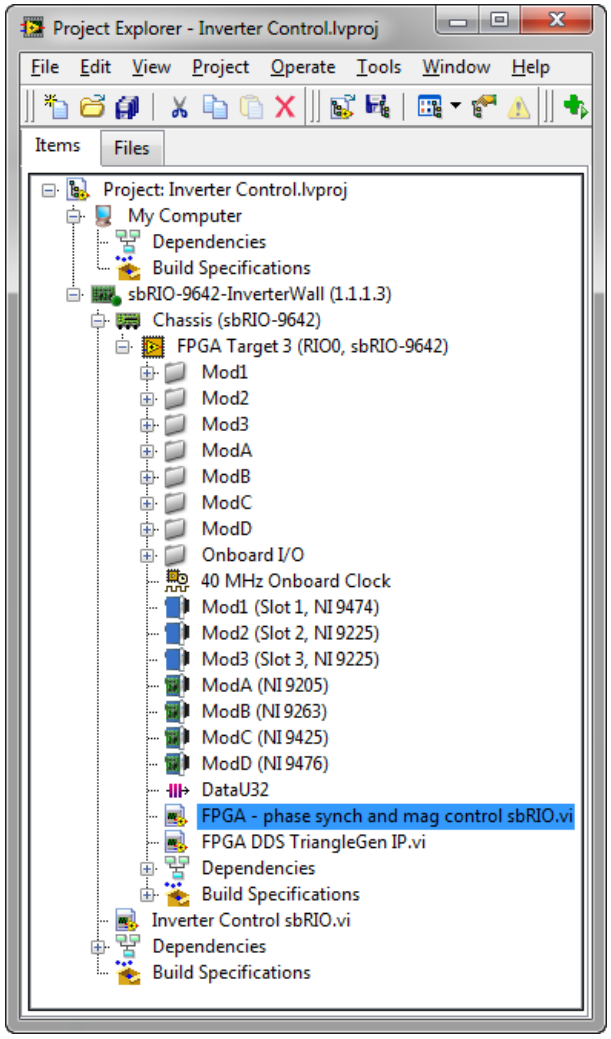

Note:

- In this screenshot, the Vac waveform is displayed rather than Vca. That is why the three phase power signals appear as if they are not correctly 120 degrees out of phase. In fact, if the line-to-line voltage, Vca, was displayed correctly they would appear as expected.

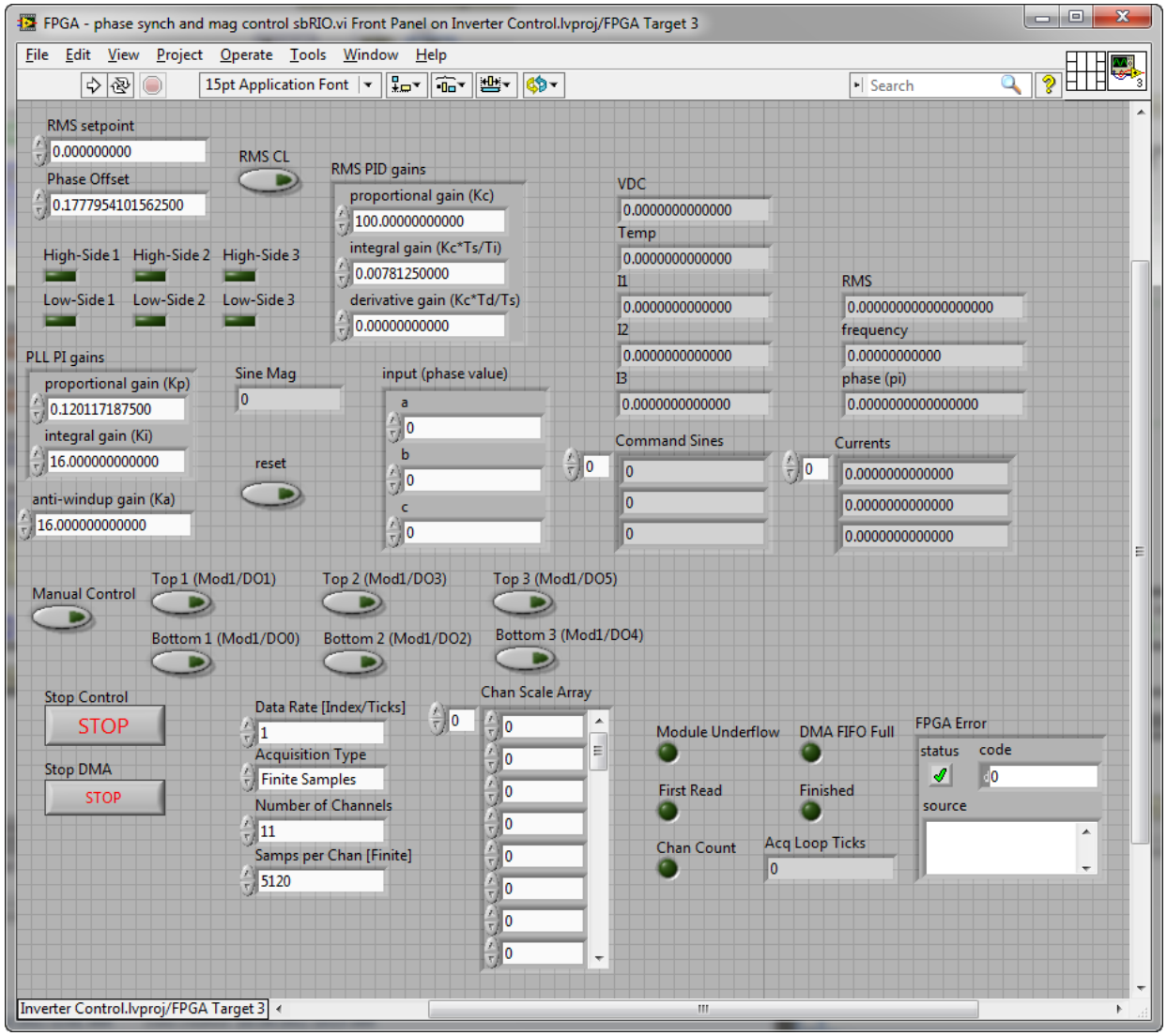
- Multsim schematic
    - The Multisim schematic below, located in the **Solutions** subfolder, is a close match to the physical system. Notice that the 3900 uF DC-link capacitors are tied to the neutral voltage point.



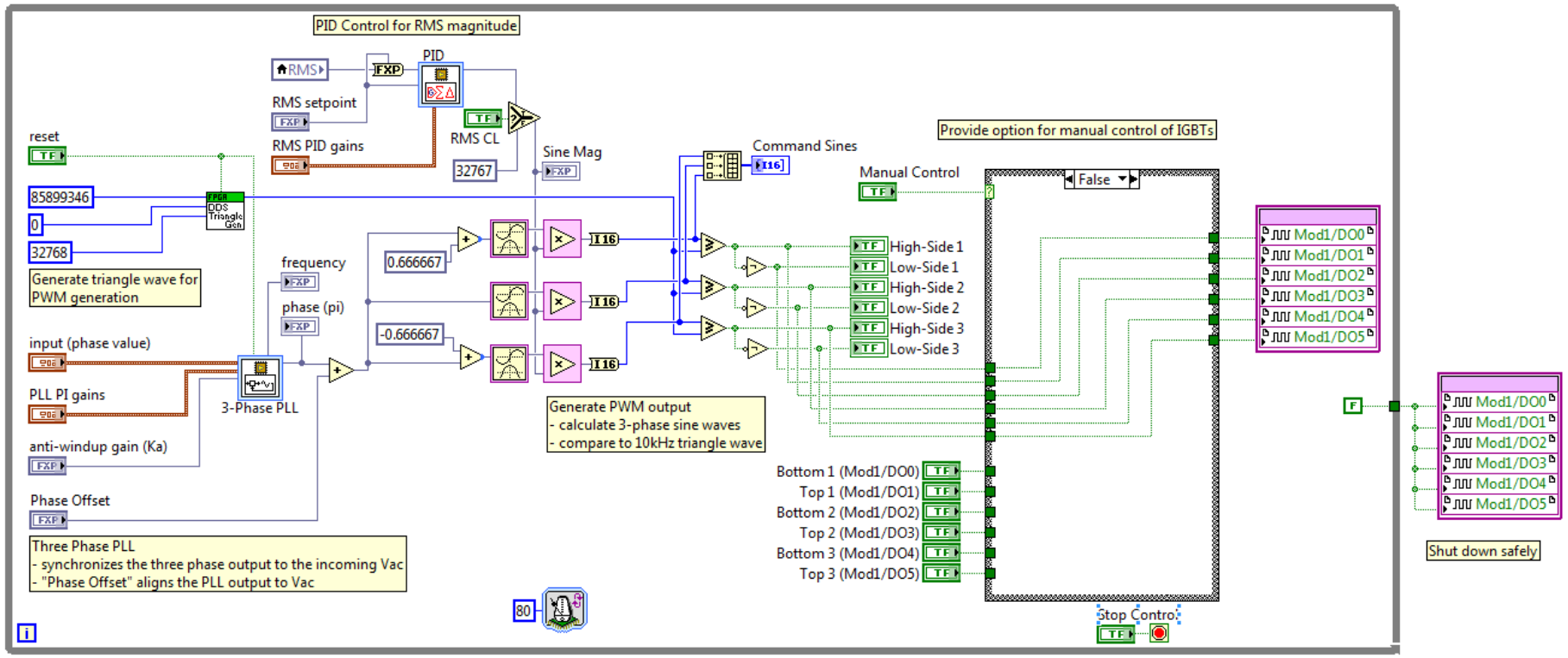
- LabVIEW Project Code
  - Note: Install the NI CompactRIO Waveform Reference Library (download here) before opening this project. This library facilitates high speed DMA data streaming between the FPGA and real-time processor.
  - Navigate to the **SB-RIO Deployment Code** subfolder and open the LabVIEW Project file, **Inverter Control.lvproj**.
  - Explore the LabVIEW Project used for the physical NI Single-Board RIO inverter control application. Expand the project tree to see the LabVIEW FPGA and LabVIEW Real-Time LabVIEW applications.
  - Double click to open the LabVIEW FPGA application, **FPGA - phase synch and mag control sbRIO.vi**.



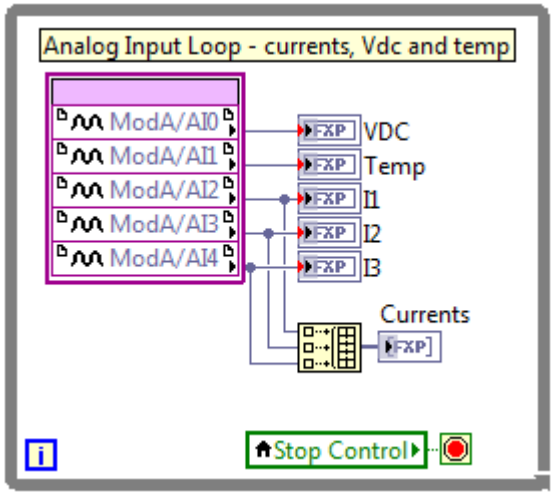
- The LabVIEW FPGA front panel is used for point-by-point communication with the FPGA application when running, for debugging, monitoring and control tuning purposes. The front panel controls and indicators are used for single point interactive mode operation (at typical speeds less than 10 Hz), and also act as a single-point, bidirectional communication interface with the real-time processor (at typical speeds less than 5 kHz).
    - Note: High speed data streaming between the FPGA and real-time host processor is performed using DMA data transfer, using the NI CompactRIO Waveform Reference Library (learn more).



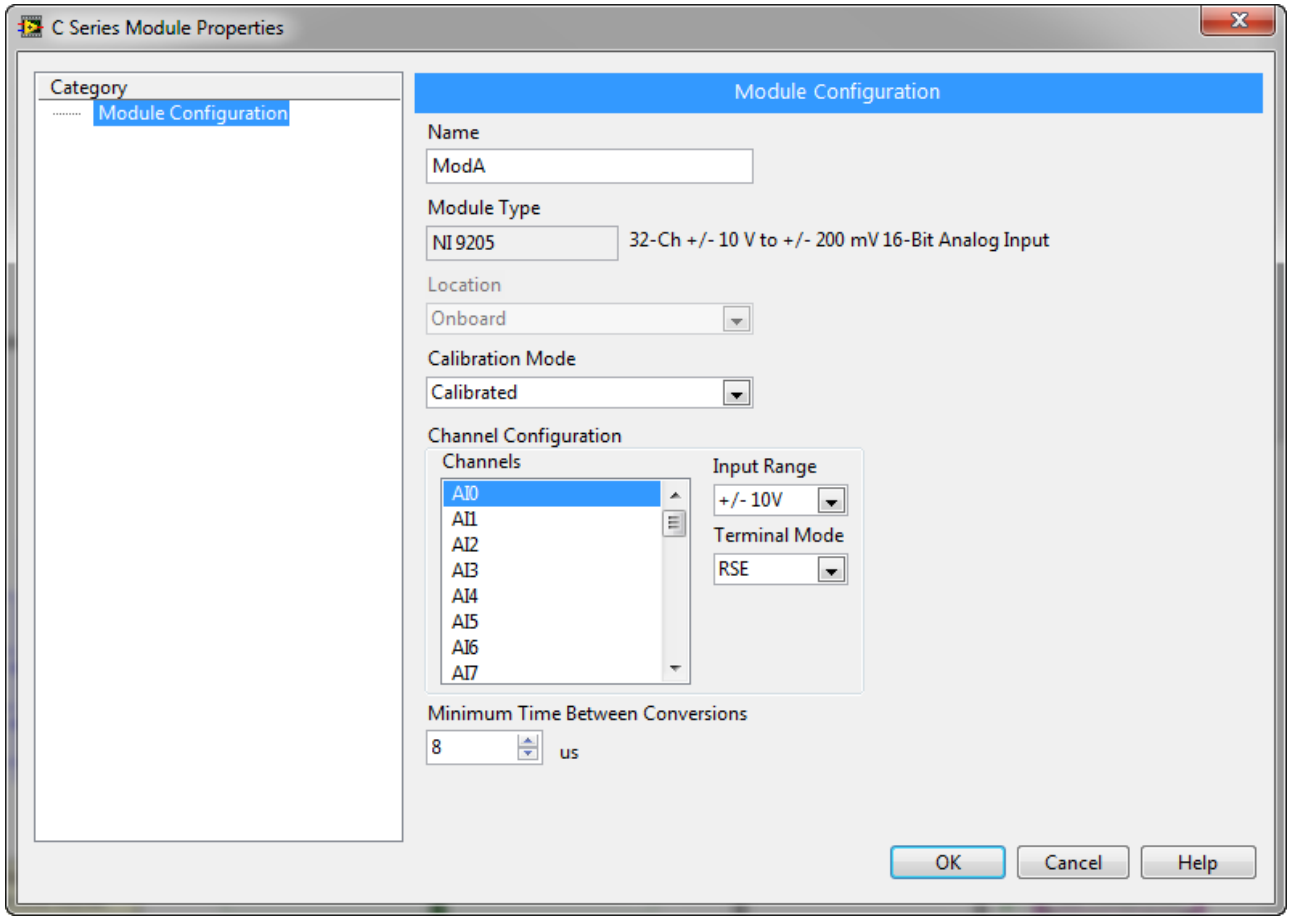
- Navigate to the block diagram (Ctrl+E), and explore the multiple parallel control loops. A big advantage of FPGA-based control systems is the true hardware parallelism of the code. If you add an additional control or DSP loop, it does not affect the execution timing of the existing code. Furthermore, there are no penalties for running control loops at high speed since they are not competing for resources (learn more).
- Explore the PID control loop with 3-Phase PLL and Sine-Triangle PWM generation. Note that the loop timer function causes the loop to execute every 80 ticks of the 40 MHz FPGA clock, which is a 500 kHz control loop rate.
    - Caution: In this case, dead-time insertion is unnecessary since the Semikron SKiiP®3 Intelligent Power Module (IPM) gate driver circuit provides the required deadtime insertion. If your gate-drive circuitry does not include deadtime insertion, you must do this in the FPGA application see the **Brushless DC Motor with Hall Sensors (Trapezoidal Flux)** section of this document for an example and rising edge deadtime insertion IP core.



- A separate loop performs analog input for the inverter currents, DC link voltage and inverter temperature using an NI 9205 analog input module. In this case, these conditioned signals are provide by the Semikron Intelligent Power Module (IPM). Because no loop timer is included, the data acquisition function is executed at the maximum sampling rate of the NI 9205 module.

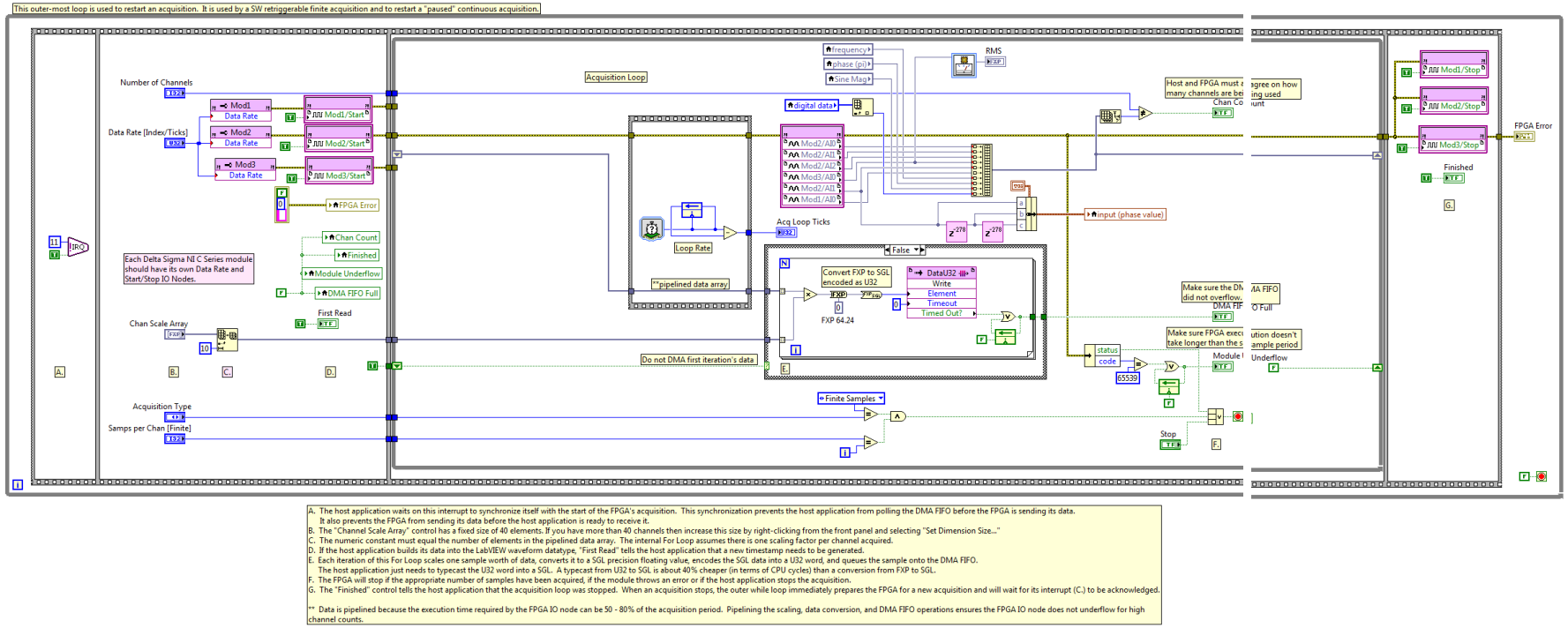


- Because no loop timer is included, the data acquisition function is executed at the sampling rate of the NI 9205 module. This scanned analog input module is configured for an 8 us conversion time per channel and 5 channels are being acquired, so the loop rate for this analog input loop is 40 us (25 kHz). To view this configuration, navigate to the LabVIEW Project, right-click on **Mod A (NI 9205)**, and select **Properties**. This module also has a per channel selectable input range (+/- 10, 5, 1, or 200 mV) and terminal mode (referenced single ended, non-referenced single ended, or differential).

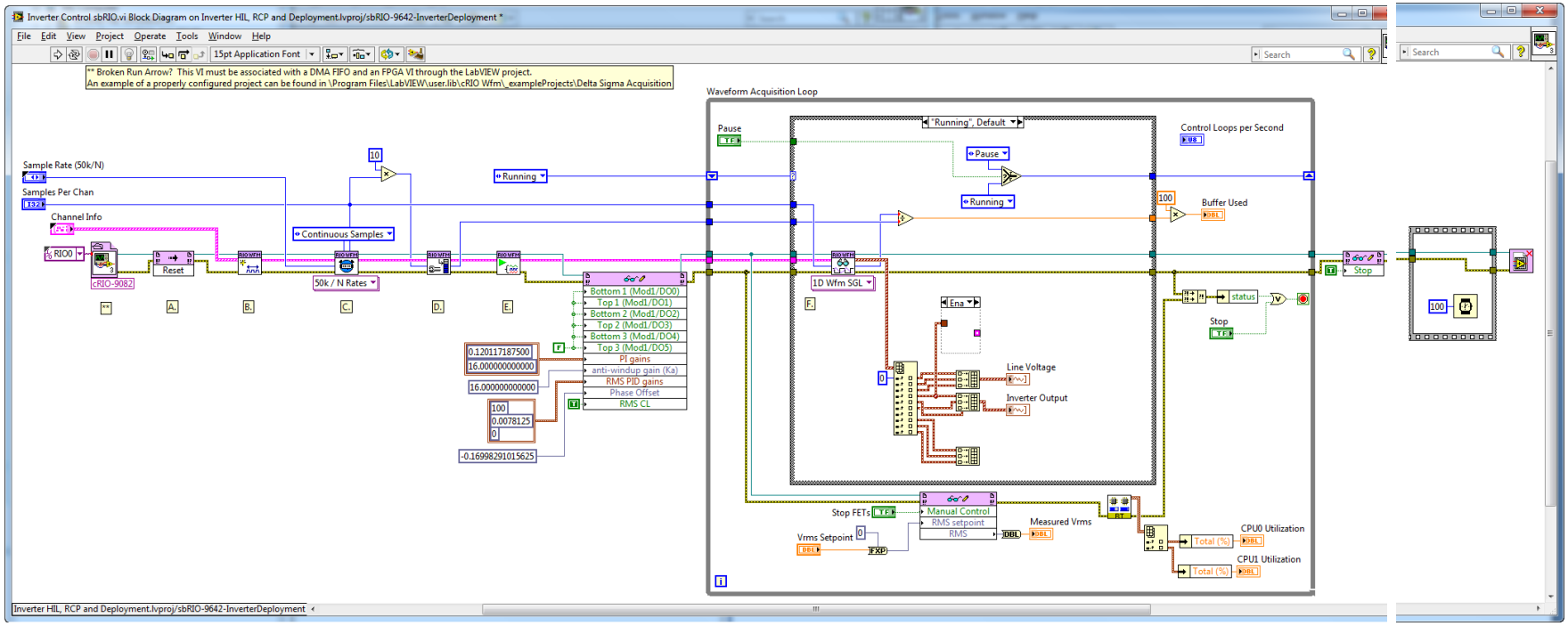



- Note: To avoid problems with inter-channel crosstalk, take care to properly wire, terminate and ground the channels. Per the user manual, you must always connect the COM pin of your NI 9205 to a ground reference (learn more).

- The DMA data transfer loop is used to pass high speed waveforms from the FPGA directly to the real-time processor memory, acting like a digital scope for your LabVIEW FPGA application. This is useful for debugging during development, performing power quality analysis on the real-time processor, or you can program the FPGA to trigger a waveform acquisition in the event of an abnormal operating condition of your inverter and capture the waveforms for later analysis.
    - Learn more about the [NI CompactRIO Waveform Reference Library]



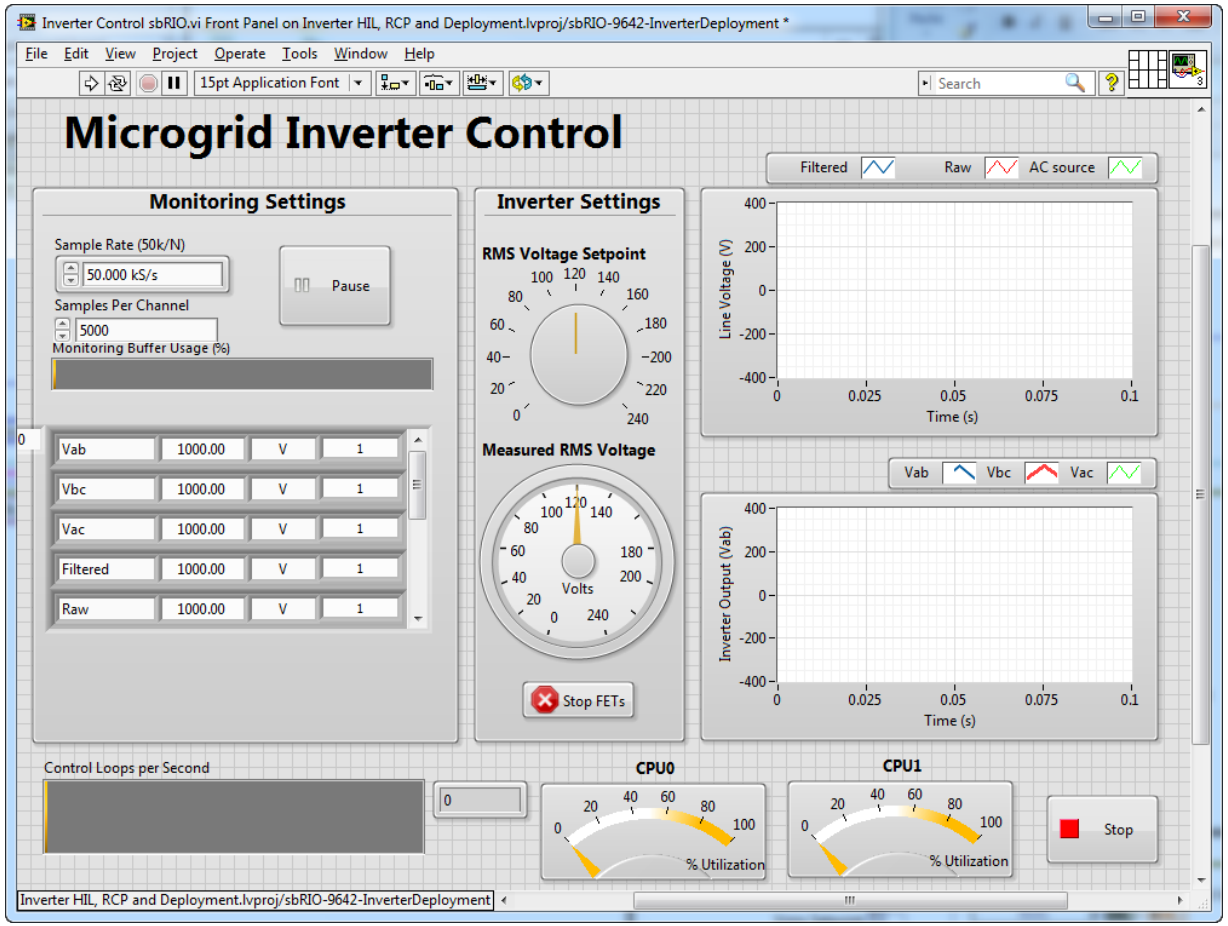
- In the LabVIEW Project, double-click to open the LabVIEW Real-Time application, **Inverter Control sbRIO.vi**. This application downloads and runs the LabVIEW FPGA application and can provide deterministic real-time supervisory control. In this case, the real-time control system is running the VxWorks real-time operating system.



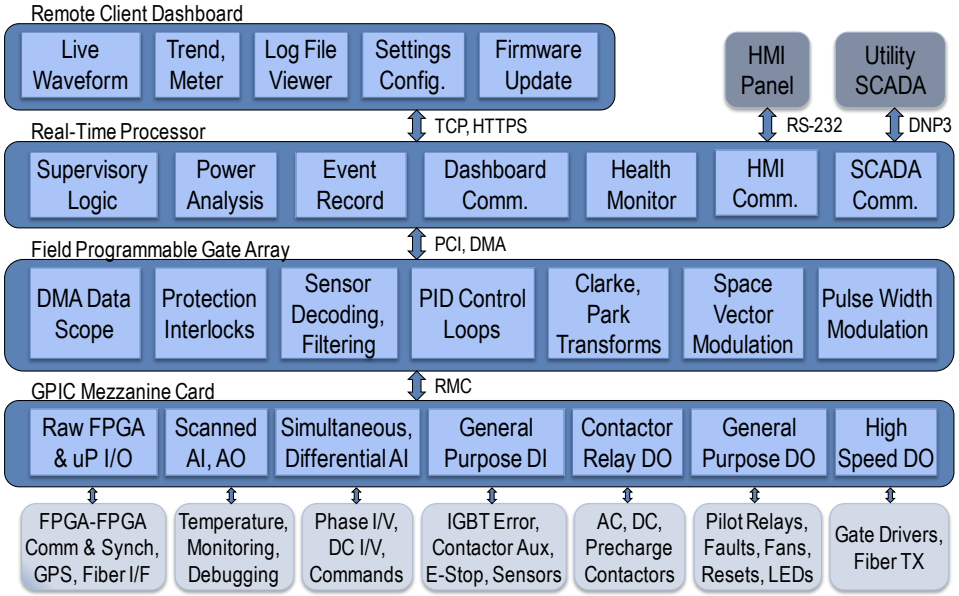

Note:

- To execute the LabVIEW Real-Time application, you would first need to compile the LabVIEW FPGA application. To do this, save the LabVIEW FPGA application and click the run arrow to compile it.

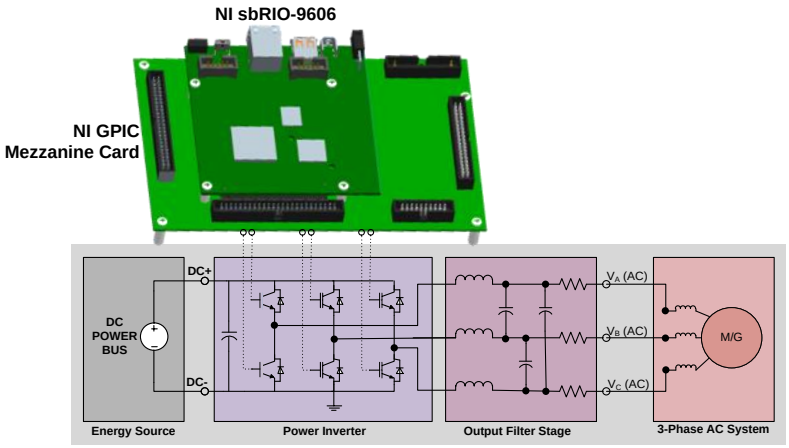
- Examine the LabVIEW Real-Time front panel.



- A complete LabVIEW based inverter control and power quality monitoring system typically contains some or all of the software elements shown below.




- National Instruments is also developing a general purpose inverter control I/O board for NI Single-Board RIO (shown below). This board is a deployment-ready commercial embedded system for high-volume grid-tied inverter and motor/generator drive applications, typically 50 kW and larger. To request a preliminary datasheet or access to the pioneer program please email Clean Energy Product Manager, Brian.MacCleery@ni.com.



## Brushed DC Motor H-Bridge (Basic)

**Co-Simulation Concepts**

- Developing, debugging and modifying LabVIEW FPGA IP Cores
    - It's recommended that you develop a test bench application for each power electronics IP core and use that to verify that each block behaves as expected. Whenever changes are made to your control code, run the test bench application again to confirm that the code is still working as expected.
        - You can automate your code verification process using tools like NI Veristand, NI Teststand, and the NI Unit Test Framework. These tools can automatically test your power electronics IP blocks, determine pass/fail, and generate reports.
        - For information on how to write your LabVIEW FPGA code so that it can be validated within a testbench or co-simulation application, see part three of the CompactRIO Developers Guide and navigate to the section titled "LabVIEW FPGA Development Best Practices."
- Using electromechanical simulation blocks and sensor feedback blocks in Multisim and using simulated sensor feedback to create a closed loop control system

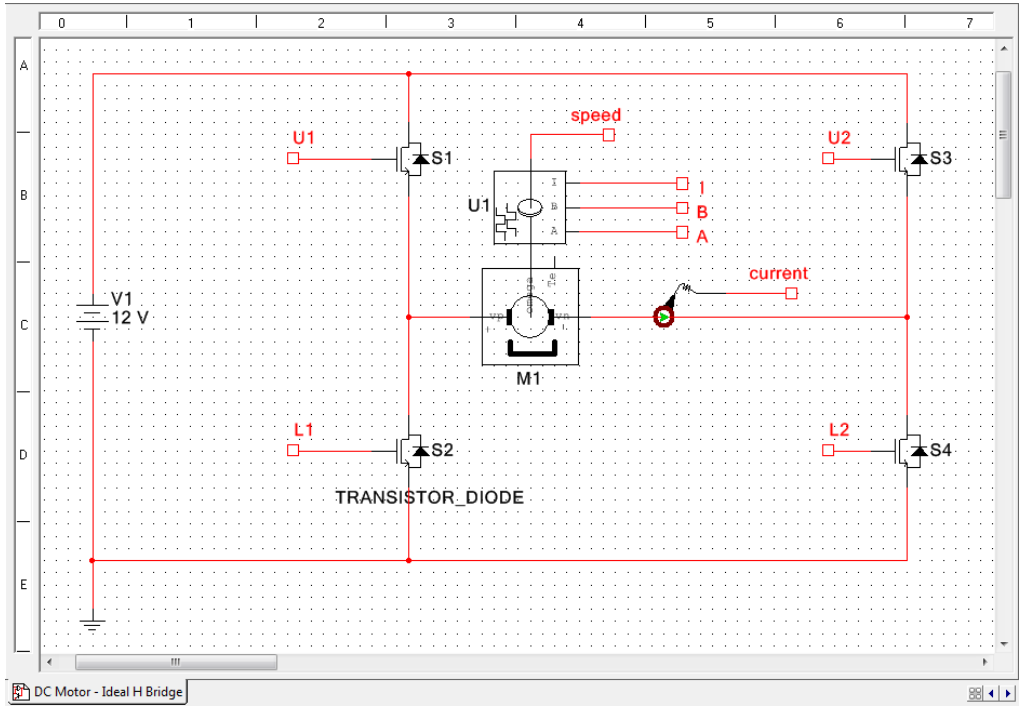
**Power Electronics Concepts**

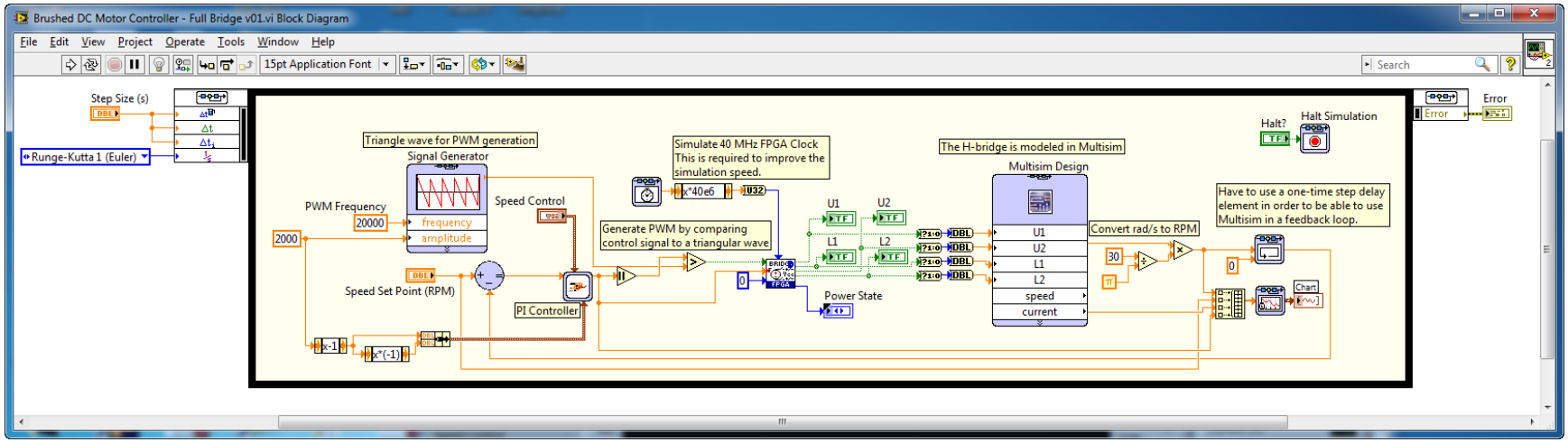
- Fundamentals of H-bridge motor control circuits, a common topology used in DC to DC converters and DC to AC inverters that enables you to connect a DC supply voltage to the load in either a positive or negative polarity (learn more)
    - H-bridges are also commonly used in single phase inverter circuits for applications such as grid-tied photovoltaic (PV) solar arrays. In this case, the H-bridge PWM output is modulated to produce a sine wave, which is filtered and then boosted for connection to the grid through an isolating transformer (learn more).
- Understanding basic pulse width modulation (PWM), a method of regulating the amount of current flowing to an inductive load such as a motor using power transistors such as IGBTs and MOSFETs (learn more)
- Understanding the difference between slow decay and fast decay (synchronous rectification) modes of controlling an H bridge circuit:
    - In slow decay mode, the motor current freewheels through the H bridge diodes when the controlled transistor switches are off. Therefore, current tends to decay slowly as it circulates through the bridge circuit and is dissipated in the motor and diode resistances.
    - In fast decay mode, also known as synchronous rectification, the lower two switches are turned on when braking the motor, causing both motor winding terminals to be connected to ground. By grounding the motor windings, the current decays very quickly to zero and motor braking occurs very quickly.

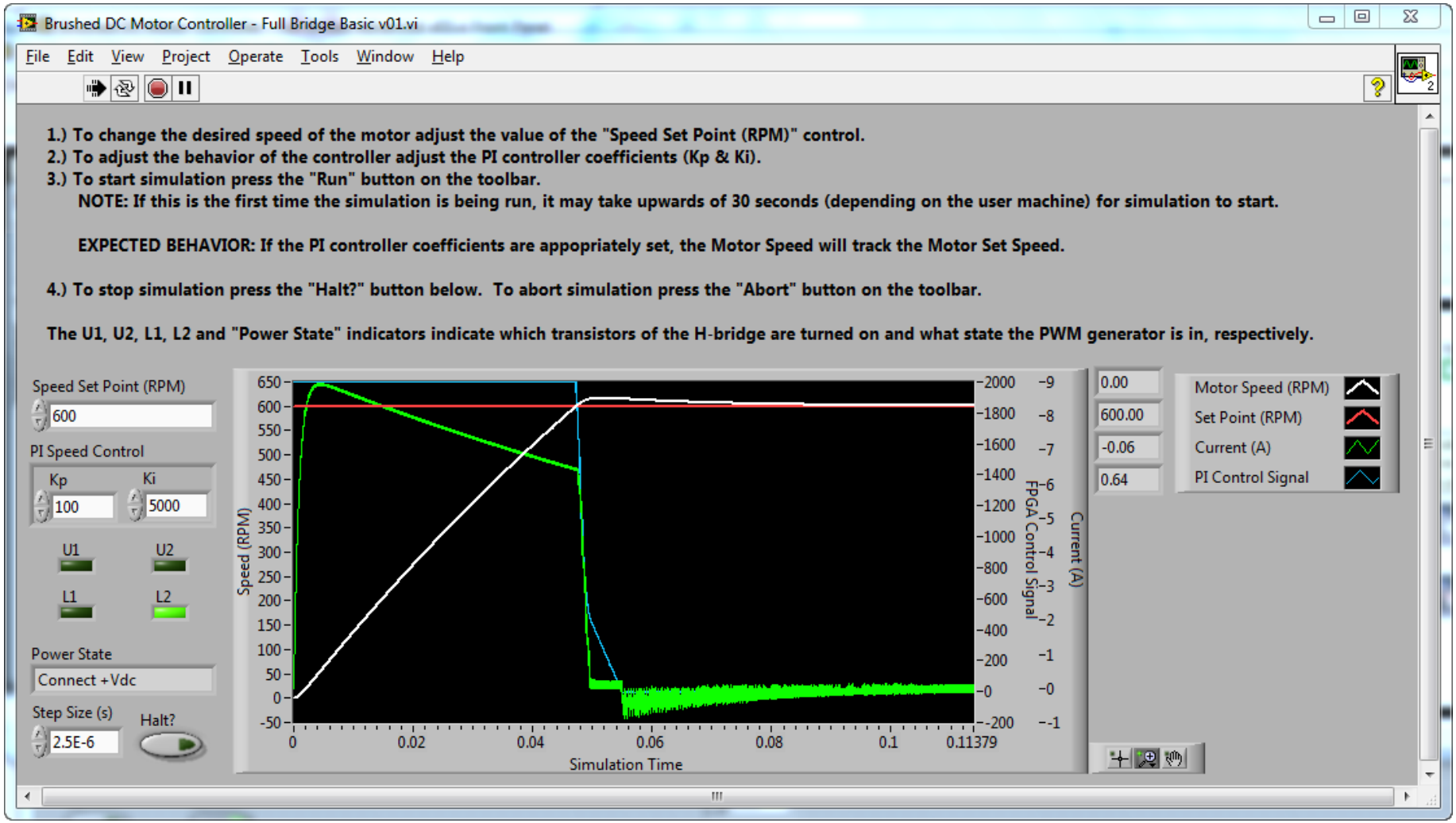
**Required Toolkits or Modules**

- NI LabVIEW 2011 (ni.com/labview)
- NI LabVIEW FPGA Module 2011 (ni.com/fpga)

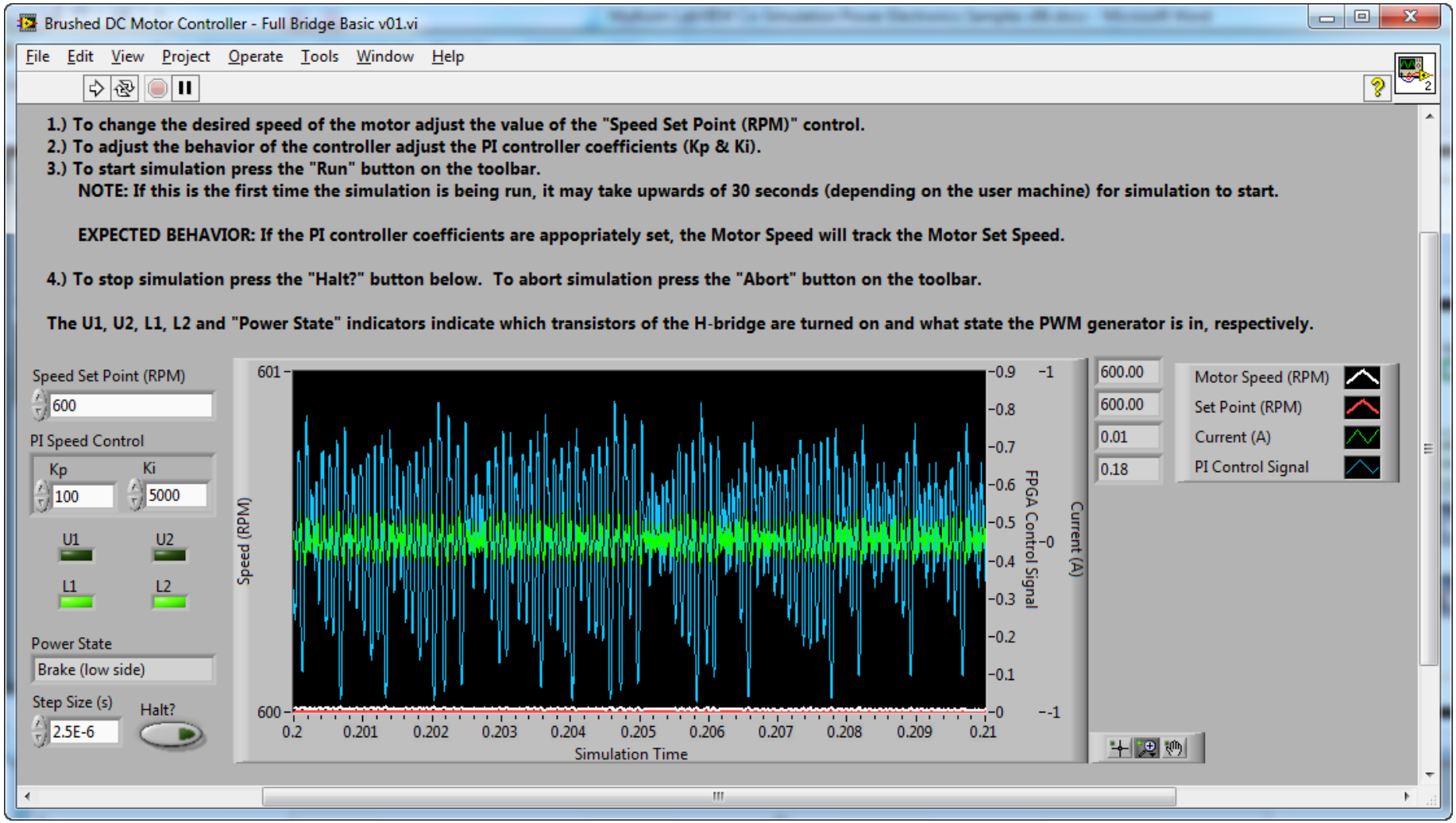
- NI LabVIEW Control Design & Simulation Module 2011 (help files)
- NI Circuit Design Suite 12.0 Beta 0 (or higher)(ni.com/multisim)




Brushed DC Motor Controller - Full Bridge Basic v01.vi
File Edit View Project Operate Tools Window Help
1.) To change the desired speed of the motor adjust the value of the "Speed Set Point (RPM)" control.
2.) To adjust the behavior of the controller adjust the PI controller coefficients (Kp & Ki).
3.) To start simulation press the "Run" button on the toolbar.
NOTE: If this is the first time the simulation is being run, it may take upwards of 30 seconds (depending on the user machine) for simulation to start.
EXPECTED BEHAVIOR: If the PI controller coefficients are appropriately set, the Motor Speed will track the Motor Set Speed.
4.) To stop simulation press the "Halt?" button below. To abort simulation press the "Abort" button on the toolbar.
The U1, U2, L1, L2 and "Power State" indicators indicate which transistors of the H-bridge are turned on and what state the PWM generator is in, respectively.
Speed Set Point (RPM)
600
PI Speed Control
Kp
100
Ki
5000
U1
U2
L1
L2
Power State
Connect +Vdc
Step Size (s)
2.5E-6
Halt?
Speed (RPM)
FPGA Control Signal
Current (A)
Simulation Time
0
0.02
0.04
0.06
0.08
0.1
0.11379
0.00
600.00
-0.06
0.64
Motor Speed (RPM)
Set Point (RPM)
Current (A)
PI Control Signal

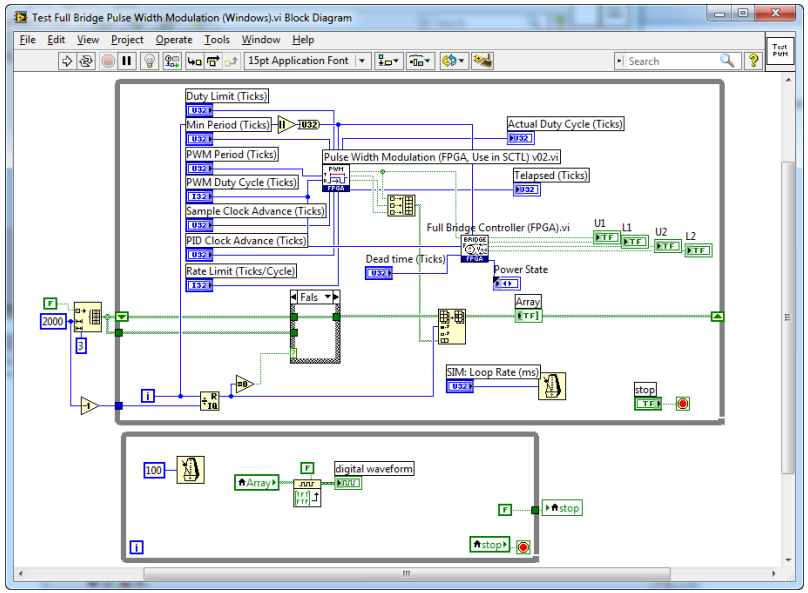
Steady state at 600 RPM:

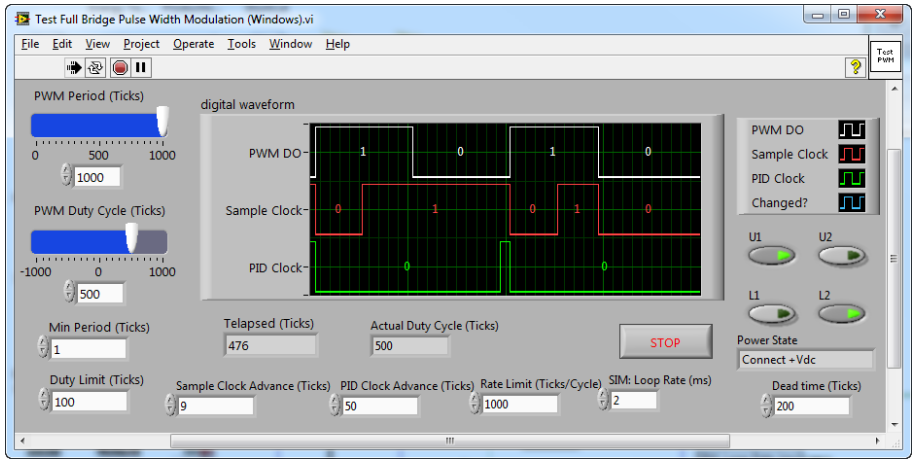


**Suggested Exercises**

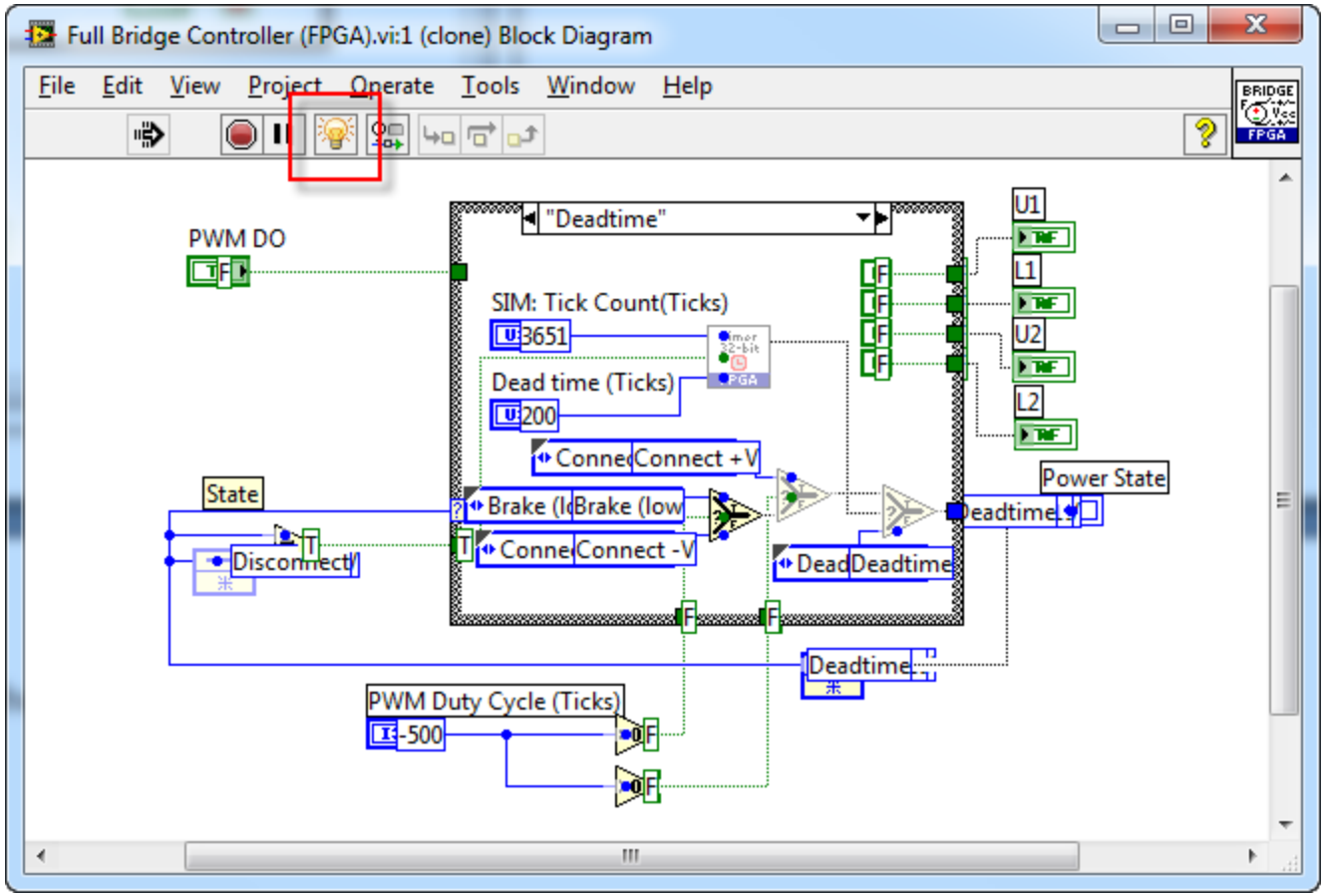
**A.** Open and run the IP core test bench application **Test Full Bridge Pulse Width Modulation (Windows).vi**.

- How does the commutation of the upper (U1, U2) and lower (L1, L2) switches respond when the **PWM Duty Cycle (Ticks)** command is changed from a positive to a negative number? What is the polarity of the voltage applied to the motor windings in each case?
- What happens when the duty cycle command is set to zero? Is the **Full Bridge Controller (FPGA).vi** IP core implementing a slow decay or fast decay (synchronous rectification) type of control logic?



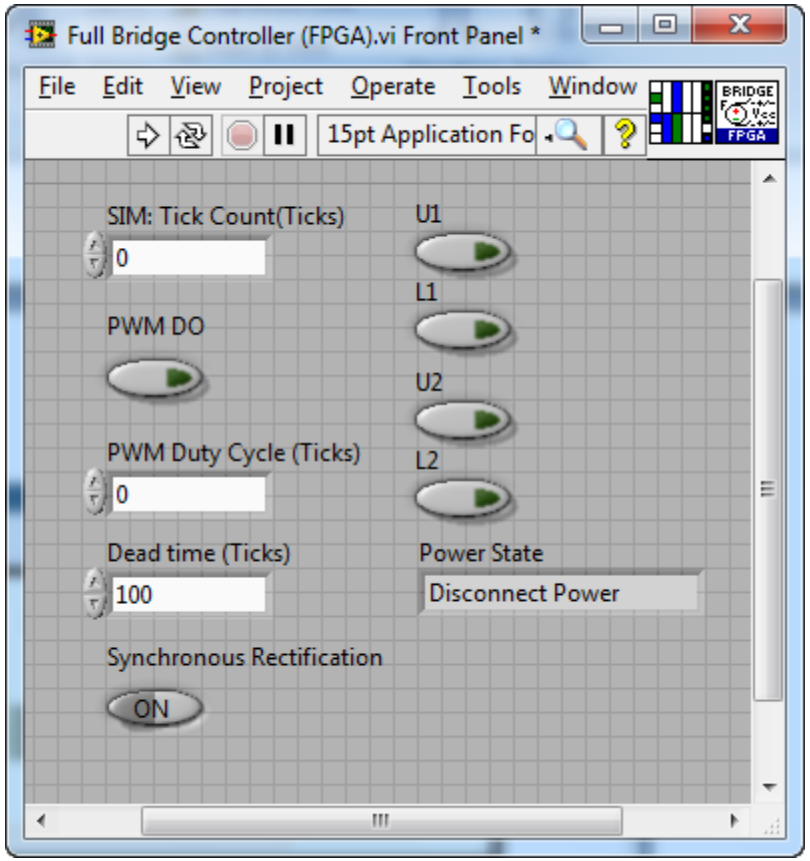
- While the testbench application is running, double click and open the **Full Bridge Controller (FPGA).vi** subVI. While the IP Core is executing, click the LabVIEW Highlight Execution button. Can you explain the operation of the state-machine logic contained in this IP core? In which cases does it transition to the **Deadtime** state? You may wish to stop execution and explore the design of this LabVIEW FPGA subVI.



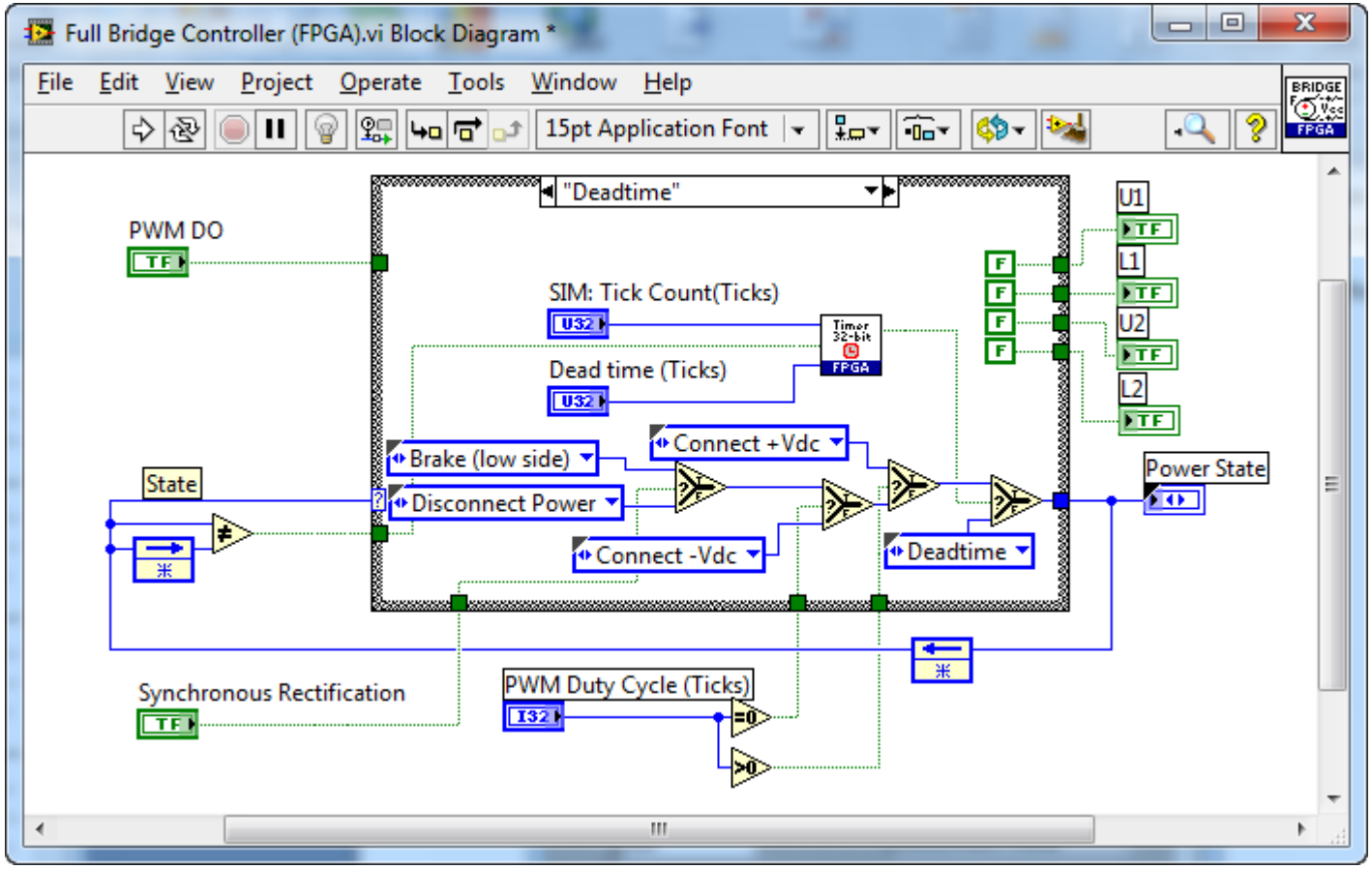

- To learn more about how to write modular, reusable LabVIEW FPGA code and test bench validation applications, see part three of the CompactRIO Developers Guide. Navigate to the section titled “LabVIEW FPGA Development Best Practices.”

**B.** Try modifying the LabVIEW FPGA H-Bridge control subVI to enable synchronous rectification to be turned on or off. Synchronous rectification connects the lower two switches in the H-bridge to ground when the PWM duty cycle is zero, causing the current in the motor to decay quickly, thereby actively braking the motor. If synchronous rectification is disabled, all four switches are open when the PWM duty cycle is zero, causing the current to decay more slowly by freewheeling through the diodes. The steps to enable synchronous rectification to be turned on or off are described below.

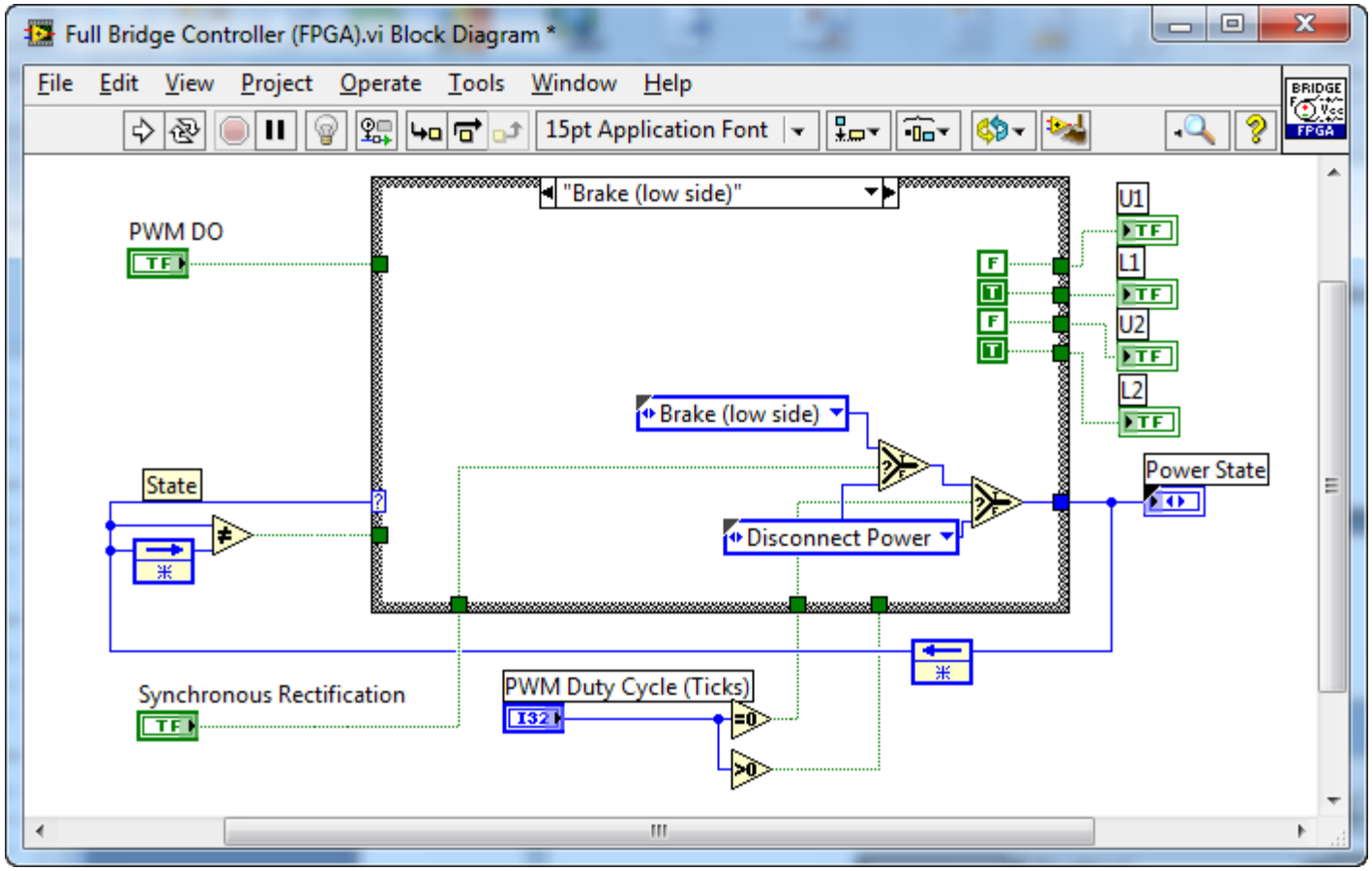
1. Add a Boolean control to the front panel of the subVI and connect it to the terminal pane.




2. Modify the **Deadtime** state machine as follows. If Synchronous Rectification is FALSE, go to the **Disconnect Power** state rather than the **Brake (low side)** state.


Full Bridge Controller (FPGA).vi Block Diagram *
File Edit View Project Operate Tools Window Help
15pt Application Font
"Deadtime"
PWM DO
SIM: Tick Count(Ticks)
Dead time (Ticks)
Brake (low side)
Disconnect Power
Connect +Vdc
Connect -Vdc
Deadtime
State
Power State
U1
L1
U2
L2
Synchronous Rectification
PWM Duty Cycle (Ticks)

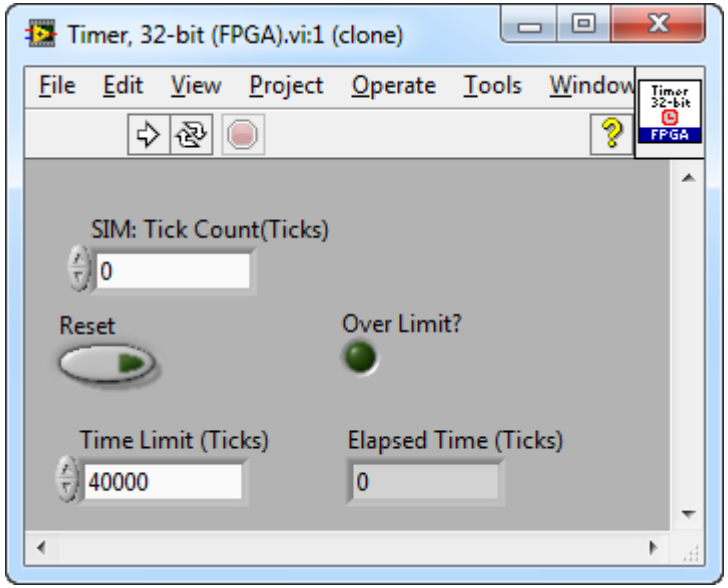
3. Modify the **Brake (low side)** state as follows.

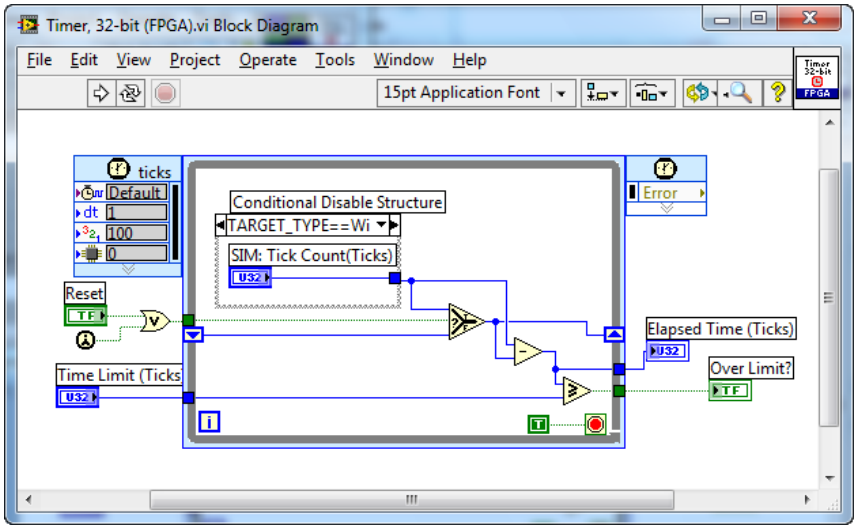



4. On the top-level simulation, right click on the modified **Full Bridge Controller (FPGA).vi** subVI and create a **Synchronous Rectification** control.
5. Run the simulation. How does the motor response differ when **Synchronous Rectification** is disabled?

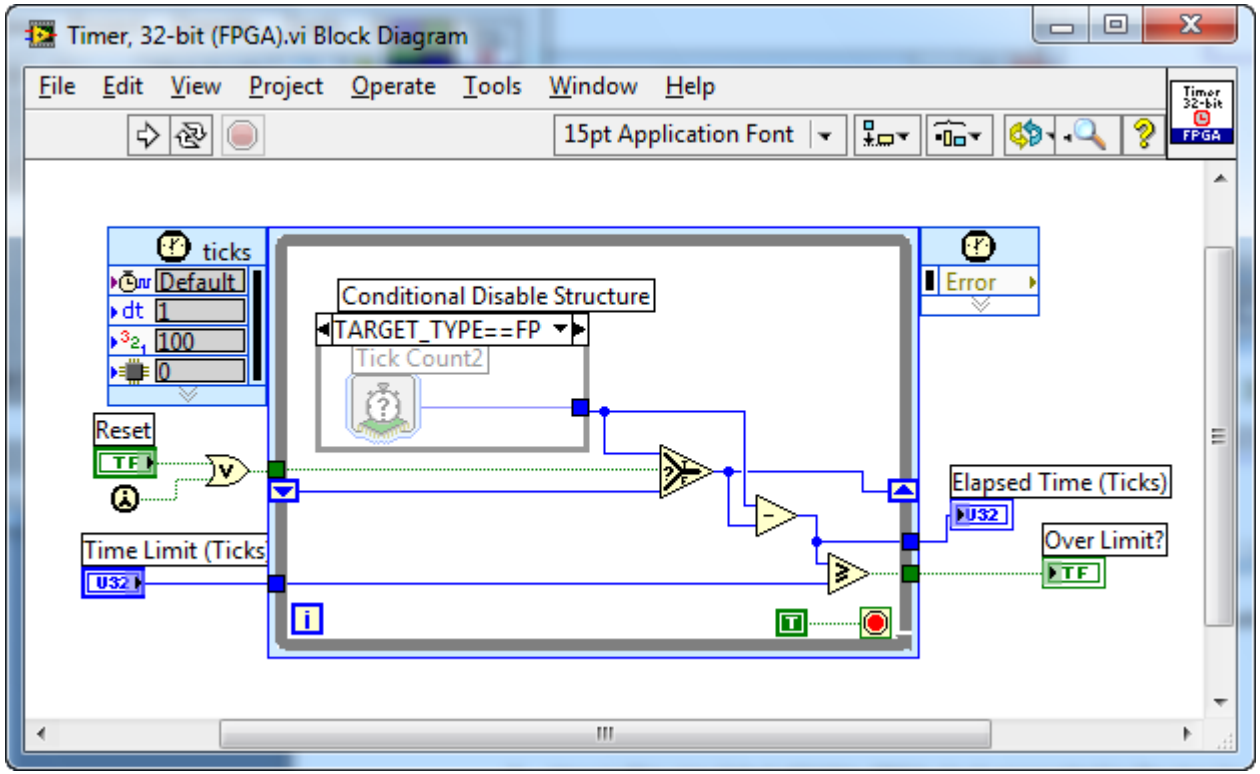
**Bonus Exercise**

Examine the design of the **Timer, 32-bit (FPGA).vi** IP core, which contains a Conditional Disable Structure that causes the code to compile differently when targeted to Windows than it does when compiled to the FPGA.

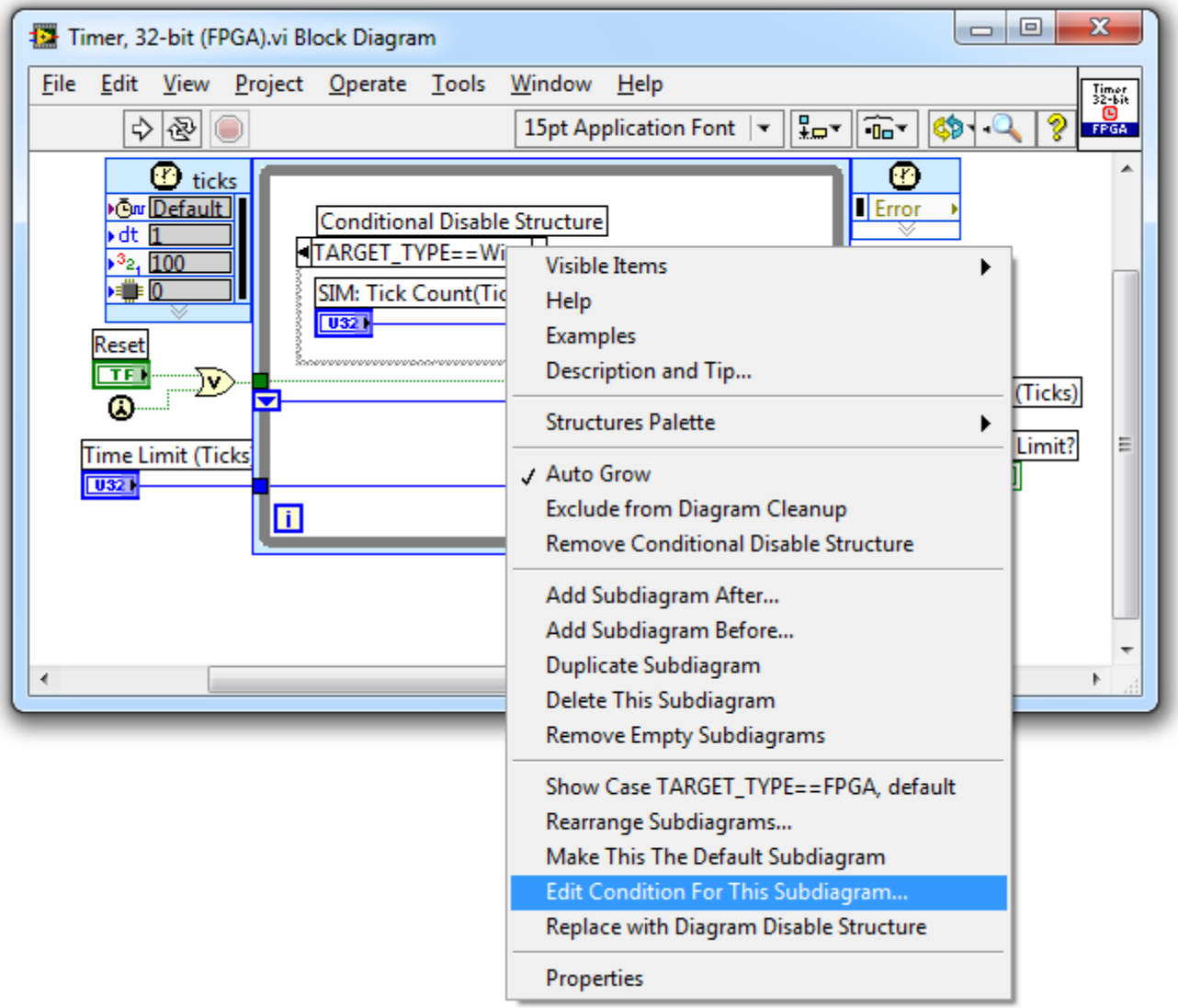


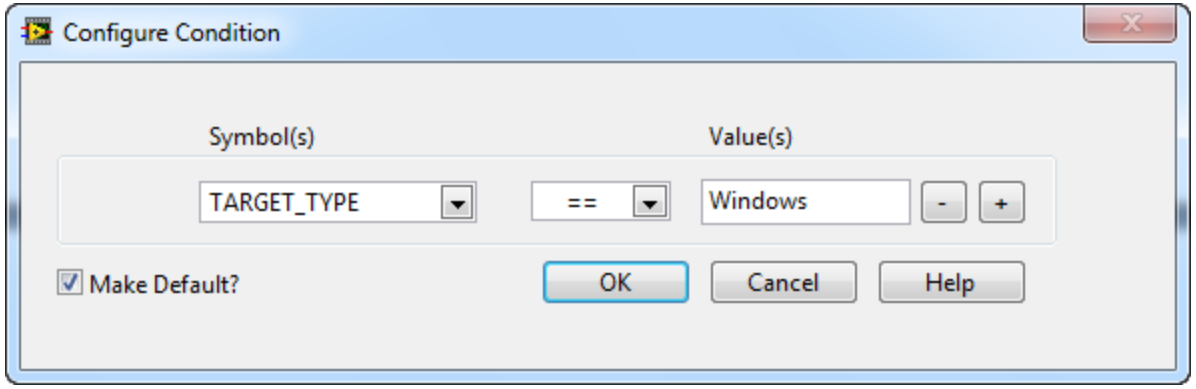

1. After opening the subVI, navigate to Operate>Change to Edit Mode.
2. Explore the Conditional Disable Structure. What is compiled differently when the TARGET_TYPE is FPGA?


Timer, 32-bit (FPGA).vi Block Diagram
File Edit View Project Operate Tools Window Help
15pt Application Font
Timer 32-bit FPGA
ticks
Default
dt
1
100
0
Conditional Disable Structure
TARGET_TYPE==FP
Tick Count2
Error
Reset
TF
Time Limit (Ticks)
U32
Elapsed Time (Ticks)
U32
Over Limit?
TF

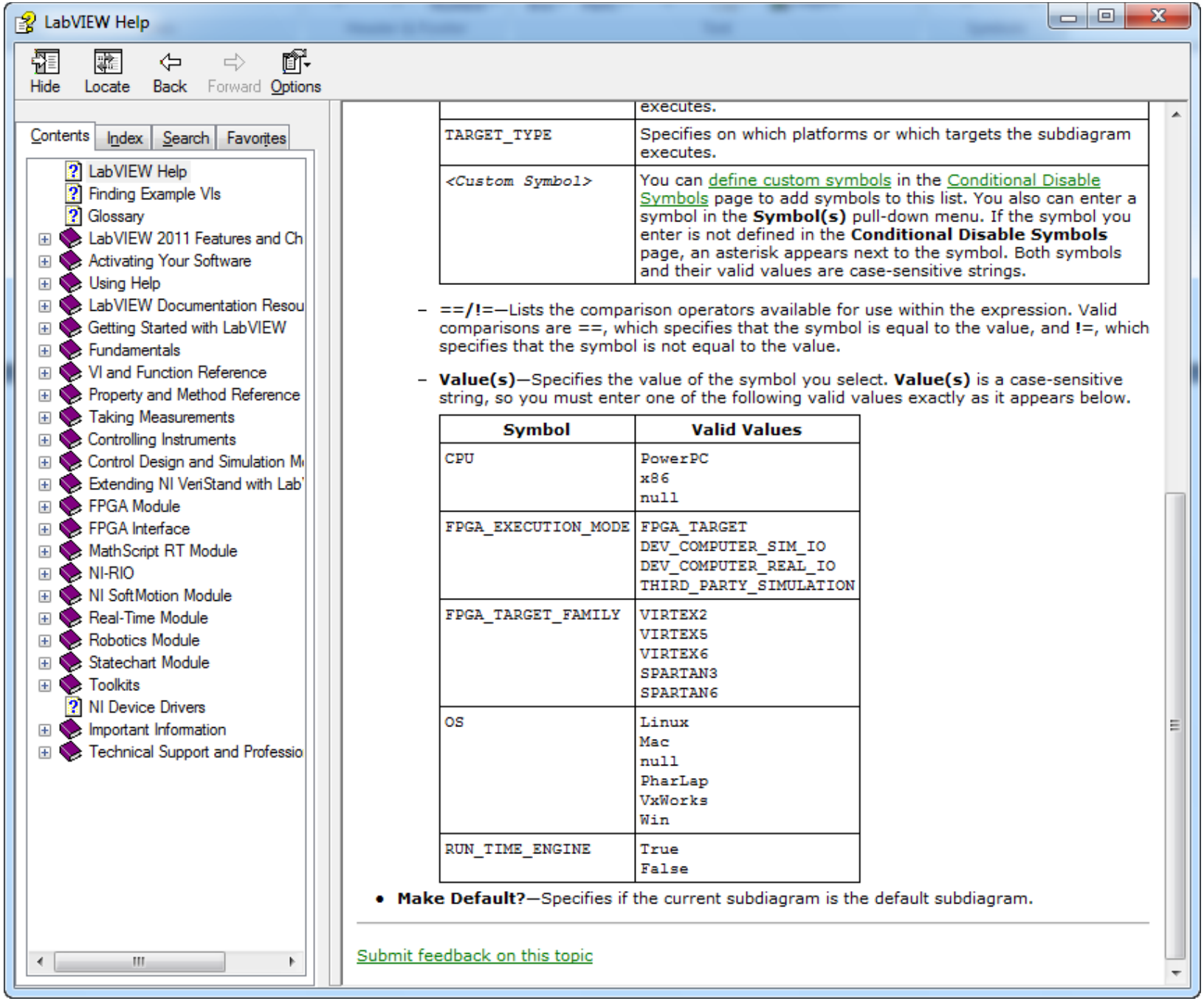
3. Modify the Conditional Disable structure so TARGET_TYPE==FPGA is not the Default. This change will cause it to use the simulated clock whenever it is NOT compiled to the FPGA. To do this, change to the TARGET_TYPE==Windows case, right-click on the **Conditional Disable Structure** and navigate to **Edit the Condition for this Subdiagram…**.



4. Check the **Make Default?** box to make **TARGET_TYPE==Windows** the default. Now it will compile for this condition when executing on any type of target other than the FPGA.




5. Optionally, click the Help button to explore the different symbol tokens that are available for a Conditional Disable structure.




executes.

| | |
|---|---|
| TARGET_TYPE | Specifies on which platforms or which targets the subdiagram executes. |
| *<Custom Symbol>* | You can define custom symbols in the Conditional Disable Symbols page to add symbols to this list. You also can enter a symbol in the **Symbol(s)** pull-down menu. If the symbol you enter is not defined in the **Conditional Disable Symbols** page, an asterisk appears next to the symbol. Both symbols and their valid values are case-sensitive strings. |

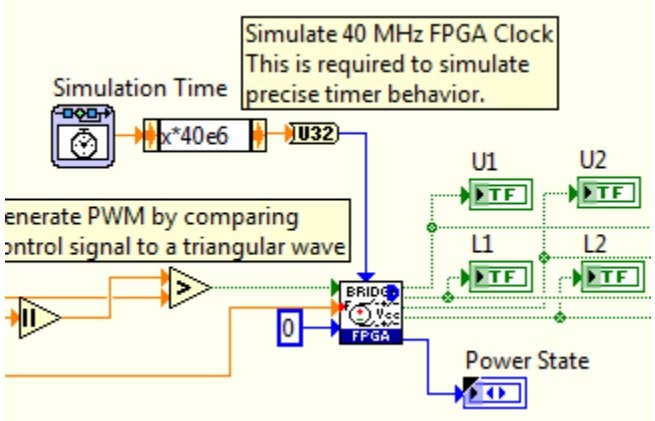
- **==/!=**—Lists the comparison operators available for use within the expression. Valid comparisons are ==, which specifies that the symbol is equal to the value, and **!=**, which specifies that the symbol is not equal to the value.
- **Value(s)**—Specifies the value of the symbol you select. **Value(s)** is a case-sensitive string, so you must enter one of the following valid values exactly as it appears below.

| Symbol | Valid Values |
|---|---|
| CPU | PowerPC<br>x86<br>null |
| FPGA_EXECUTION_MODE | FPGA_TARGET<br>DEV_COMPUTER_SIM_IO<br>DEV_COMPUTER_REAL_IO<br>THIRD_PARTY_SIMULATION |
| FPGA_TARGET_FAMILY | VIRTEX2<br>VIRTEX5<br>VIRTEX6<br>SPARTAN3<br>SPARTAN6 |
| OS | Linux<br>Mac<br>null<br>PharLap<br>VxWorks<br>Win |
| RUN_TIME_ENGINE | True<br>False |

- **Make Default?**—Specifies if the current subdiagram is the default subdiagram.

Submit feedback on this topic

6. Explore how the simulated 40 MHz FPGA clock is generated in the top level simulation application.



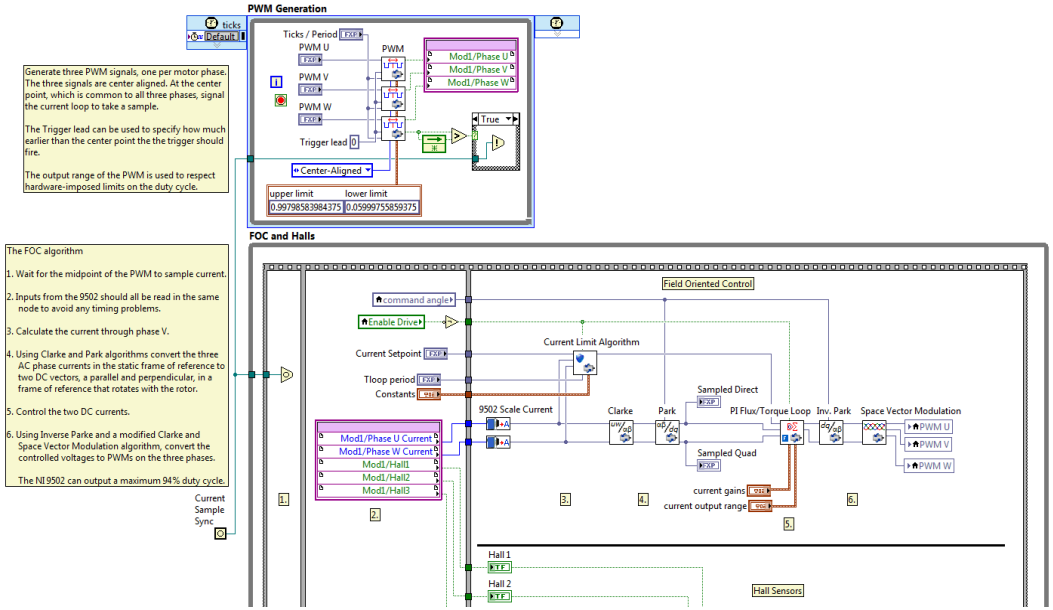

**Bonus Exercise**

- Add a power spectrum measurement to your LabVIEW code to analyze the harmonic noise in the filtered inverter output voltages.
  - What changes to the circuit design and control algorithms could be made to reduce the harmonic energy?
  - Can you develop a PWM switching algorithm for LabVIEW FPGA that performs harmonic cancellation/reduction?

# Brushed DC Motor with Gate Drive Circuit

### Co-Simulation Concepts

- Including vendor specific gate drive circuitry models in your Multisim simulations
- Using IP blocks from the NI power electronics IP library (included with NI SoftMotion 2011 f1)
    - This includes new officially supported LabVIEW FPGA IP blocks for power electronics control, and complete reference design examples for space vector field oriented control and trapezoidal commutation (see snippet below):

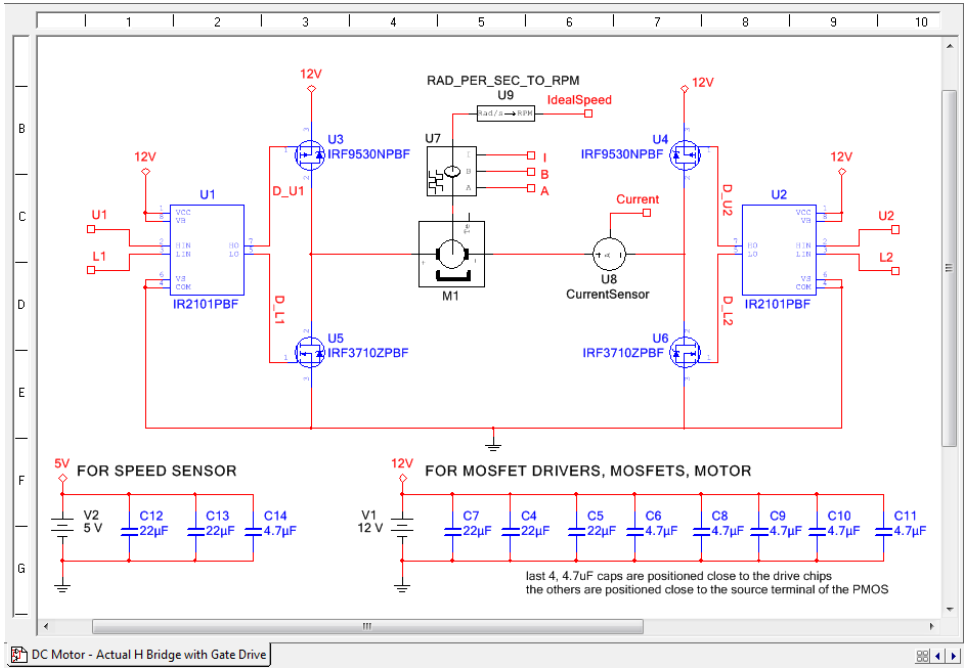


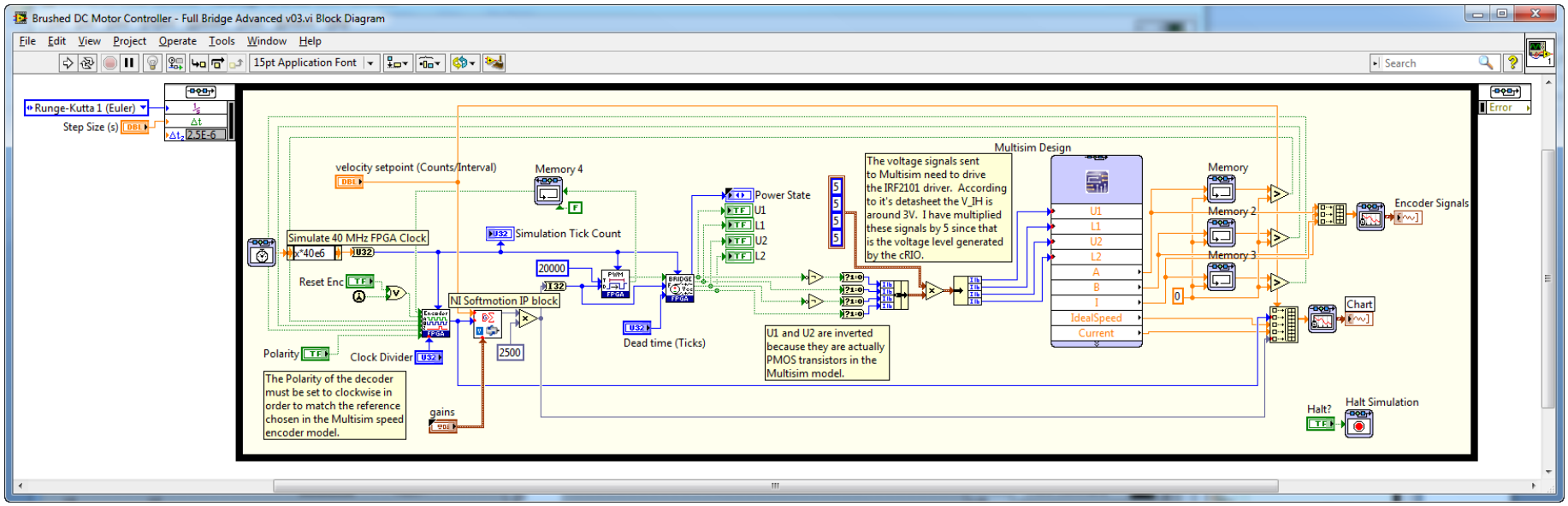

### Power Electronics Concepts

- Fundamentals of power transistor gate drive circuits, used to supply the special control signals required to effectively switch on and off the gate of power transistor components

### Required Toolkits or Modules

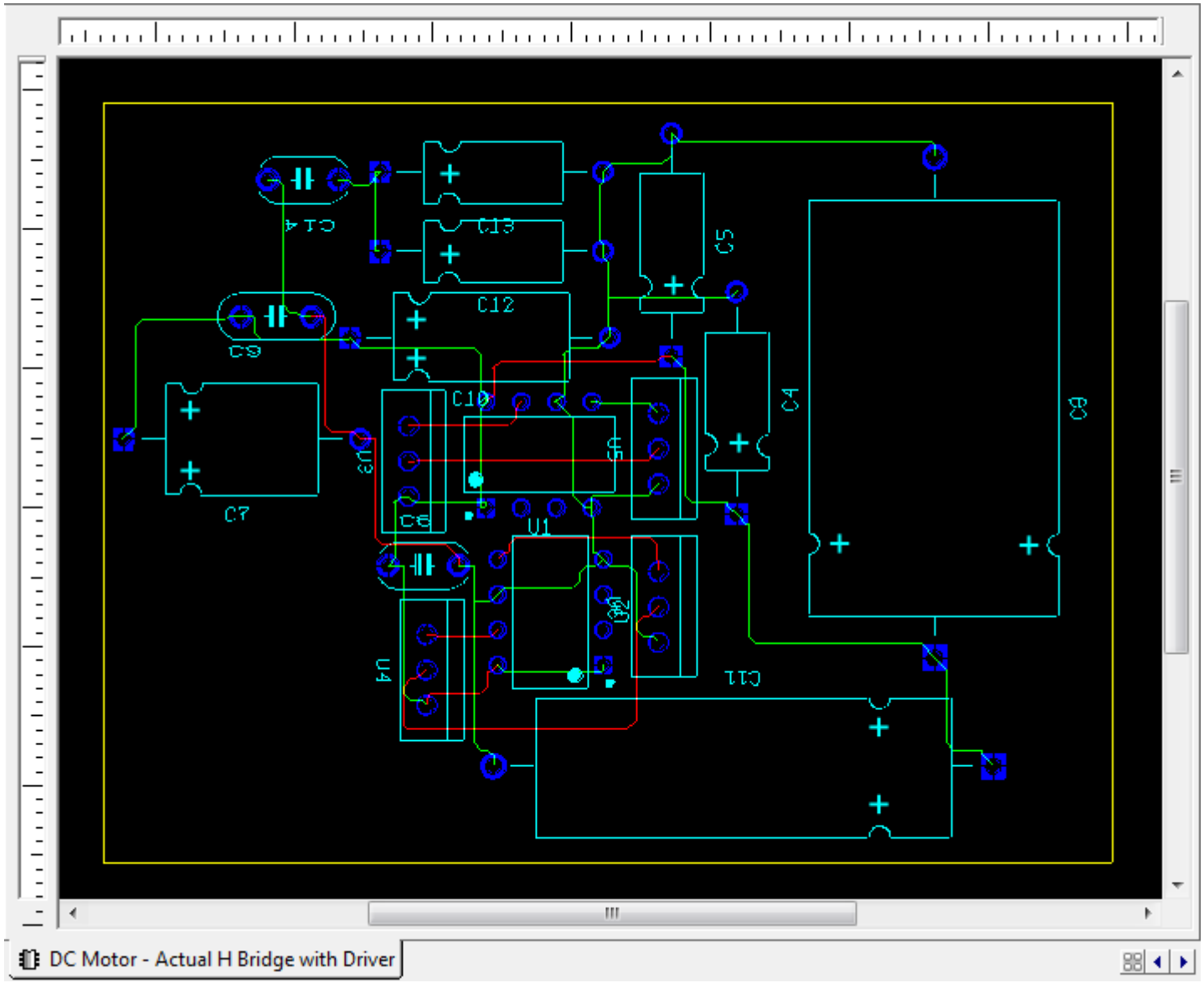
- NI LabVIEW 2011 (ni.com/labview)
- NI LabVIEW FPGA Module 2011 (ni.com/fpga)
- NI SoftMotion Module 2011 f1 (or higher) (ni.com/motion)
- NI LabVIEW Control Design & Simulation Module 2011 (help files)
- NI Circuit Design Suite 12.0 Beta 0 (or higher)(ni.com/multisim)


RAD_PER_SEC_TO_RPM
U9
IdealSpeed
12V
U3
IRF9530NPBF
U4
IRF9530NPBF
U7
I
B
A
U1
IR2101PBF
U2
IR2101PBF
D_U1
D_U2
D_L1
D_L2
U5
IRF3710ZPBF
U6
IRF3710ZPBF
M1
Current
U8
CurrentSensor
L1
L2
5V
FOR SPEED SENSOR
V2
5 V
C12
22µF
C13
22µF
C14
4.7µF
12V
FOR MOSFET DRIVERS, MOSFETS, MOTOR
V1
12 V
C7
22µF
C4
22µF
C5
22µF
C6
4.7µF
C8
4.7µF
C9
4.7µF
C10
4.7µF
C11
4.7µF
last 4, 4.7uF caps are positioned close to the drive chips
the others are positioned close to the source terminal of the PMOS
DC Motor - Actual H Bridge with Gate Drive


Brushed DC Motor Controller - Full Bridge Advanced v03.vi Block Diagram
File Edit View Project Operate Tools Window Help
15pt Application Font
Search
Runge-Kutta 1 (Euler)
Step Size (s)
2.5E-6
velocity setpoint (Counts/Interval)
Memory 4
Simulate 40 MHz FPGA Clock
Simulation Tick Count
Reset Enc
20000
NI Softmotion IP block
Power State
U1
L1
U2
L2
Dead time (Ticks)
2500
Polarity
Clock Divider
The Polarity of the decoder must be set to clockwise in order to match the reference chosen in the Multisim speed encoder model.
gains
The voltage signals sent to Multisim need to drive the IRF2101 driver. According to it's detasheet the V_IH is around 3V. I have multiplied these signals by 5 since that is the voltage level generated by the cRIO.
Multisim Design
U1
L1
U2
L2
A
B
I
IdealSpeed
Current
U1 and U2 are inverted because they are actually PMOS transistors in the Multisim model.
Memory
Memory 2
Memory 3
Encoder Signals
Chart
Halt?
Halt Simulation
Error


Brushed DC Motor Controller - Full Bridge Advanced v03.vi
File Edit View Project Operate Tools Window Help
Encoder Signals
A B I
Time
Speed (C/I)
Speed (RPM)
Current (A)
PWM Command
Simulation Time
Step Size (s)
1E-5
Dead time (Ticks)
10
Clock Divider
10
Reset Enc
Polarity
CW
Halt?
Simulation Tick Count
1081360
gains
Kp
1
Ki
0
ILim
1000
velocity setpoint (Counts/Interval)
100
U1
U2
L1
L2
Power State
Connect +Vdc
100.00
98.00
575.77
-0.35
5000.00
Speed Setpoint (C/I)
Encoder Speed (C/I)
Actual Speed (RPM)
Current (A)
PWM Command

Ultiboard Layout (draft):



# Brushless DC Motor with Hall Sensors (Trapezoidal Flux)

**Co-Simulation Concepts**

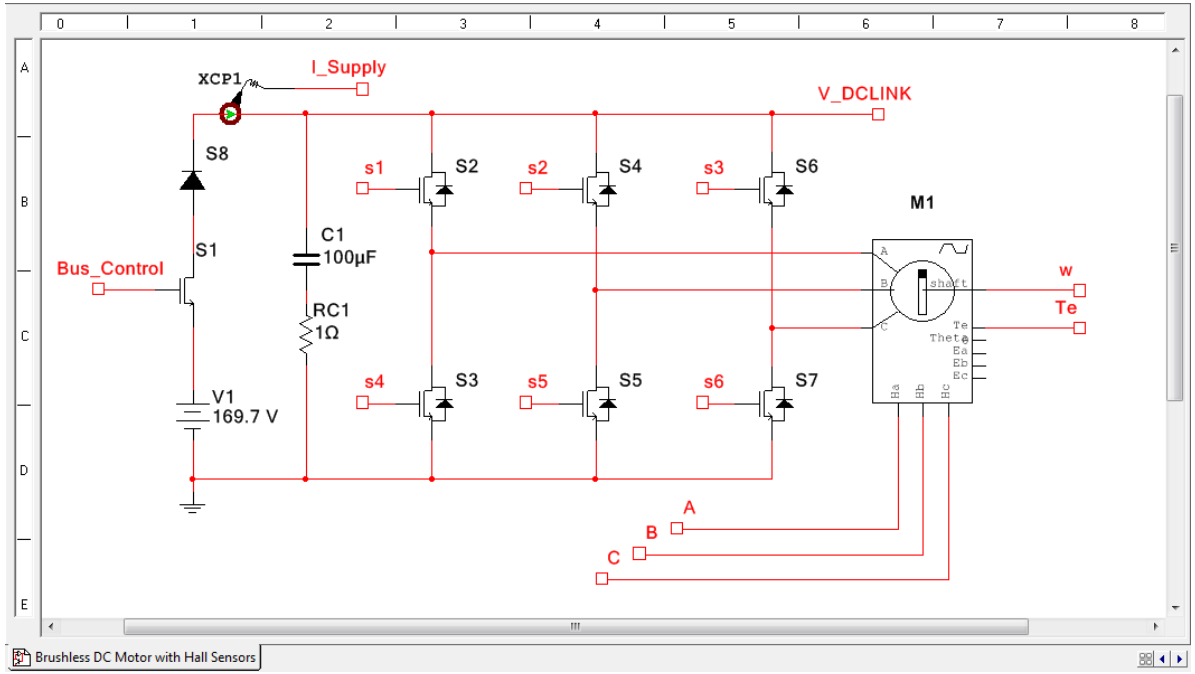
- Zero Crossing detection in LabVIEW Control Design & Simulation
    - Enables a variable step solver to precisely capture the timing of digital (ON/OFF) event transitions.
- Dead-time insertion for preventing shoot through currents that could potentially damage power transistor switches if the upper and lower switches are closed simultaneously
    - The LabVIEW FPGA IP core in this example delays only the rising edge transitions, enabling the IGBTs to be switched off without delay while inserting a delay during switch turn on events.

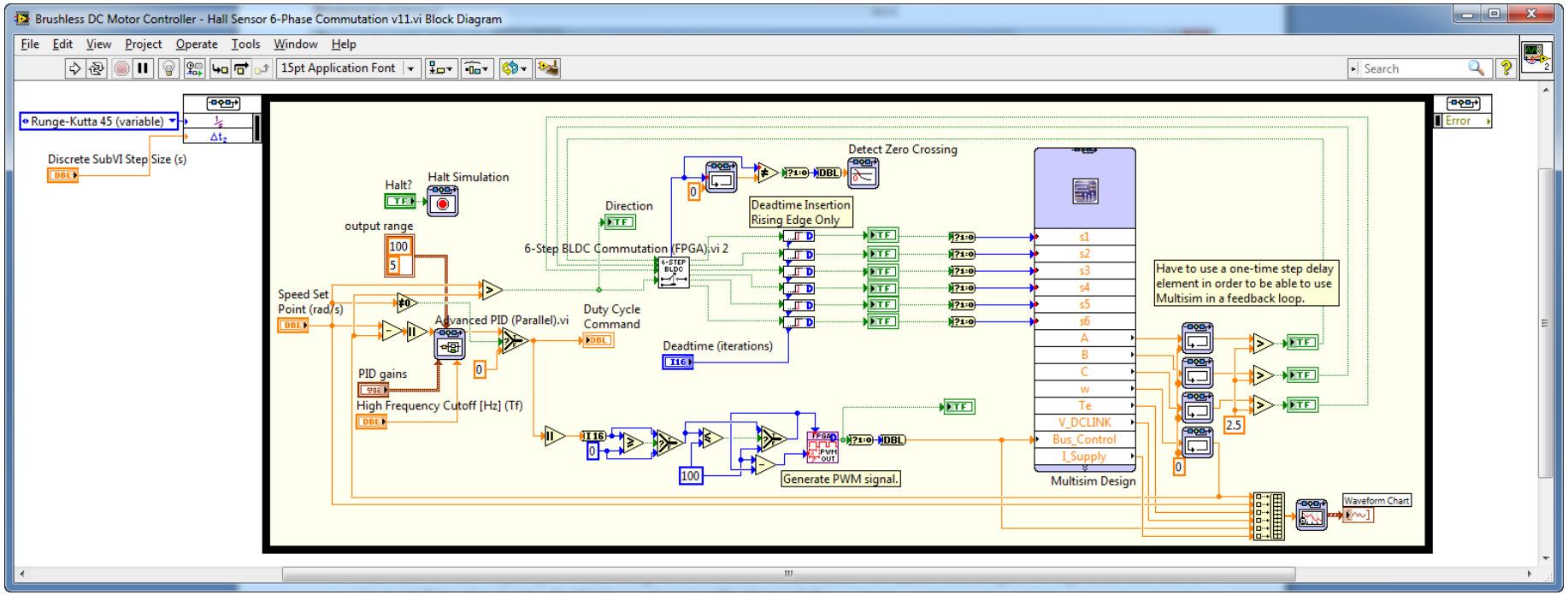
**Power Electronics Concepts**

- Fundamentals of trapezoidal flux brushless DC motor control using a three-phase inverter, Hall Effect position sensors, and a basic 6-step commutation algorithm (learn more)

**Required Toolkits or Modules**

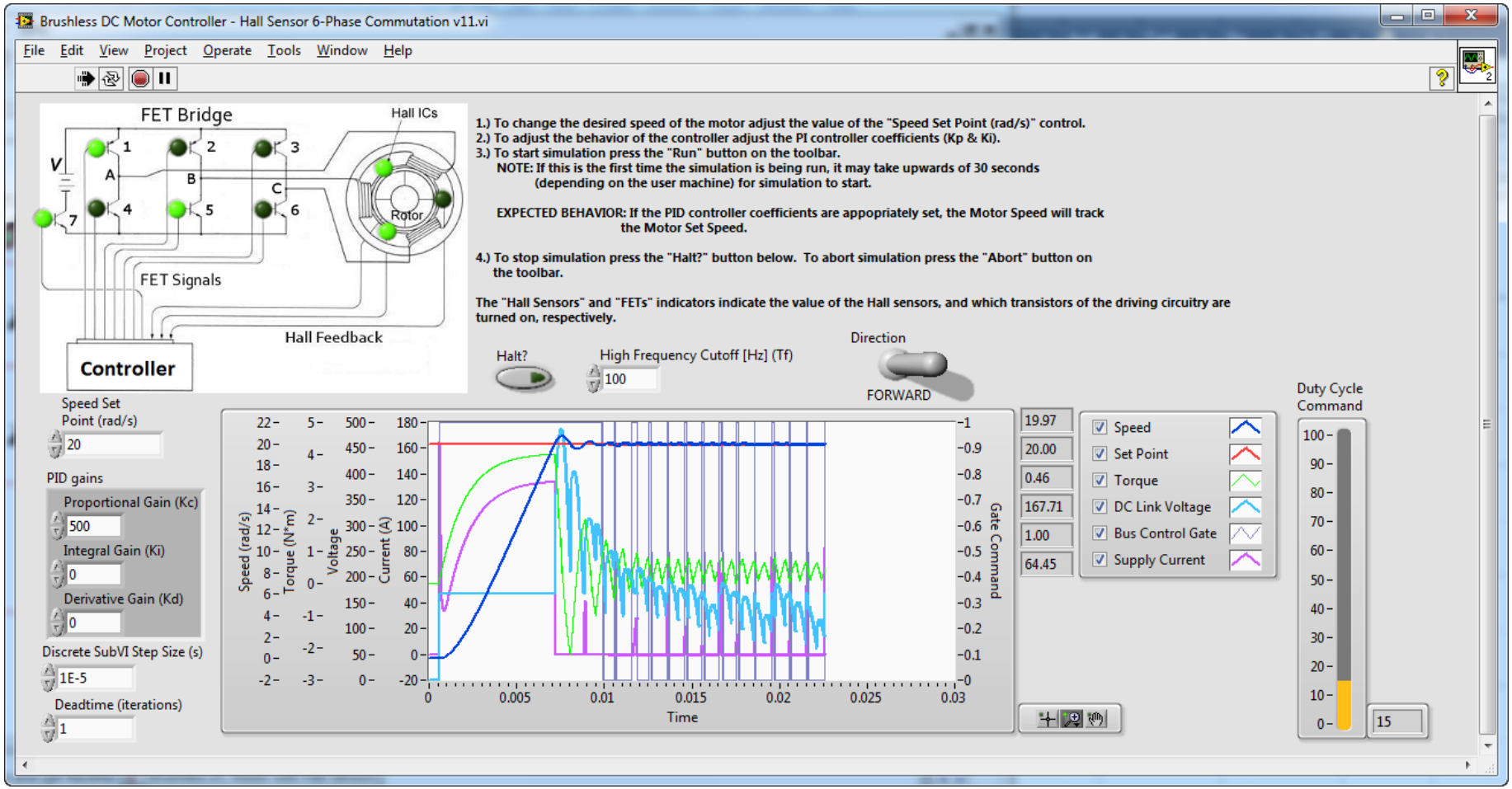
- NI LabVIEW 2011 (ni.com/labview)
- NI LabVIEW FPGA Module 2011 (ni.com/fpga)
- NI SoftMotion Module 2011 f1 (or higher) (ni.com/motion)
- NI LabVIEW Control Design & Simulation Module 2011 (help files)
- NI Circuit Design Suite 12.0 Beta 0 (or higher)(ni.com/multisim)


XCP1
I_Supply
V_DCLINK
S8
s1
S2
s2
S4
s3
S6
M1
S1
C1
100µF
Bus_Control
RC1
1Ω
w
Te
s4
S3
s5
S5
s6
S7
V1
169.7 V
A
B
C
Brushless DC Motor with Hall Sensors


Brushless DC Motor Controller - Hall Sensor 6-Phase Commutation v11.vi Block Diagram
File Edit View Project Operate Tools Window Help
15pt Application Font
Search
Runge-Kutta 45 (variable)
Discrete SubVI Step Size (s)
Halt?
Halt Simulation
output range
100
5
Direction
Detect Zero Crossing
Deadtime Insertion
Rising Edge Only
6-Step BLDC Commutation (FPGA).vi 2
Speed Set
Point (rad/s)
Advanced PID (Parallel).vi
Duty Cycle
Command
Deadtime (iterations)
PID gains
High Frequency Cutoff [Hz] (Tf)
Generate PWM signal.
Have to use a one-time step delay
element in order to be able to use
Multisim in a feedback loop.
s1
s2
s3
s4
s5
s6
A
B
C
w
Te
V_DCLINK
Bus_Control
I_Supply
Multisim Design
Waveform Chart
Error

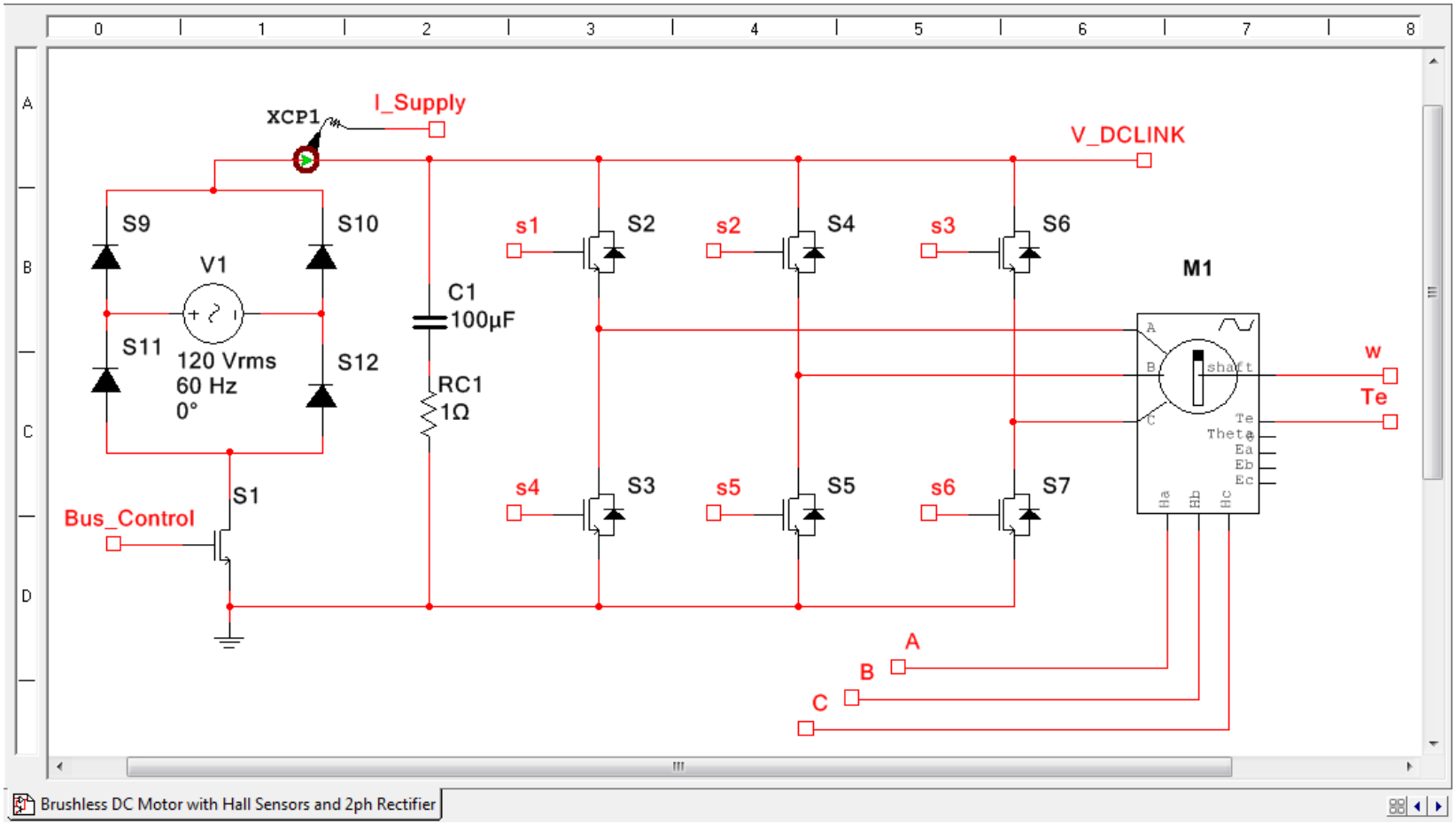
**Simulation Results**

- Note the significant torque ripple that occurs during commutation events. Can you explain why this occurs?
- Note the 1 kHz PWM frequency for the bus voltage control transistor. What would be the impact of executing the PWM generator at a faster or slower rate? How does changing the DC link capacitor value affect the results?

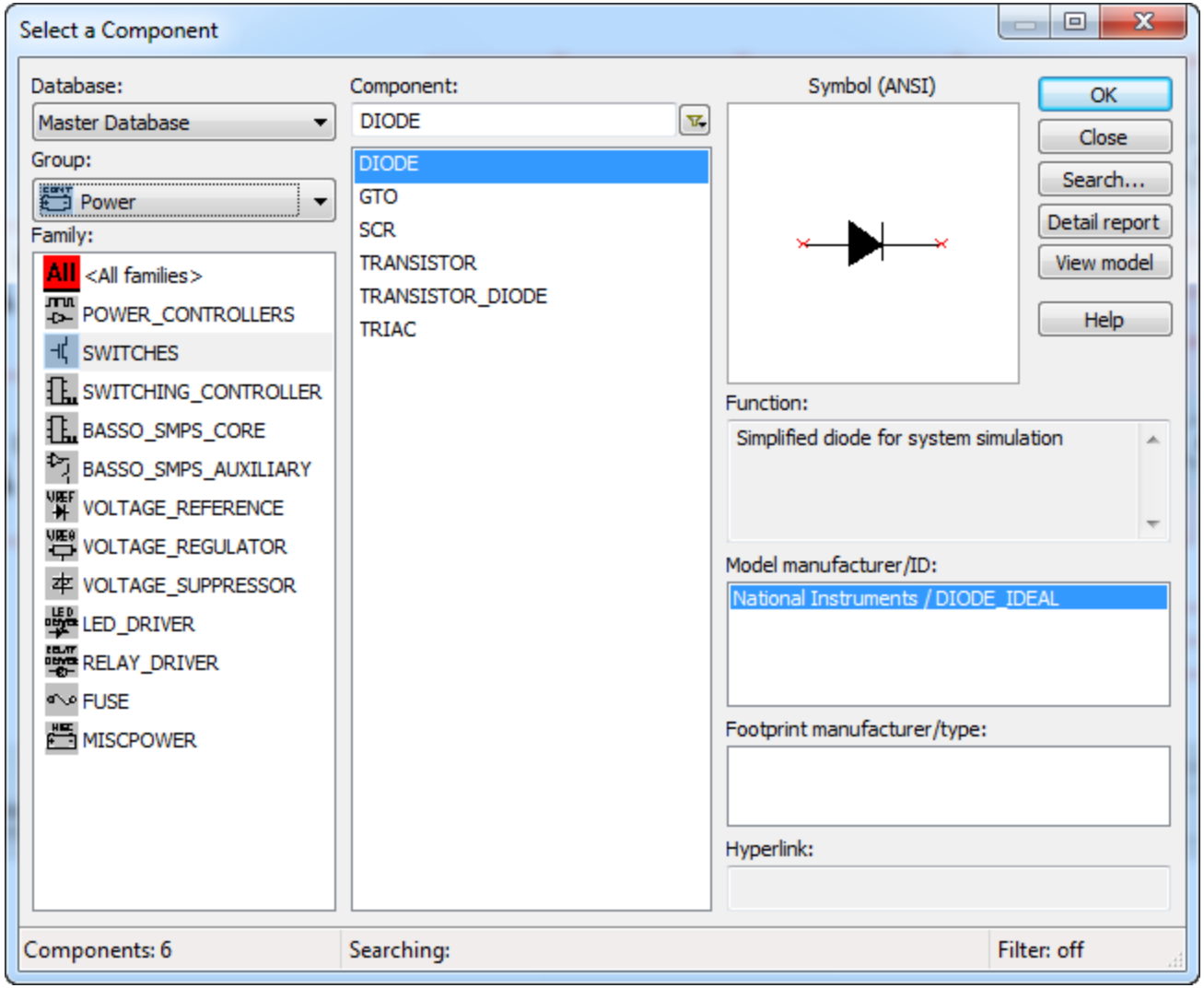


**Suggested Exercises**

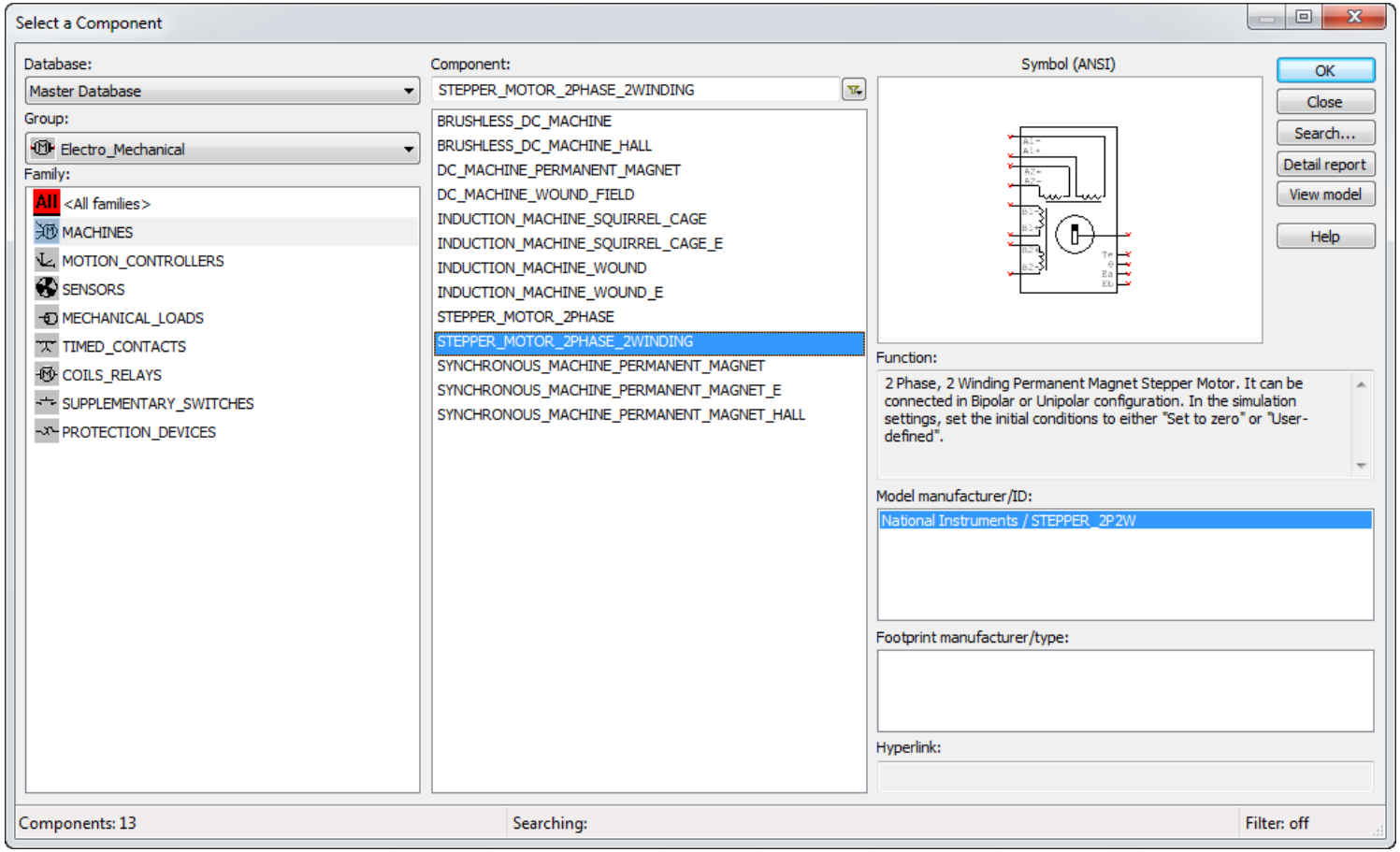
- Try adding a 120 VAC supply and 2-phase, full bridge diode rectifier circuit to produce the DC link voltage, as shown below. How does this affect the response of the system?

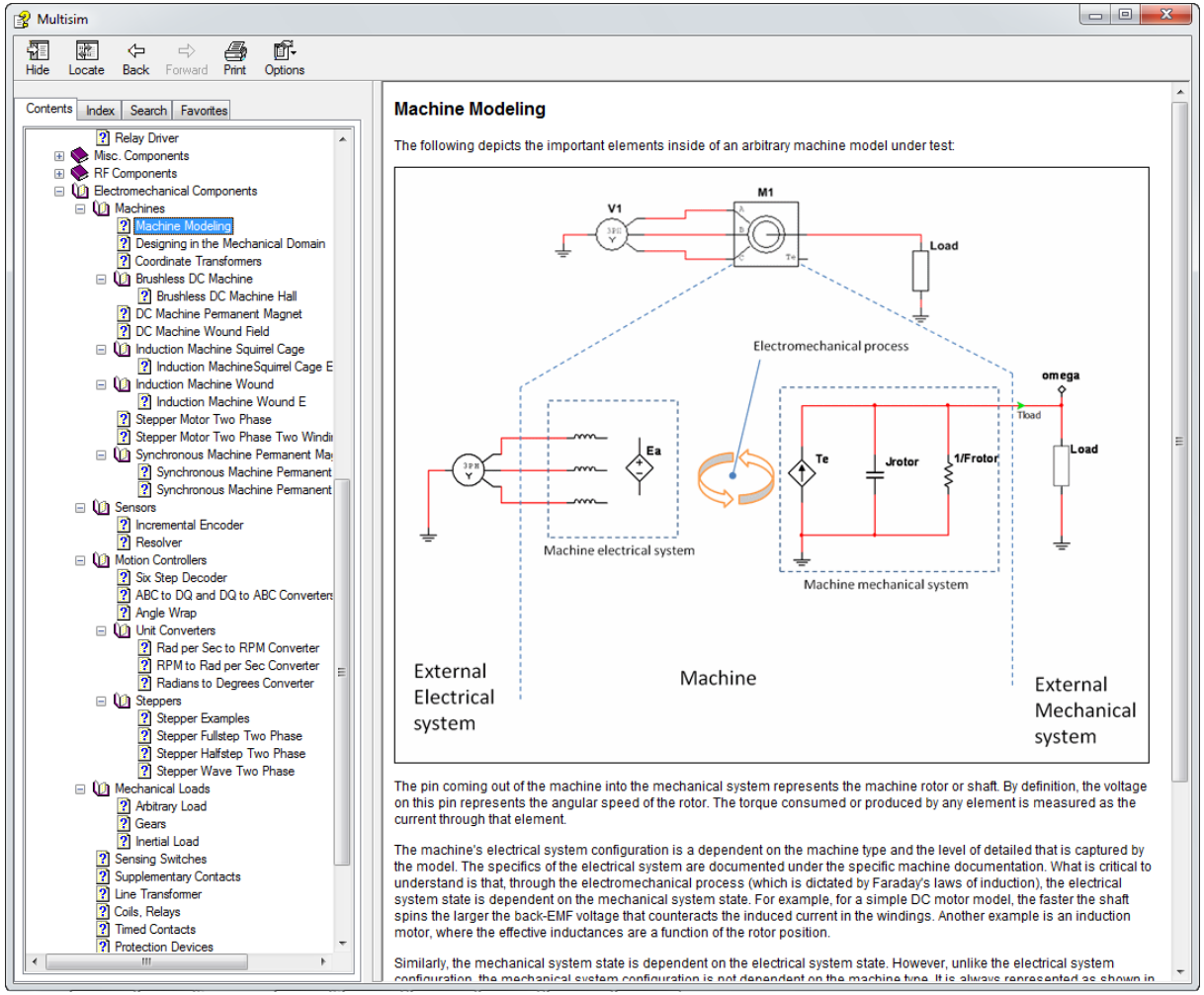



  - Hint: You can find the basic power diode component in Multisim by right-clicking on a blank area of the schematic, selecting **Place Component…** and navigating to **Group: Power**, **Family: Switches**.



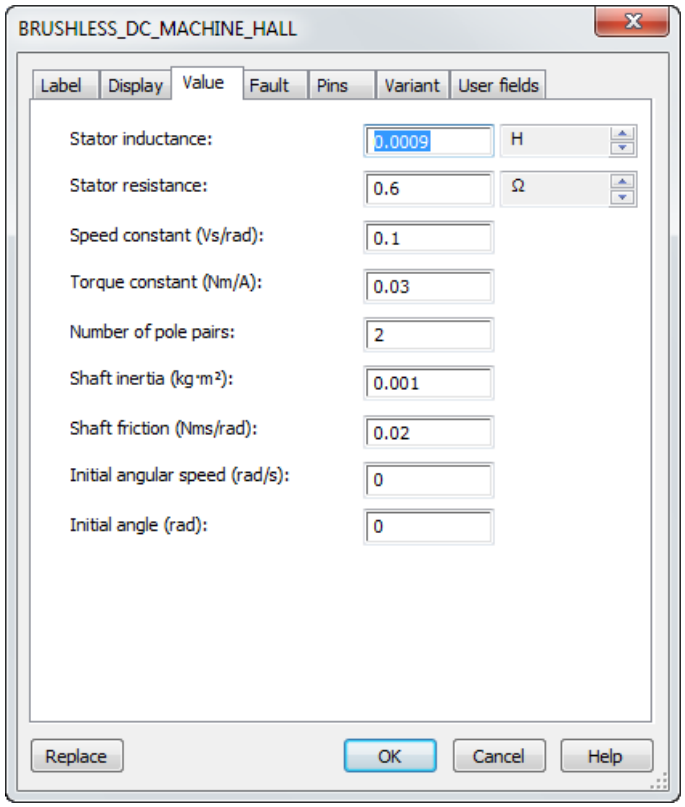

- Explore the Electromechanical component library and help documentation in Multisim (shown below). Can you figure out how to operate the BLDC as a generator by applying an external torque that you control in LabVIEW? What happens to the DC link voltage?
    - Try adding a contactor relay and power resistor to dump the extra power to ground when the DC link voltage exceeds a programmable threshold.
    - Advanced: Try adding an additional 3-leg inverter circuit to convert the DC link voltage to 3-phase AC power for the grid. You now have a bidirectional inverter, also known as an active front end for the motor, that enables you to put regenerative braking power onto the grid. Can you demonstrate proper control of your motor/generator in all four quadrants of operation?

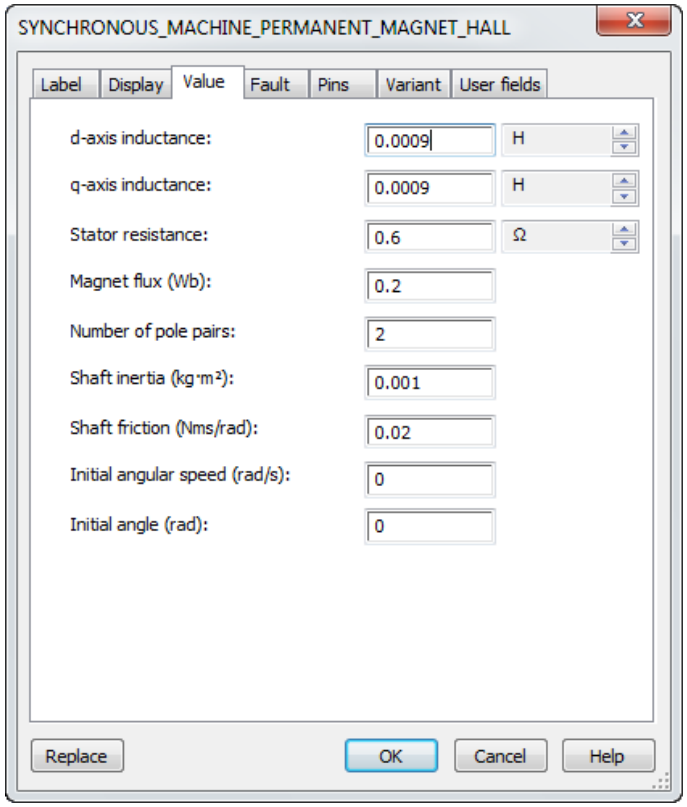

Select a Component
Database:
Master Database
Group:
Electro_Mechanical
Family:
<All families>
MACHINES
MOTION_CONTROLLERS
SENSORS
MECHANICAL_LOADS
TIMED_CONTACTS
COILS_RELAYS
SUPPLEMENTARY_SWITCHES
PROTECTION_DEVICES
Component:
STEPPER_MOTOR_2PHASE_2WINDING
BRUSHLESS_DC_MACHINE
BRUSHLESS_DC_MACHINE_HALL
DC_MACHINE_PERMANENT_MAGNET
DC_MACHINE_WOUND_FIELD
INDUCTION_MACHINE_SQUIRREL_CAGE
INDUCTION_MACHINE_SQUIRREL_CAGE_E
INDUCTION_MACHINE_WOUND
INDUCTION_MACHINE_WOUND_E
STEPPER_MOTOR_2PHASE
STEPPER_MOTOR_2PHASE_2WINDING
SYNCHRONOUS_MACHINE_PERMANENT_MAGNET
SYNCHRONOUS_MACHINE_PERMANENT_MAGNET_E
SYNCHRONOUS_MACHINE_PERMANENT_MAGNET_HALL
Symbol (ANSI)
OK
Close
Search...
Detail report
View model
Help
Function:
2 Phase, 2 Winding Permanent Magnet Stepper Motor. It can be connected in Bipolar or Unipolar configuration. In the simulation settings, set the initial conditions to either "Set to zero" or "User-defined".
Model manufacturer/ID:
National Instruments / STEPPER_2P2W
Footprint manufacturer/type:
Hyperlink:
Components: 13
Searching:
Filter: off



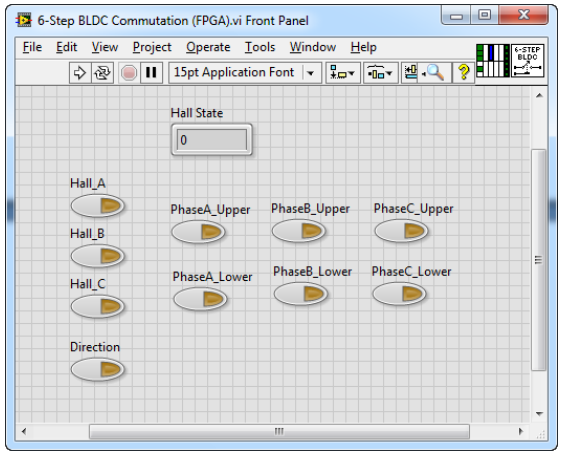

- Add HB/SC connectors for the motor back EMF signals Ea, Eb and Ec. Chart the hall sensor signals together with the back EMF signals. Can you explain the relationship?
    - Hint: Compare the following:
        - Hall_A to Ea-Ec
        - Hall_B to Eb-Ea
        - Hall_C to Ec-Eb
- In Multisim, double click the trapezoidal back EMF brushless DC machine component (BRUSHLESS_DC_MACHINE_HALL) and write down the parameters. Next, right-click and replace with a permanent magnet synchronous machine (SYNCHRONOUS_MACHINE_PERMANENT_MAGNET_HALL) and enter the same parameters where applicable.

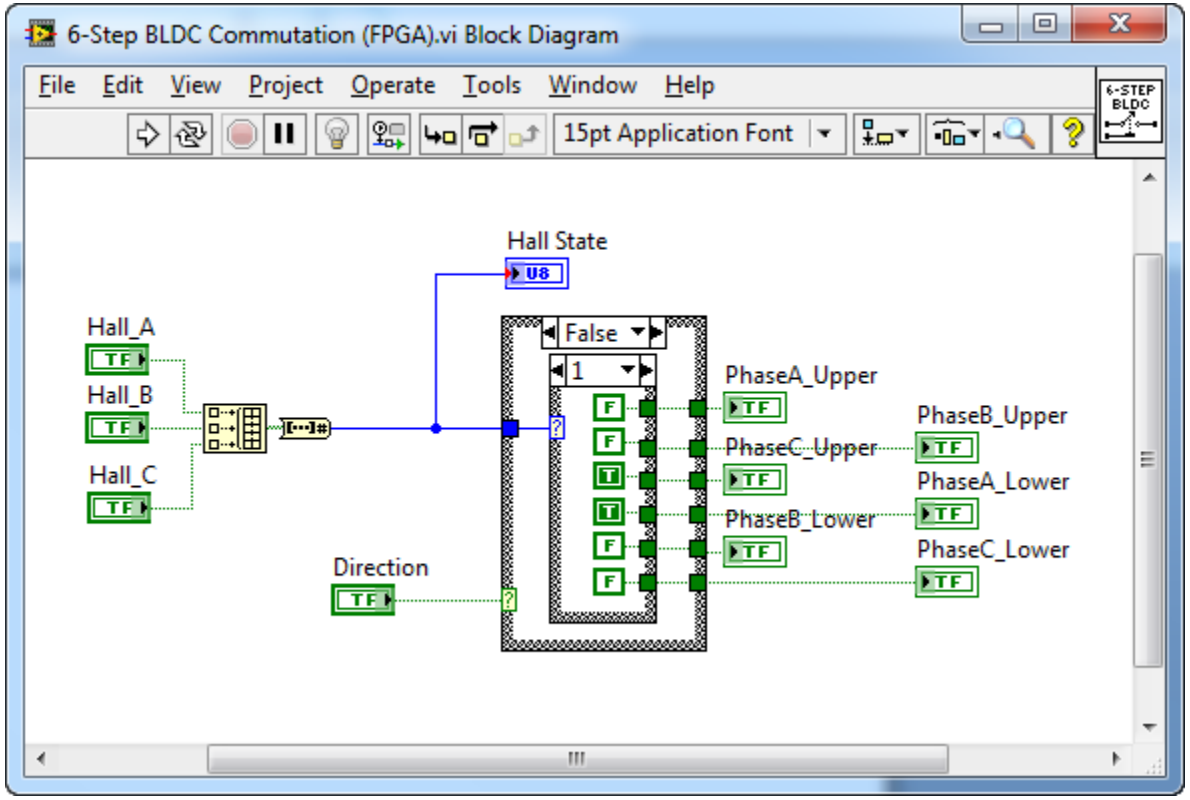
- Does your LabVIEW control system still work?
- How does performance and back EMF signals of the PMSM compare to the BLDC machine?



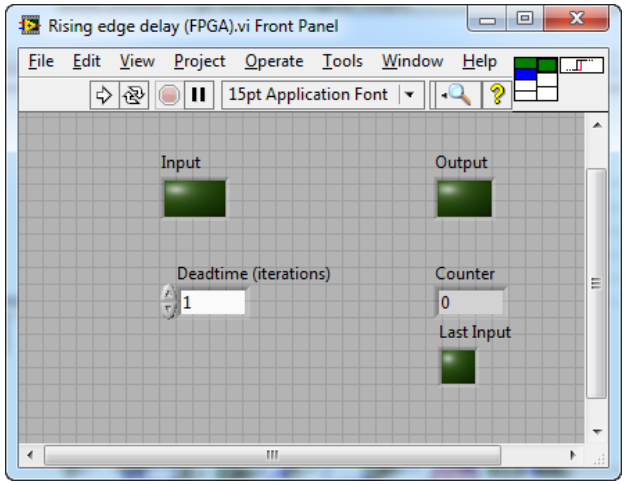



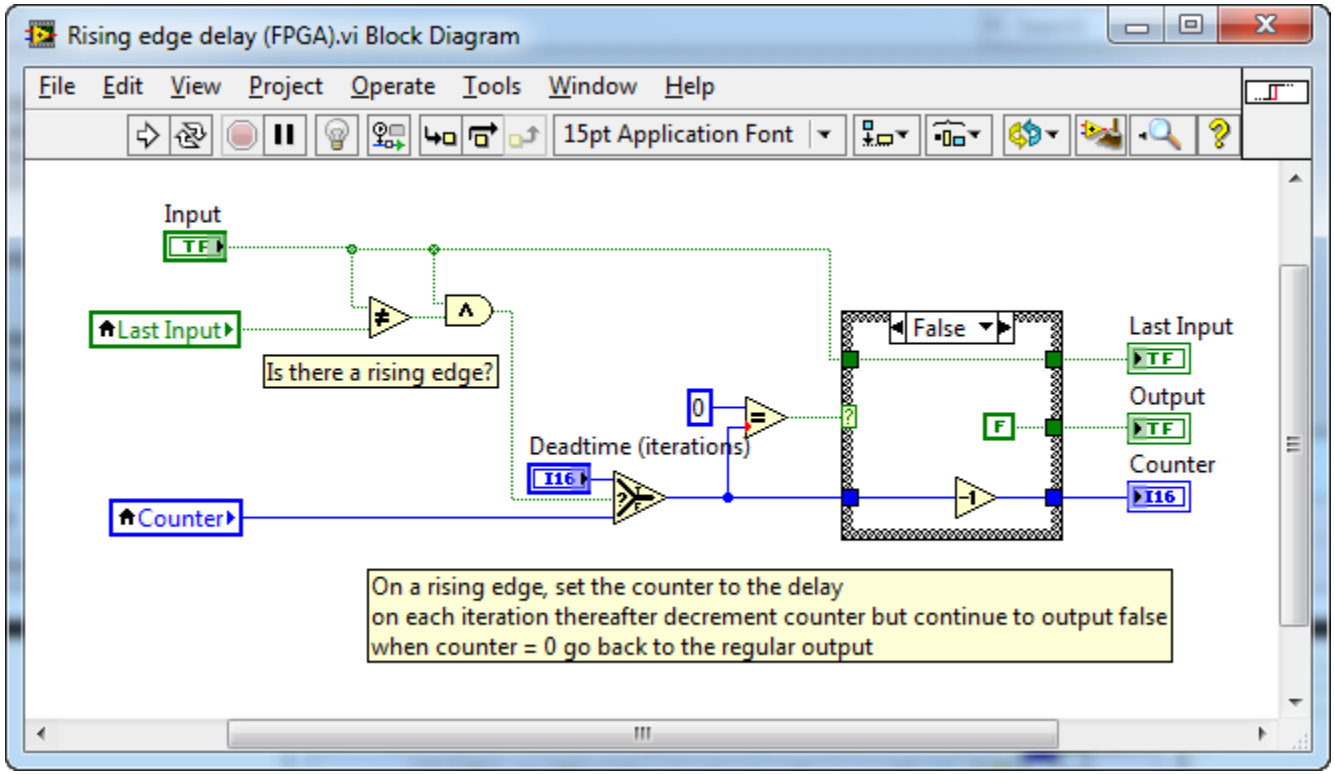
6-Step BLDC Commutation IP Core:



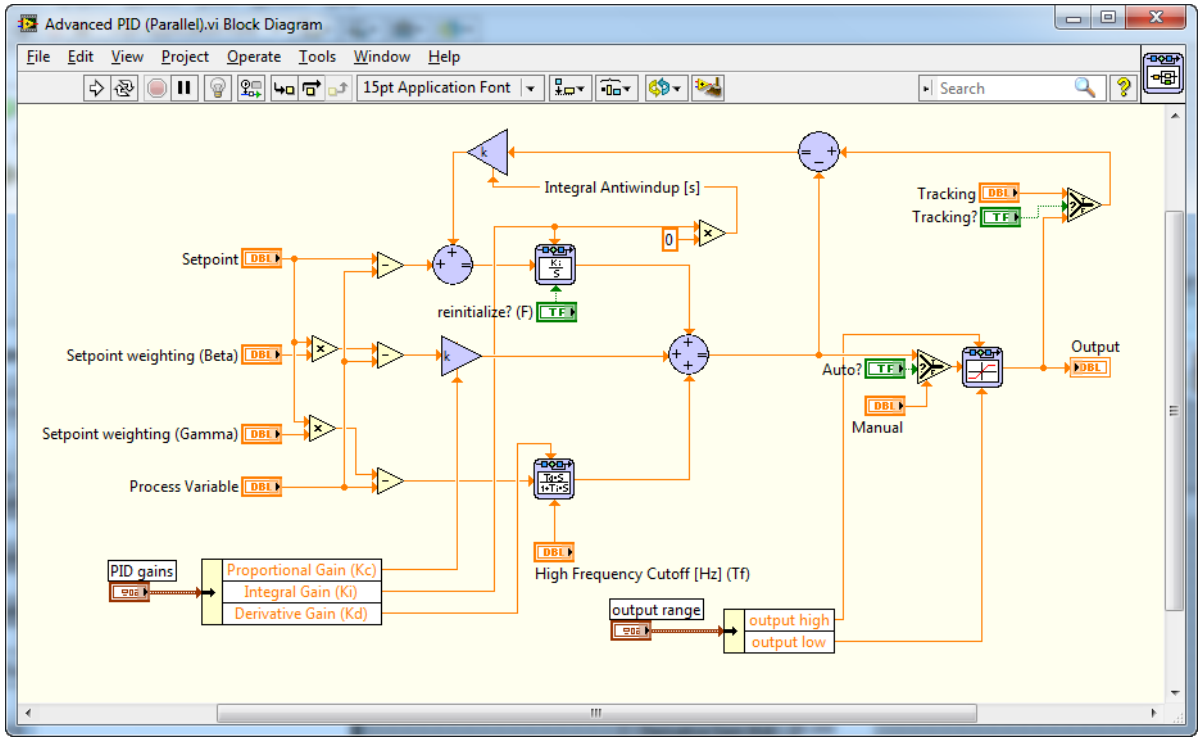



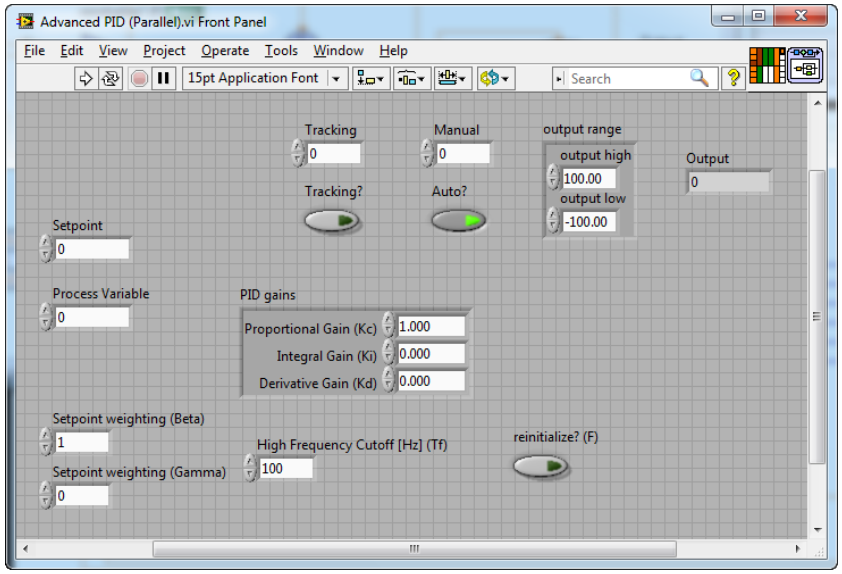
Rising Edge Delay Deadtime Insertion IP Core:



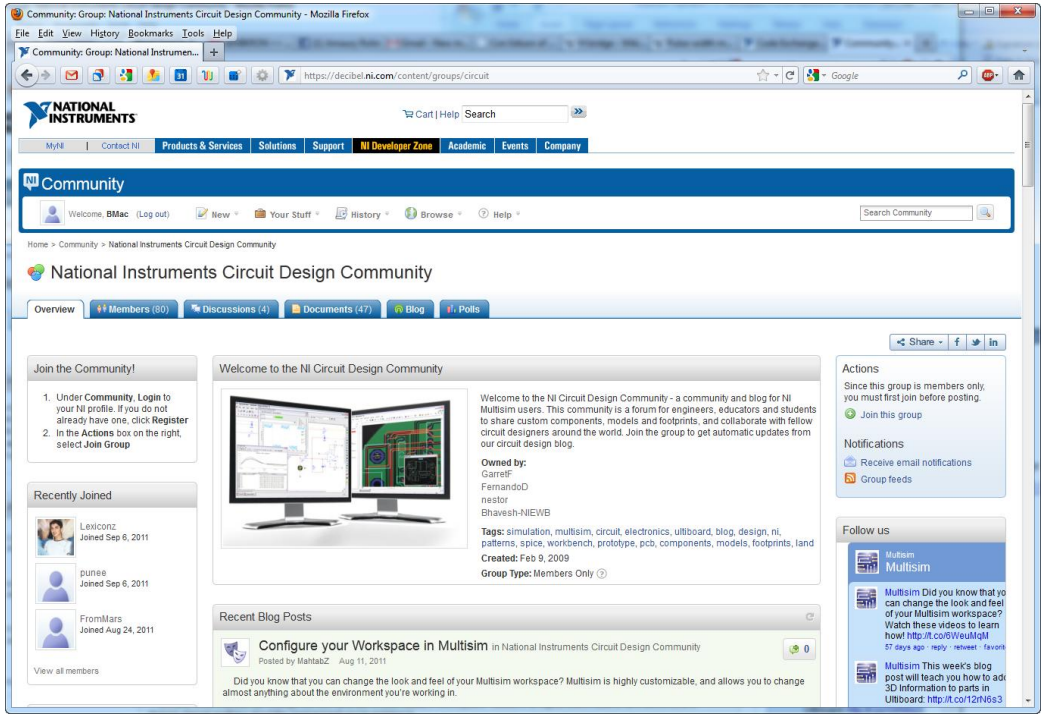
Advanced PID (Parallel) simulation subsystem:



## Invitation: Please Join the National Instruments Circuit Design Community!

National Instruments would like to strongly encourage you to enhance these samples, develop new and innovative power electronics control blocks for LabVIEW FPGA and develop more advanced power electronics simulation in Multisim. When you do, please and share your work online with other members of the power electronics developer community via the NI Circuit Design Community website.




To get you thinking, here are some suggested areas for innovation.

**Multisim Simulation Components**

- Enhanced transistor models for high voltage, high current IGBTs, IGCTs, and other power transistors
  - Models that capture power transistor behavior at breakdown, in high frequency operation, or to failure
  - Models that capture dynamic thermal behavior
  - Models that estimate component wear-and-tear and lifetime
- Models for energy storage and conversion devices such as lithium ion batteries, lead acid batteries and fuel cells
- Renewable energy systems such as wind turbines, photovoltaic solar arrays, etc.

- Enhanced models for three-phase transformers
- Models for substation power gear
- Models for smart grid systems and components
- Enhanced models for power transmission and distribution lines
- Models for switched reluctance motors
- Hybrid and plug-in electric vehicles
- Magnetic levitation systems
- Linear actuator models
- Your idea here!

**LabVIEW FPGA IP Cores**

- Control algorithms designed to be compliant with grid-tied electrical power standards such as IEEE 1547 (Standard for Interconnecting Distributed Resources with Electric Power Systems) and UL 1741 (Inverters, Converters, Controllers and Interconnection System Equipment for Use With Distributed Energy Resources)
- Multilevel inverter control algorithms
- Control algorithms that enhance energy efficiency
- Control algorithms that reduce harmonic noise
- Control algorithms that extend IGBT lifetimes by minimizing thermal cycling
- Direct torque control algorithms
- Sensorless control algorithms
- Optimal and adaptive control algorithms
- Your idea here!

**Training Material, Samples, Examples and Proof of Concept Demonstration**

- Professional or academic training content and curriculum
- Experimental comparisons between simulation results and experimental data
- Samples that demonstrate coupled plant model dynamics on both the LabVIEW and Multisim sides showing the benefits of coordinated variable timestep solvers
- Samples that demonstrate control algorithms on both the LabVIEW and Multisim sides
- Examples of model refinement, parameter estimation or system identification using physical test data
- Design optimization examples- tweak both circuit and control design parameters simultaneously to produce an optimal system design
- Autotuning PID examples

- Buck-boost converter samples
- Your idea here!

**Improve this Guide!**

- What questions did you have when using the tools that aren't covered in this guide? Did you find a mistake? Do you have ideas for improving and extending it? Please share your comments and suggestions to help us improve it for future readers.
- Interested in editing or adding to this guide? For an editable copy of this document in Microsoft Word format, please email NI Clean Energy Product Manager, Brian.MacCleery@ni.com, or call him at 512-683-8759.

On behalf of NI, thank you for being a co-simulation beta tester and helping us improve the graphical system design tool chain for power electronics! Don't hesitate to drop us a line and share your thoughts.

Best Regards,
Brian MacCleery

brian.maccleery@ni.com | Principal Product Manager for Clean Energy Technology | National Instruments | 512.683.8759 |
http://www.ni.com/greenengineering | http://www.ni.com/smartgridwebcastseries

**Learn about the NI Green Engineering Grant Program**


**Find out more and apply at ni.com/greengrant**