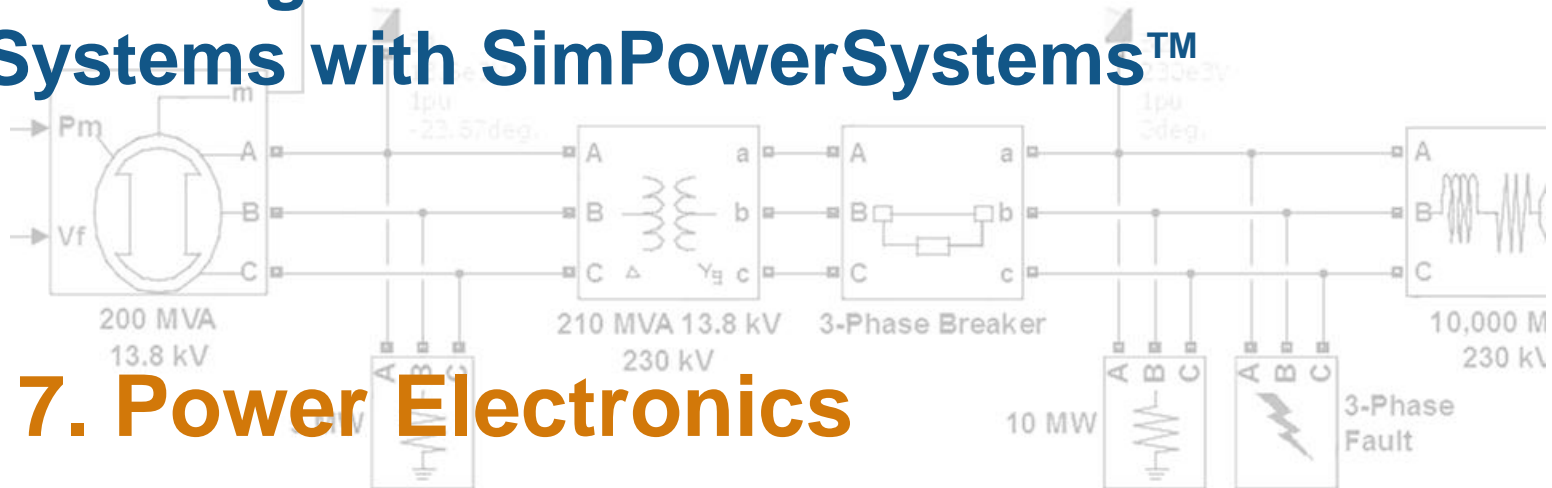


SimPowerSystems Hands-on Workshop: Modeling and Simulation of Electrical Power Systems with SimPowerSystems™



7. Power Electronics



Carlos Osorio

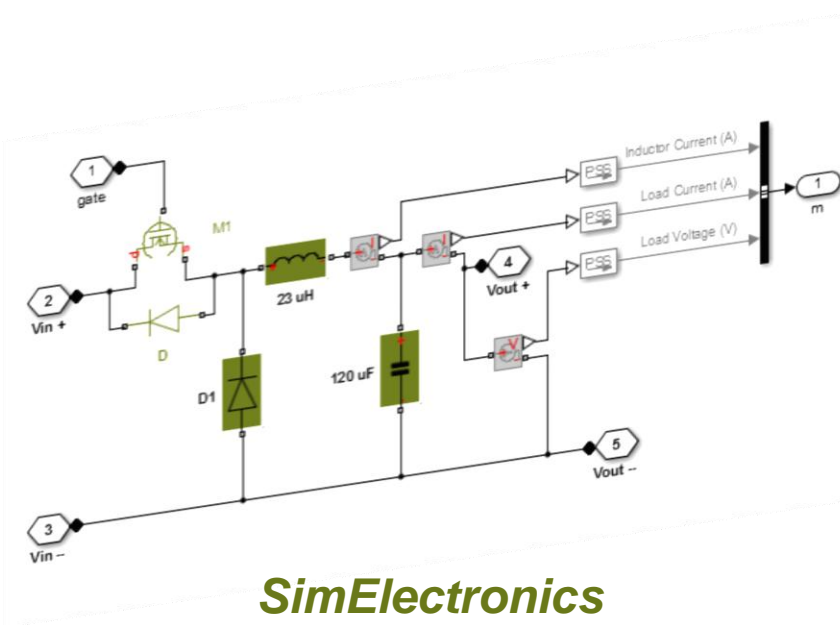
Principal Application Engineer

MathWorks – Natick, MA

Outline

- SimElectronics or SimPowerSystems?
- Ideal switching algorithm
- Power quality and harmonic analysis
- Control design and linearization

SimElectronics or SimPowerSystems?



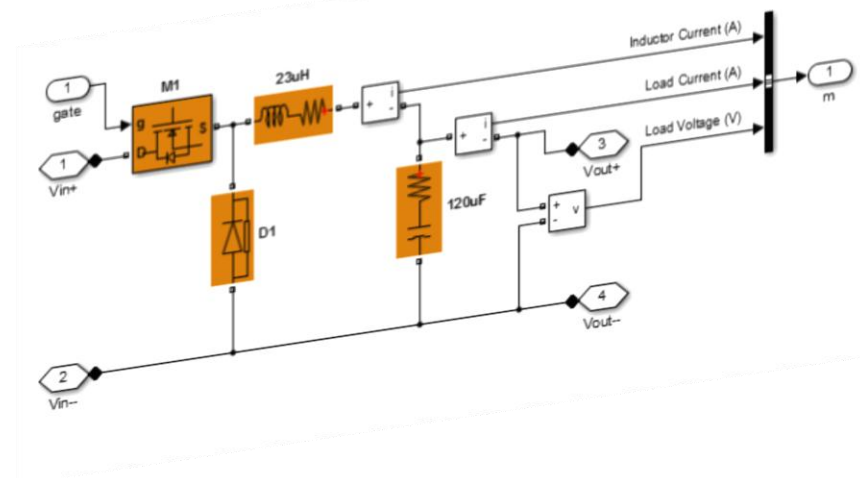
SimElectronics

Simultaneous nonlinear equations solution
 SPICE level switching device models
 Include switching losses
 Include parasitic current effects
 Include temperature effects
 Higher fidelity simulation

```
>> se_dcdcbuckconverter
```

SimPowerSystems

Piecewise linear systems solution
 Multiphase bridges and pulse generators
 Detailed and average voltage models
 Transient and harmonic analysis
 Faster simulation



```
>> sps_dcdcbuckconverter
```

SimElectronics or SimPowerSystems?

SimElectronics

SimPowerSystems

SimElectronics Semiconductor Devices

SimPowerSystems Specialized Technology Power Electronics

The collector and base currents are [1]:

$$I_C = -IS \left[\left(e^{-qV_{BE}/(kT_{m1})} - e^{-qV_{BC}/(kT_{m1})} \right) \left(1 + \frac{V_{BC}}{V_A} \right) - \frac{1}{\beta_R} \left(e^{-qV_{BC}/(kT_{m1})} - 1 \right) \right]$$

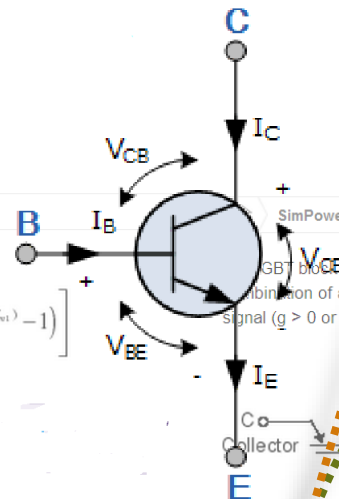
$$I_B = -IS \left[\frac{1}{\beta_F} \left(e^{-qV_{BE}/(kT_{m1})} - 1 \right) + \frac{1}{\beta_R} \left(e^{-qV_{BC}/(kT_{m1})} - 1 \right) \right]$$

Where:

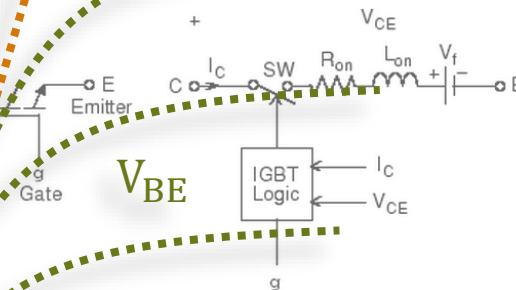
- I_B and I_C are base and collector currents, defined as positive into the device.
- IS is the saturation current.
- V_{BE} is the base-emitter voltage and V_{BC} is the base-collector voltage.
- β_F is the ideal maximum current gain BF
- β_R is the ideal maximum current gain BR
- V_A is the forward Early voltage VAF
- q is the elementary charge on an electron ($1.6021766 \times 10^{-19}$ Coulombs).
- k is the Boltzmann constant ($1.3806503 \times 10^{-23}$ J/K).
- T_{m1} is the transistor temperature, as defined by the Measurement temperature parameter value.

I_{CE}

V_{CE}



IGBT model implements a semiconductor device controllable by the gate signal. The IGBT is simulated as a series combination of a resistor R_{on} , inductor L_{on} and a DC voltage source V_f in series with a switch controlled by a logical signal ($g > 0$ or $g = 0$).



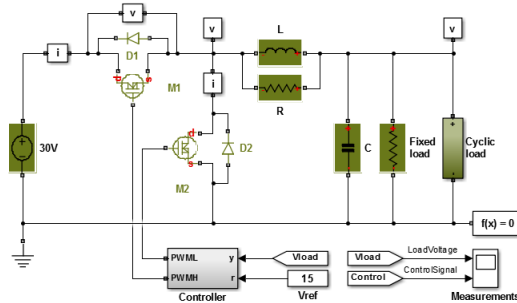
The IGBT turns on when the collector-emitter voltage is positive and greater than V_f and a positive signal is applied at the gate input ($g > 0$). It turns off when the collector-emitter voltage is positive and a 0 signal is applied at the gate input ($g = 0$).

The IGBT device is in the off state when the collector-emitter voltage is negative. Note that many commercial IGBTs do not have the reverse blocking capability. Therefore, they are usually used with an antiparallel diode.

SimElectronics or SimPowerSystems?

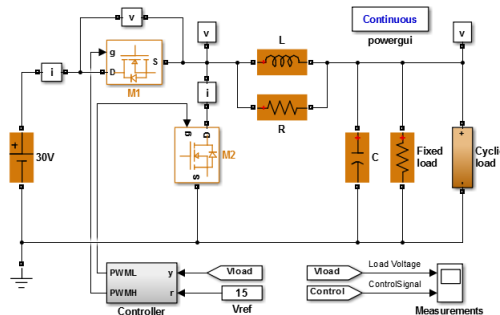
SimElectronics

Synchronous Buck Converter

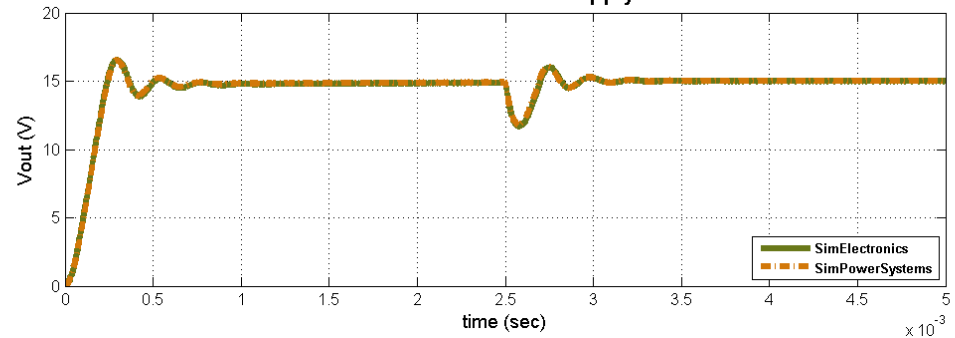


SimPowerSystems

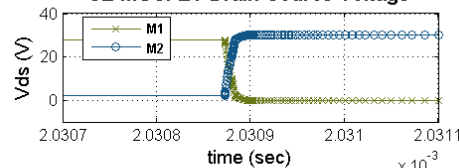
Synchronous Buck Converter



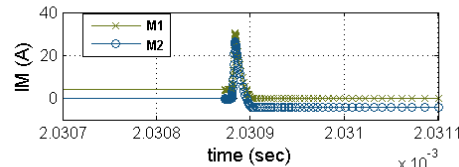
Switched Power Supply



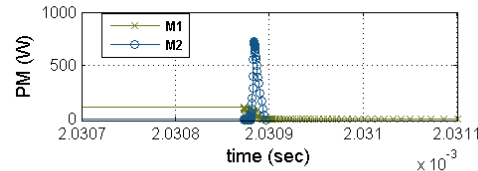
SE MOSFET Drain-Source Voltage



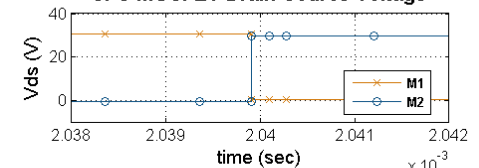
SE MOSFET Drain-Source Current



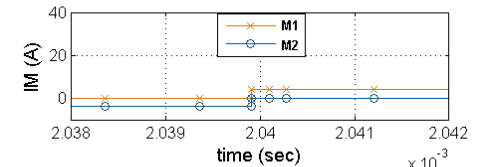
SE MOSFET Power



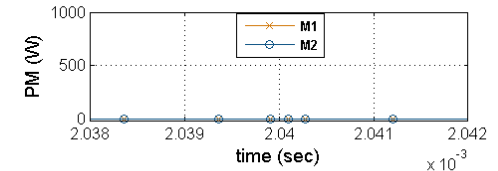
SPS MOSFET Drain-Source Voltage



SPS MOSFET Drain-Source Current



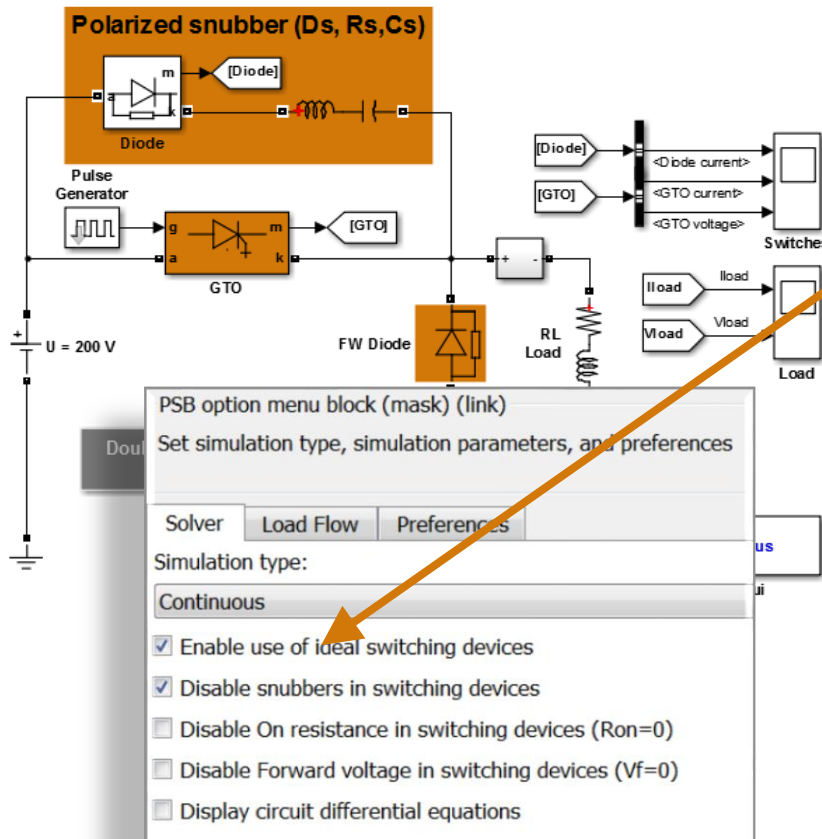
SPS MOSFET Power



```
>> edit compare_powersupply
```

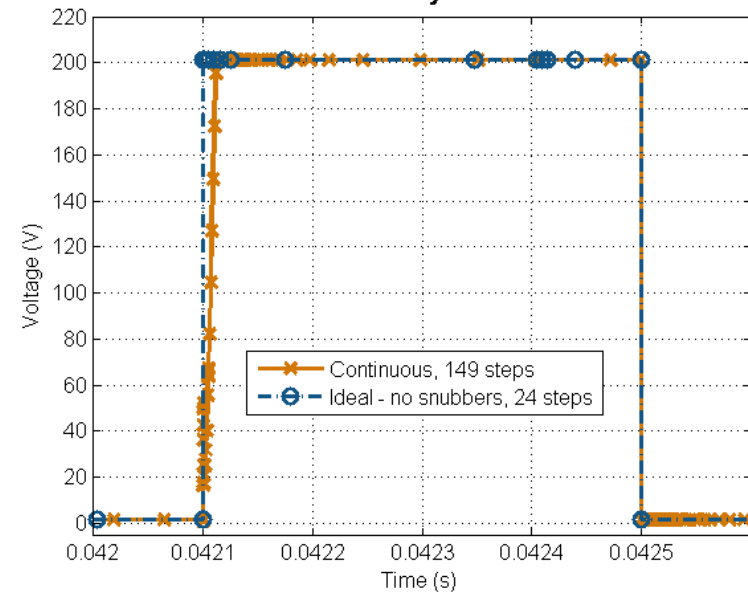
Ideal switching algorithm

GTO Buck Converter



The ideal switching method can be selected from the solver configuration parameters tab when in Continuous simulation mode

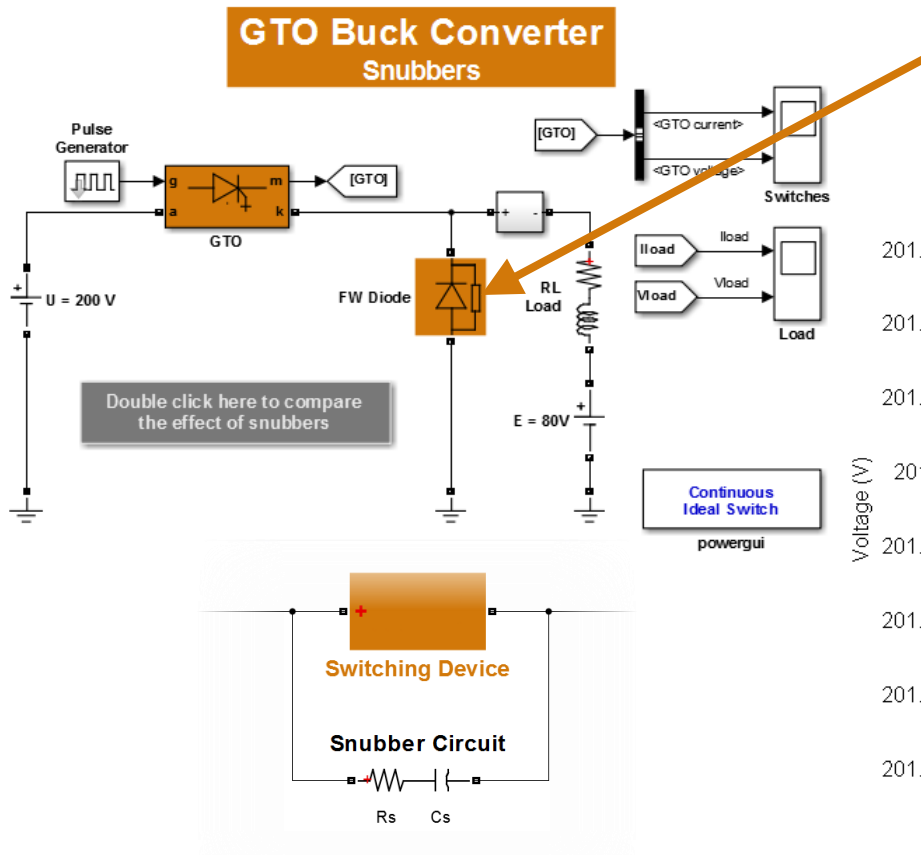
GTO Thyristor



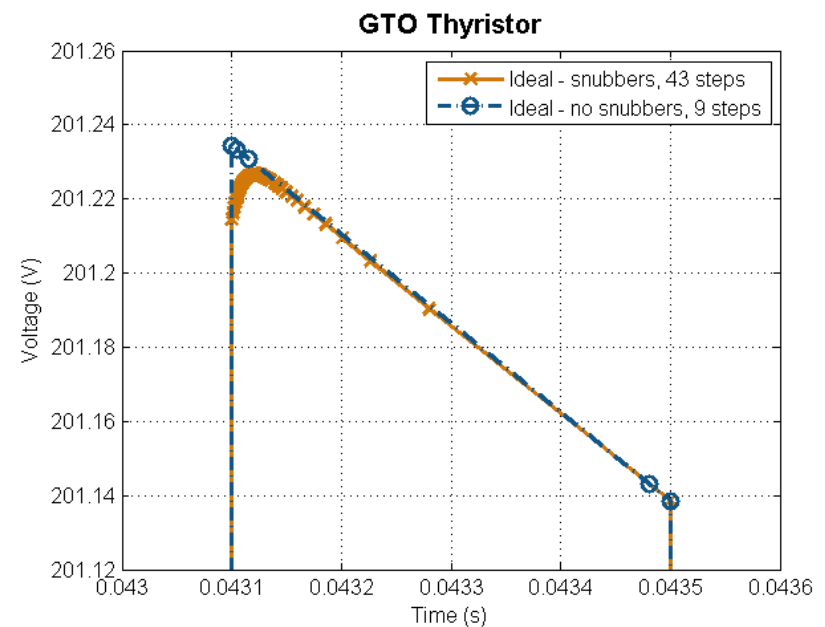
```
>> edit compare_idealswitching
```

Ideal switching algorithm

Effect of using snubber circuits



Snubbers provide an alternate path for discharging inductive elements and protecting switching devices in high power systems

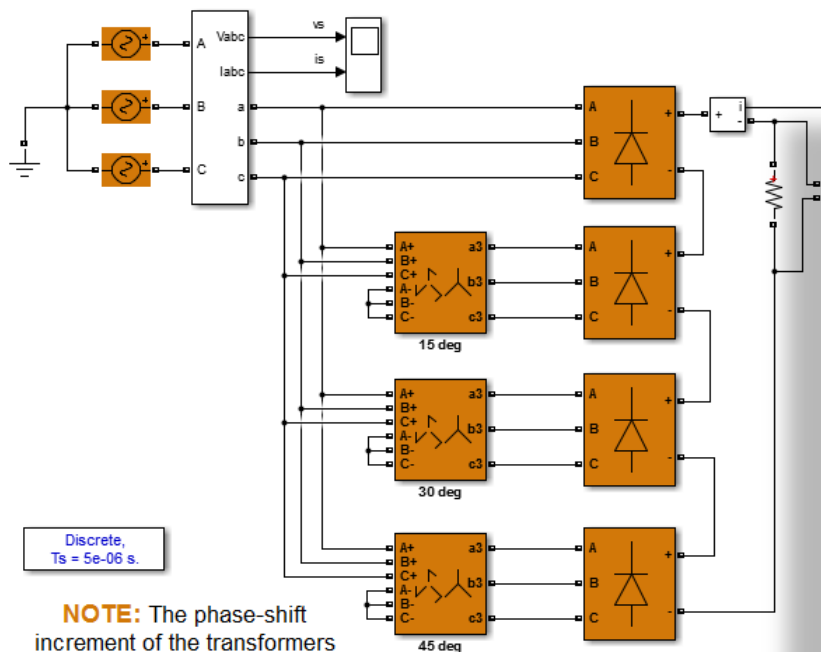


```
>> edit compare_snubbers
```

Power quality and harmonic analysis

Multi-pulse rectification

24 Pulse Diode Rectifier

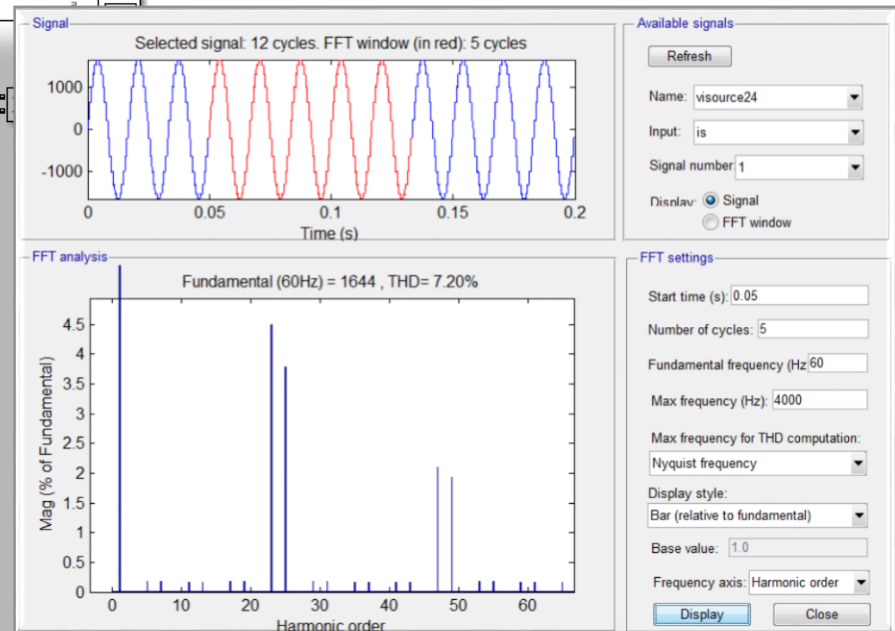


Discrete,
 $T_s = 5e-06$ s.

NOTE: The phase-shift increment of the transformers is calculated as:
 ϕ (deg) = $360 / \text{pulse number}$

powergui

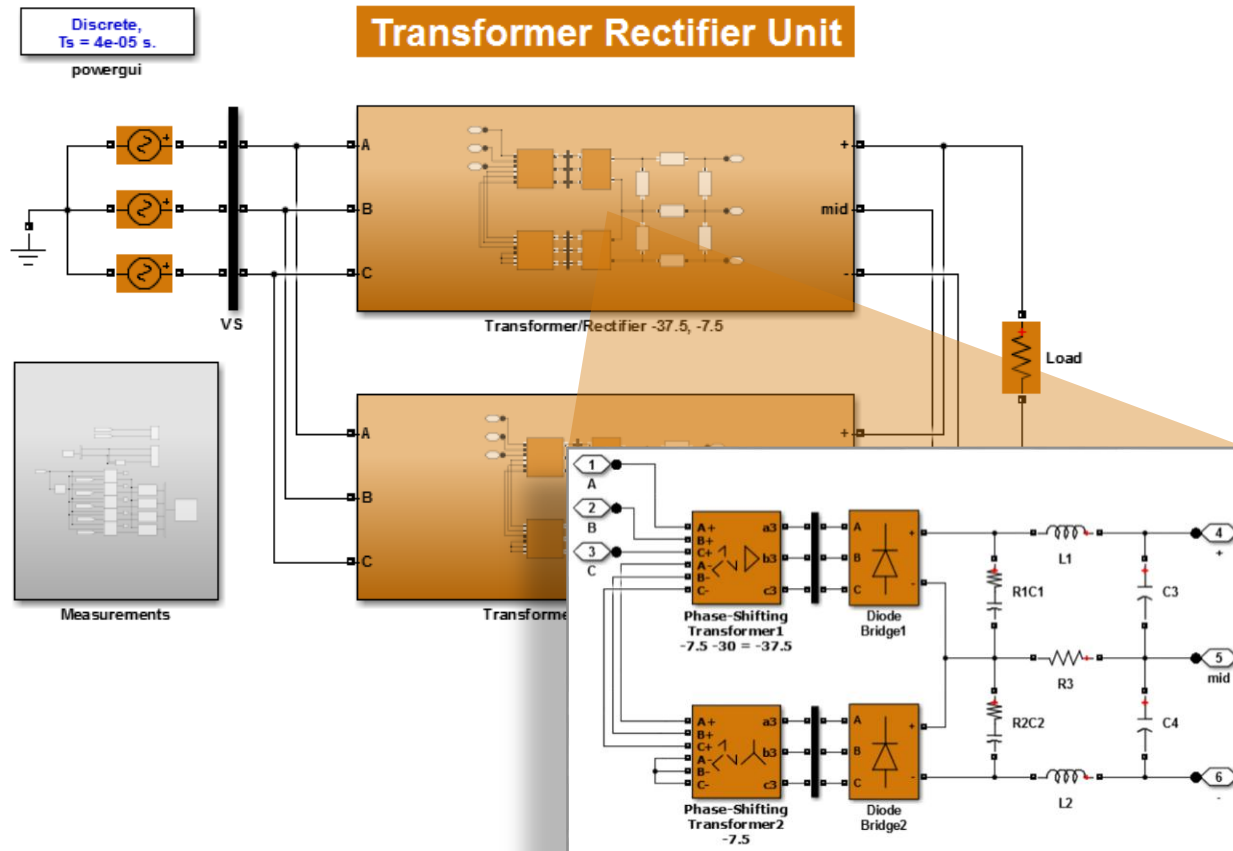
FFT Analysis



```
>> rectifier_6n12pulse
>> rectifier_24pulse
```


Power quality and harmonic analysis

Transformer rectifier units (TRUs)

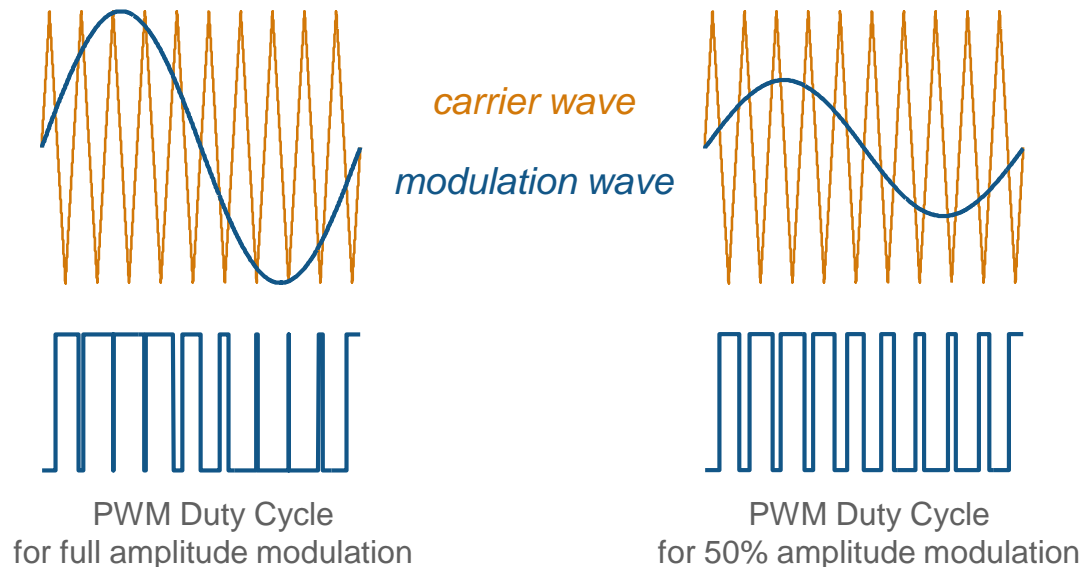


```
>> transformer_rectifier_unit
```

Power quality and harmonic analysis

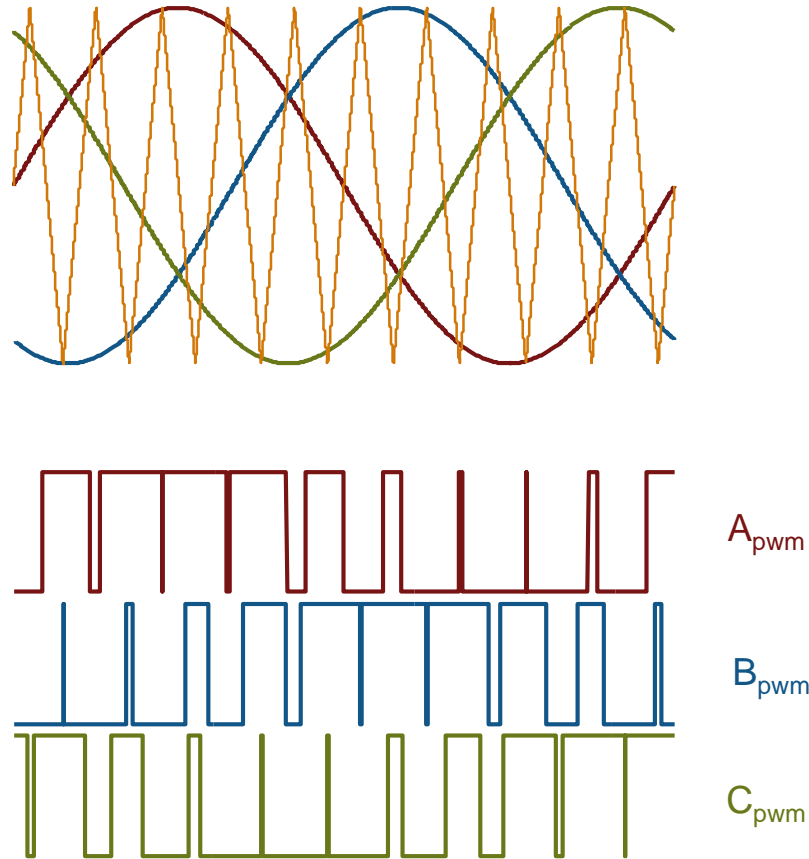
Switching strategies – Pulse-Width Modulation (PWM)

- The aim of PWM is to construct a high frequency carrier wave which contains an underlying, lower frequency modulation wave
- The modulation wave is the waveform that you want the electrical system to ‘see’
- Higher order harmonics will depend on the ratio of the carrier wave frequency and the modulation wave frequency



Power quality and harmonic analysis

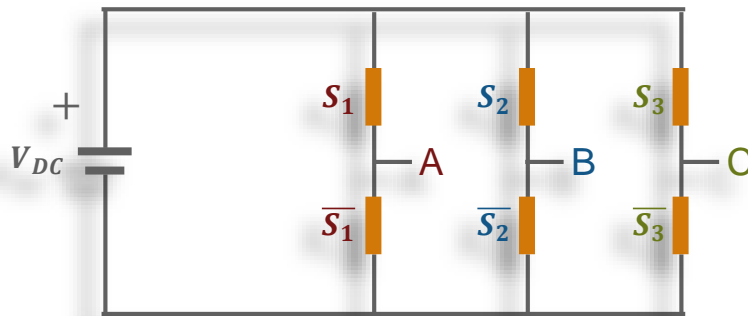
Switching strategies – Pulse-Width Modulation (PWM)



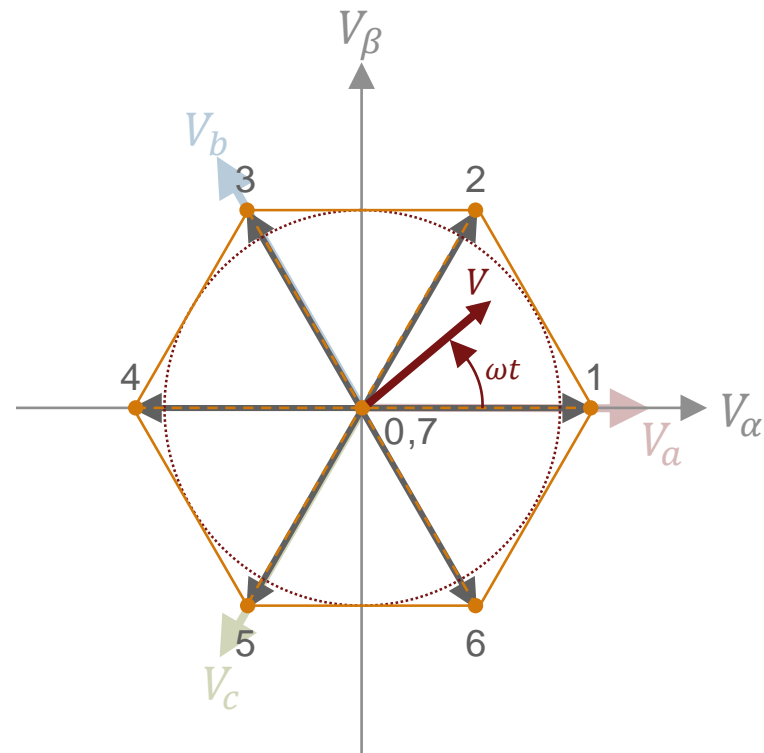
Power quality and harmonic analysis

Switching strategies – Space Vector Modulation (SVM)

- The aim of SVM is to approximate the reference voltage vector instantaneously by combining the switching states corresponding to eight basic vectors

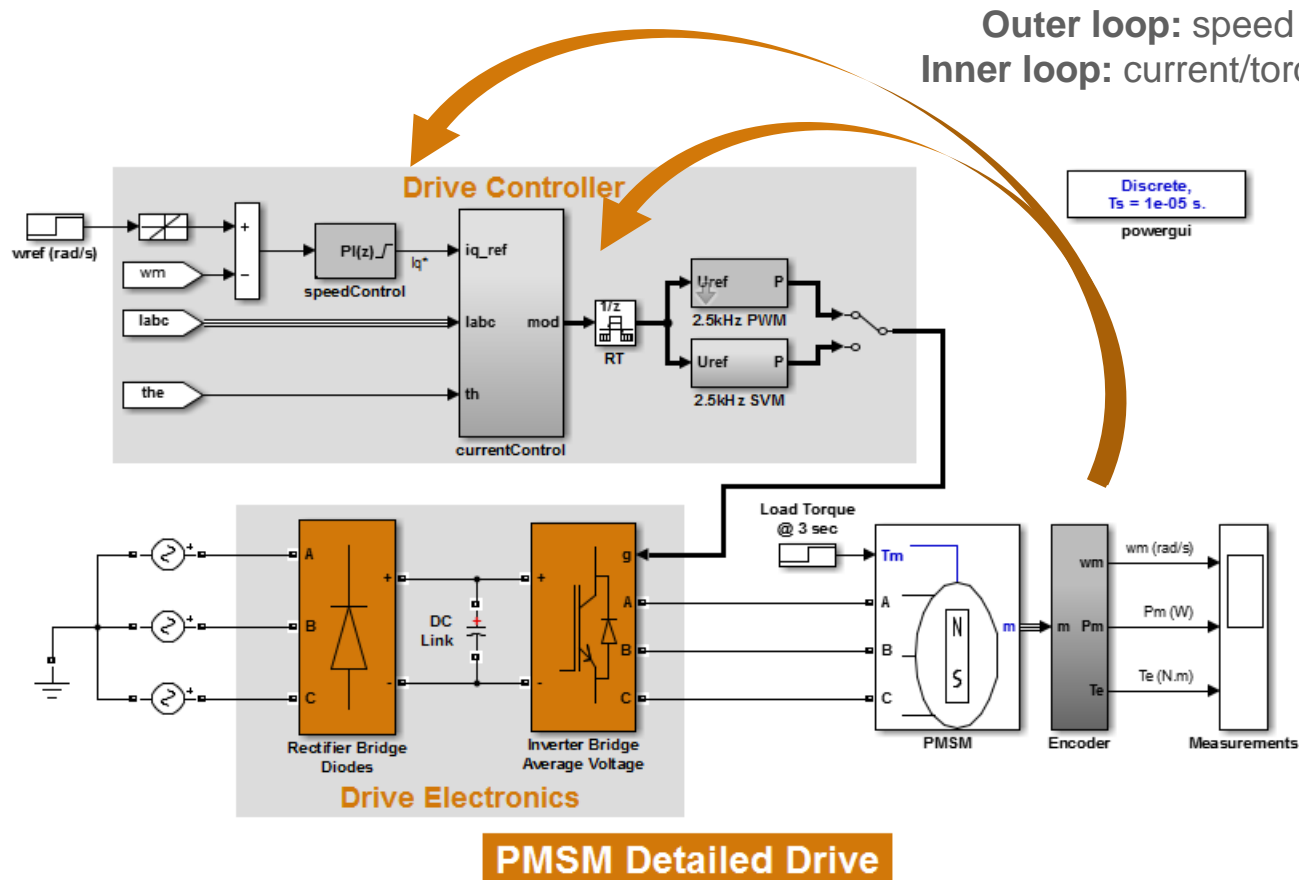


	0	1	2	3	4	5	6	7
s_1	0	1	1	0	0	0	1	1
s_2	0	0	1	1	1	0	0	1
s_3	0	0	0	0	1	1	1	1



Power quality and harmonic analysis

Comparing switching strategies

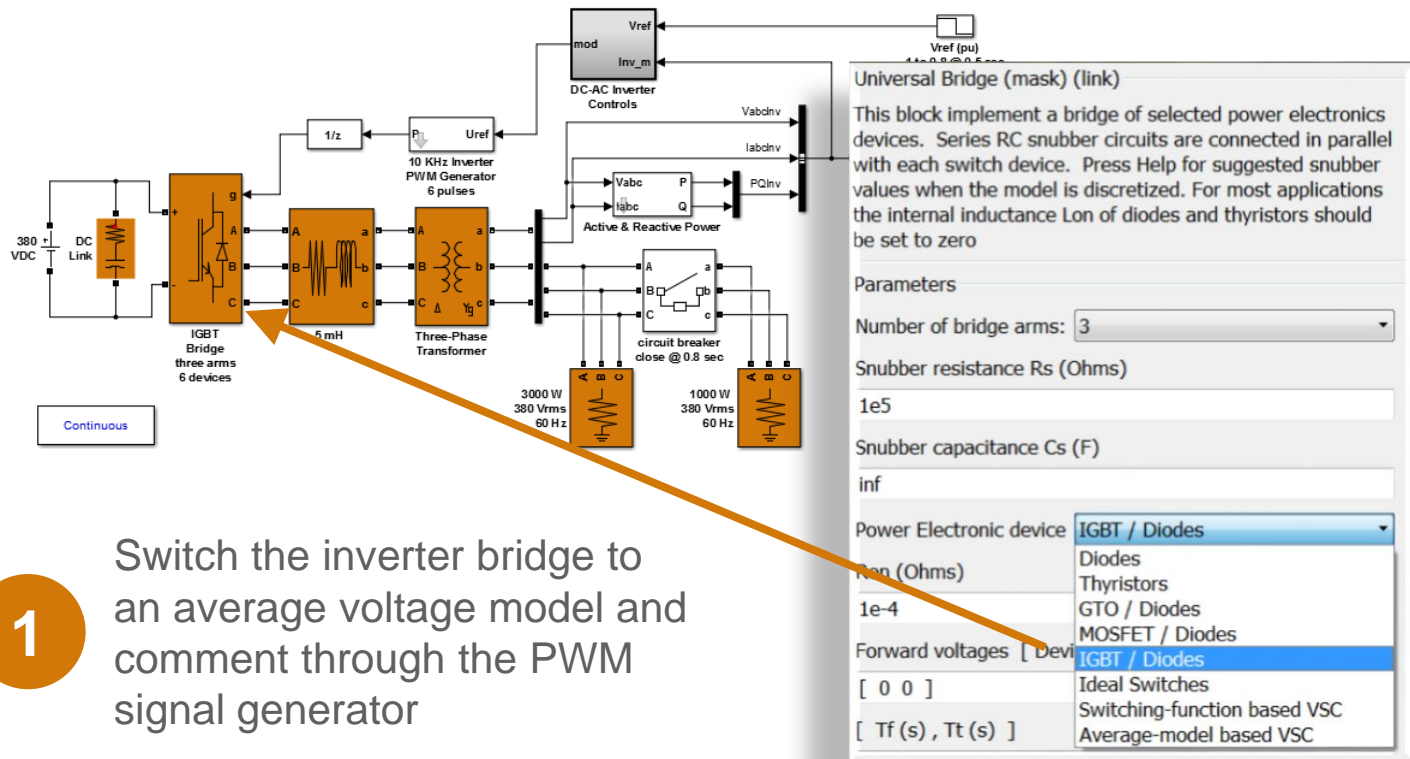


```
>> pmsm_detailed_drive
```

Control design and linearization

Detailed and average transistor bridges

DC-AC Half Bridge Inverter - Detailed Model



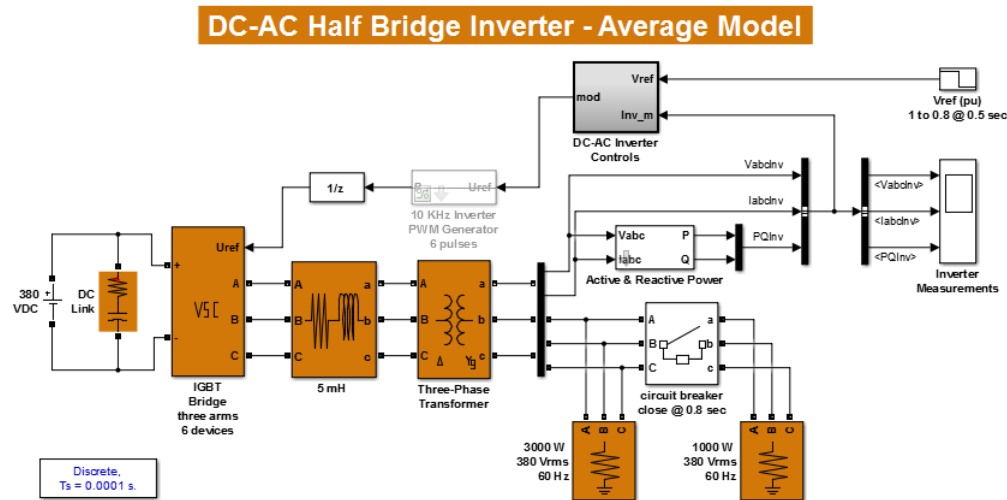
1

Switch the inverter bridge to an average voltage model and comment through the PWM signal generator

```
>> igbt_power_inverter_det
```

Control design and linearization

Detailed and average transistor bridges



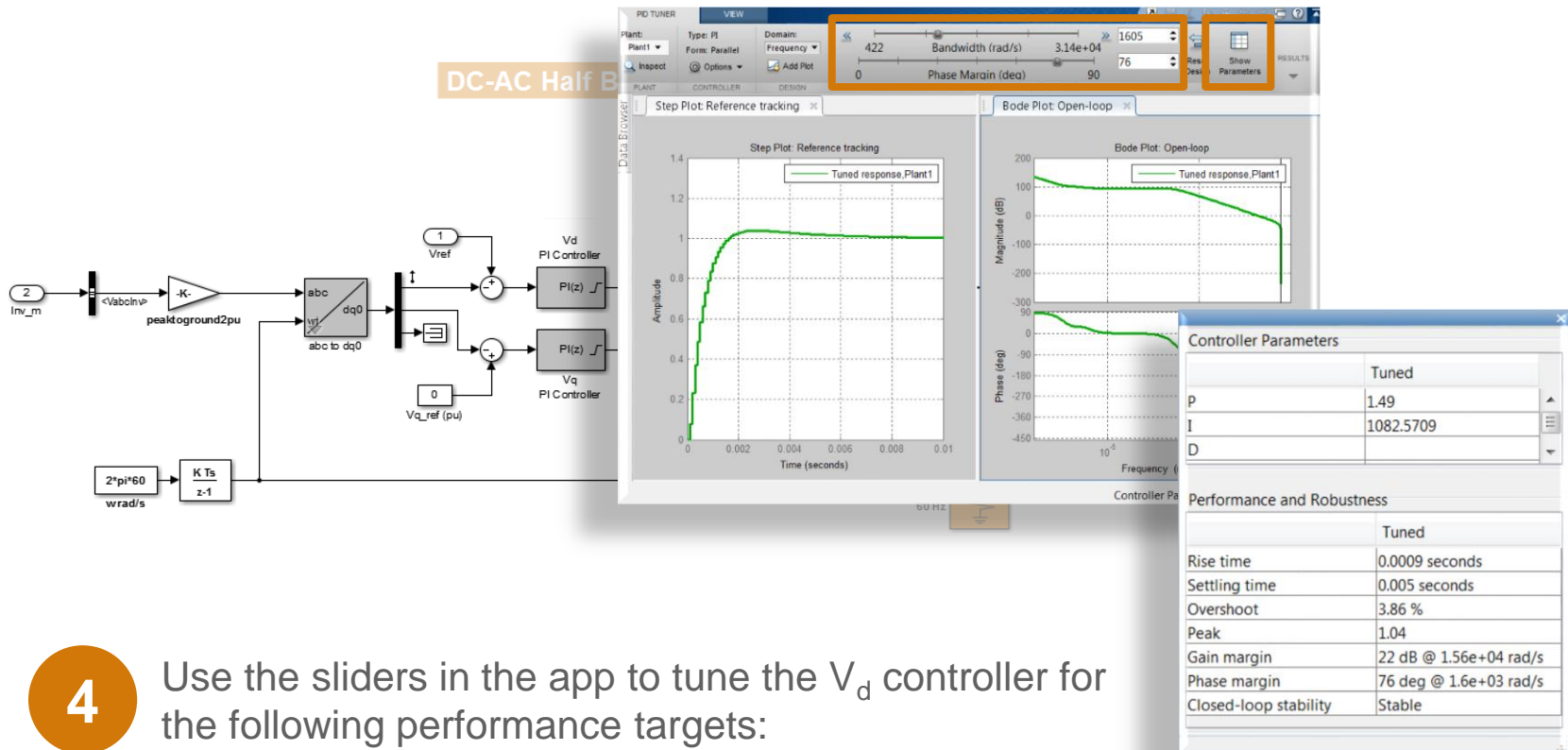
- 2 Adjust the simulation mode and sampling rates appropriately
Notice that the controller runs at a 10 kHz rate

```
>> igbt_power_inverter_avg
```



Control design and linearization

Detailed and average transistor bridges



4

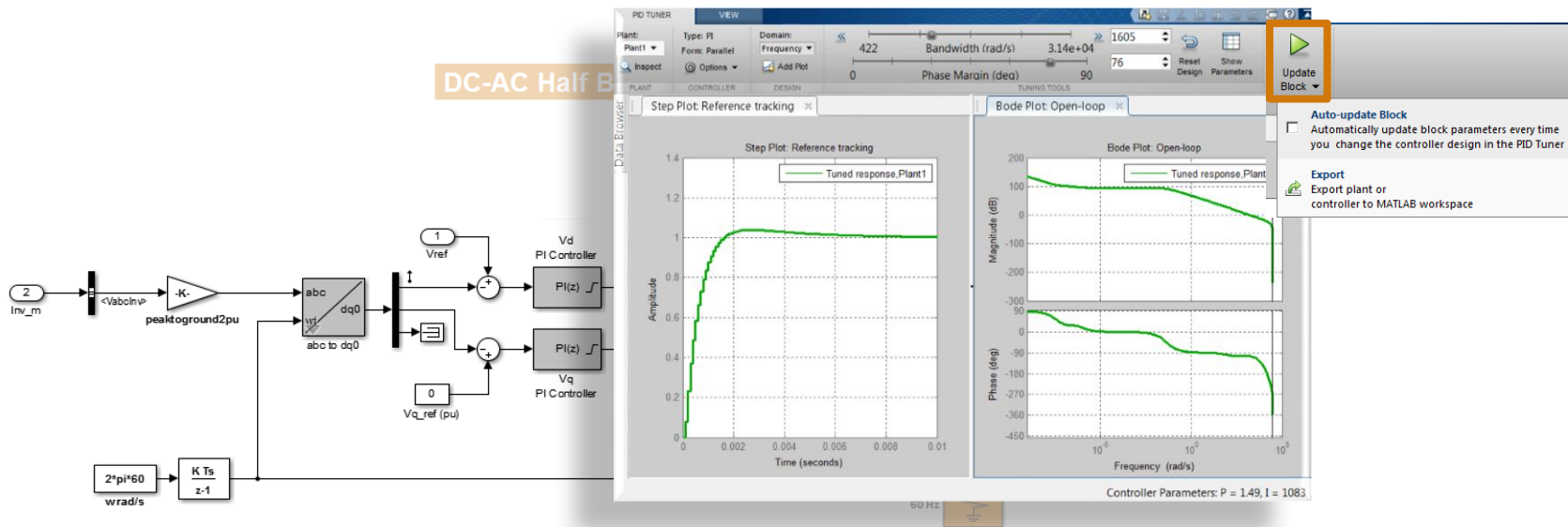
Use the sliders in the app to tune the V_d controller for the following performance targets:

- bandwidth > 200 Hz
- settling time < 5 milliseconds
- overshoot < 5%

```
>> igbt_power_inverter_avg
```

Control design and linearization

Detailed and average transistor bridges



5

Click **Update Block** and verify the performance of the controller against the detailed switching inverter model in simulation

For simplicity, we will use the same newly computed PI gains for the V_q controller

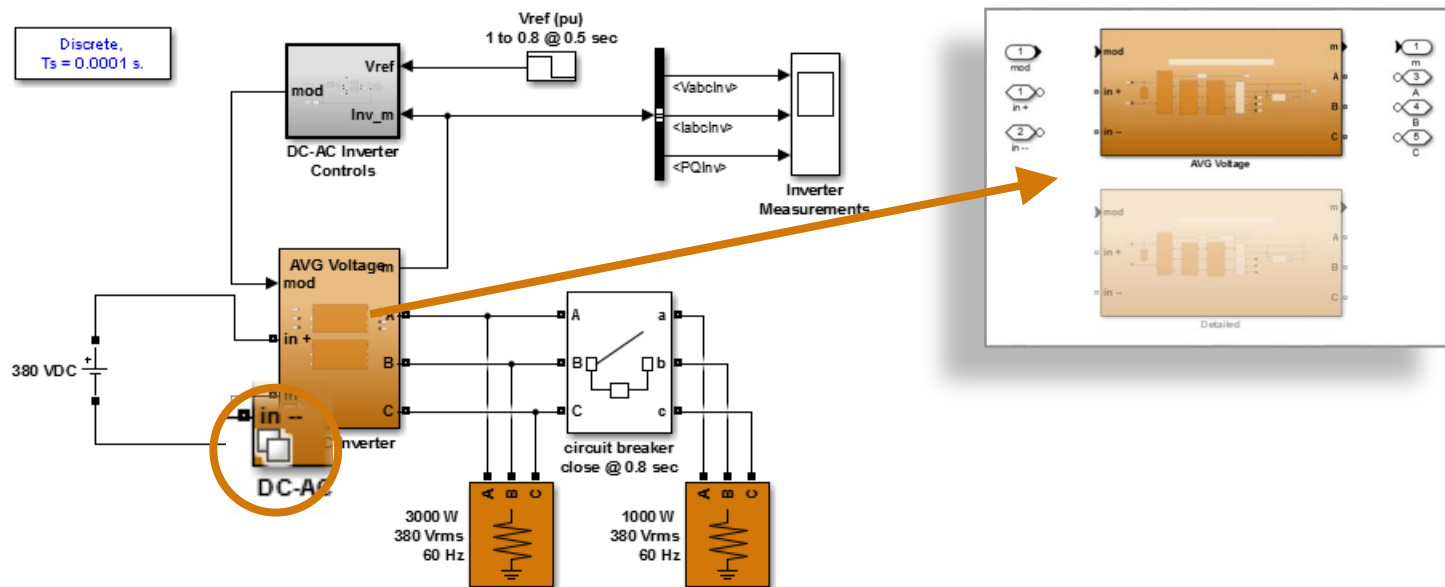
```
>> igt_power_inverter_avg
```

Control design and linearization

Detailed and average transistor bridges

6

Combine both versions of the inverter in a single model using a **Variant Subsystem**
Subsystem & Model Reference → Convert Subsystem to...



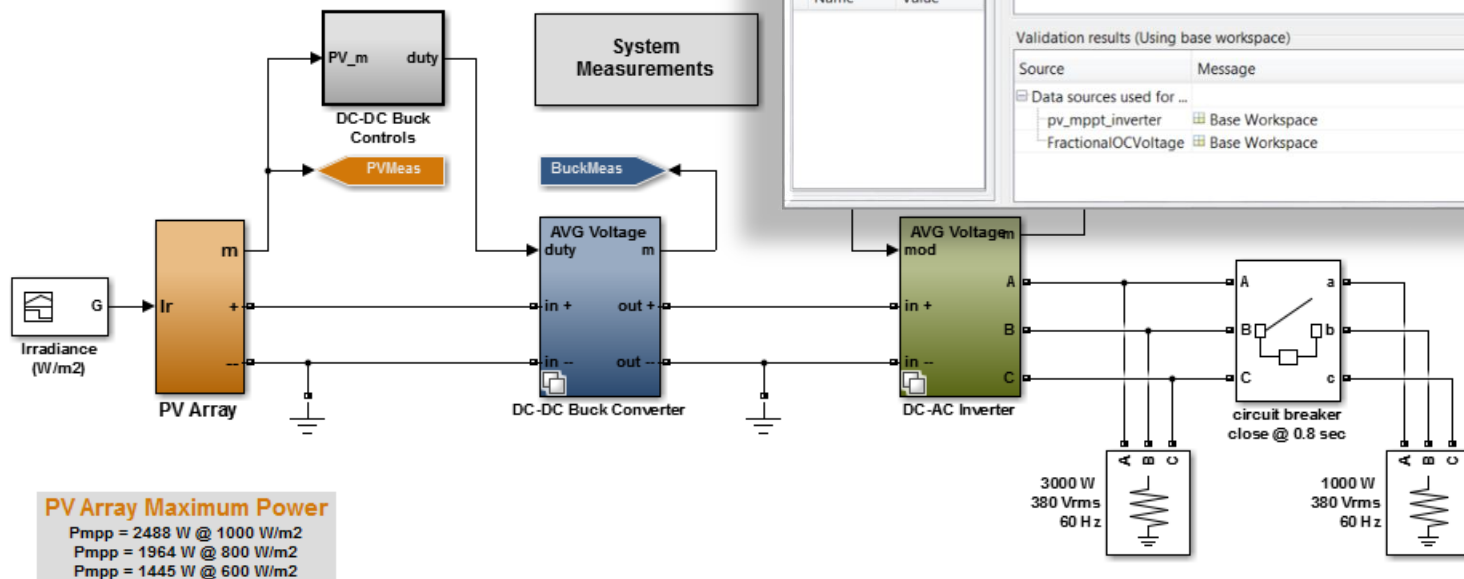
```
>> igbt_power_inverter
```

Control design and linearization

Using subsystem variants

Use the **Variant Manager** to set multiple variant configurations for simulation and testing
 Variant → Open in Variant Manager

Continuous



```
>> pv_mppt_inverter
```

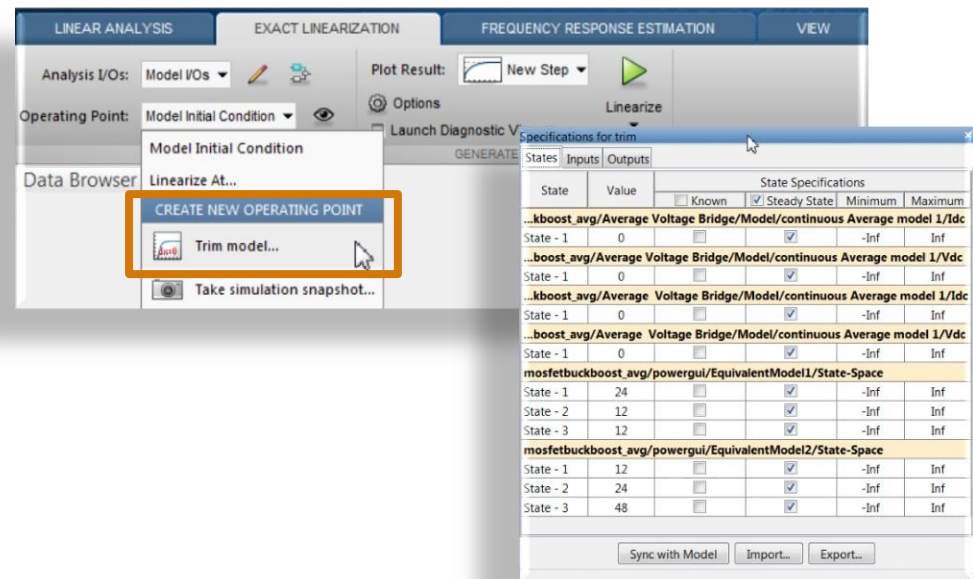
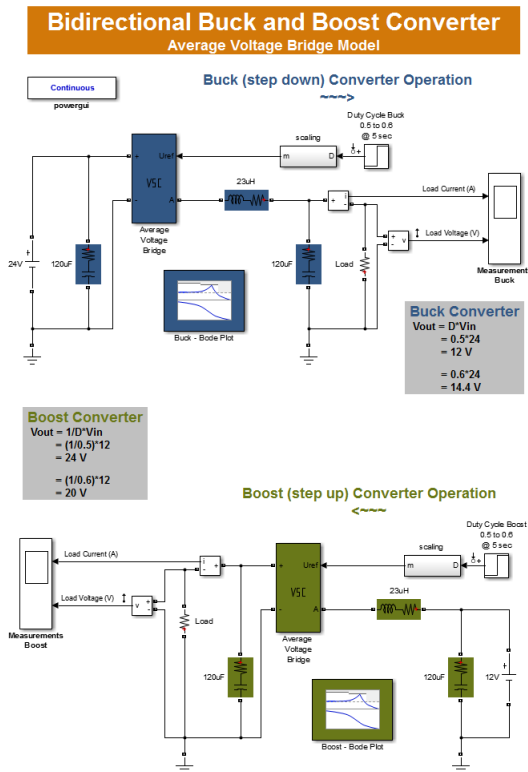
Control design and linearization

Initialization and trimming

1

Use the **Trim model** option in the Linear Analysis tool

Analysis → Control Design → Linear Analysis...



2

Check the **Specifications** for the output voltages as **known** and trim the model

Change optimization method to **nonlinear least squares**

```
>> mosfetbuckboost_avg
```

Control design and linearization

Initialization and trimming

3

Verify trimmed operating condition and initialize model

Edit: op_trim2

Optimizer Output Details

State Input Output

State	Desired Value	Actual Value	Desired dx	Actual dx
...uckboost_avg/Average Voltage Bridge/Model/continuous Average model 1/Idc				
State - 1	[-Inf , Inf]	-6e-07	0	0
...uckboost_avg/Average Voltage Bridge/Model/continuous Average model 1/Vdc				
State - 1	[-Inf , Inf]	2.4024e-06	0	-0.024
...uckboost_avg/Average Voltage Bridge/Model/continuous Average model 1/Idc				
State - 1	[-Inf , Inf]	2.4e-06	0	-0.00038365
...ckboost_avg/Average Voltage Bridge/Model/continuous Average model 1/Vdc				
State - 1	[-Inf , Inf]	2.3904e-06	0	0.095998
mosfetbuckboost_avg/powergui/EquivalentModel1/State-Space				
State - 1	[-Inf , Inf]	24	0	0
State - 2	12	12	0	-3.459e-08
State - 3	[-Inf , Inf]	-12	0	1.104e-06
mosfetbuckboost_avg/powergui/EquivalentModel2/State-Space				
State - 1	[-Inf , Inf]	12	0	0
State - 2	24	24	0	-1.0384e-07
State - 3	[-Inf , Inf]	47.9992	0	-4.105e-06

Initialize model...

Select:

- Solver
- Data Import/Export
- Optimization
- Diagnostics
- Hardware Implementation
- Model Referencing
- Simulation Target
- Code Generation
- HDL Code Generation
- Simscape
- SimMechanics 1G
- SimMechanics 2G

Load from workspace

☒ Input: getinputstruct(op_trim1)

☒ Initial state: getstatestruct(op_trim1)

Save to workspace

Time, State, Output

☐ Time: tout

☐ States: xout

☐ Output: yout

☐ Final states: xFinal

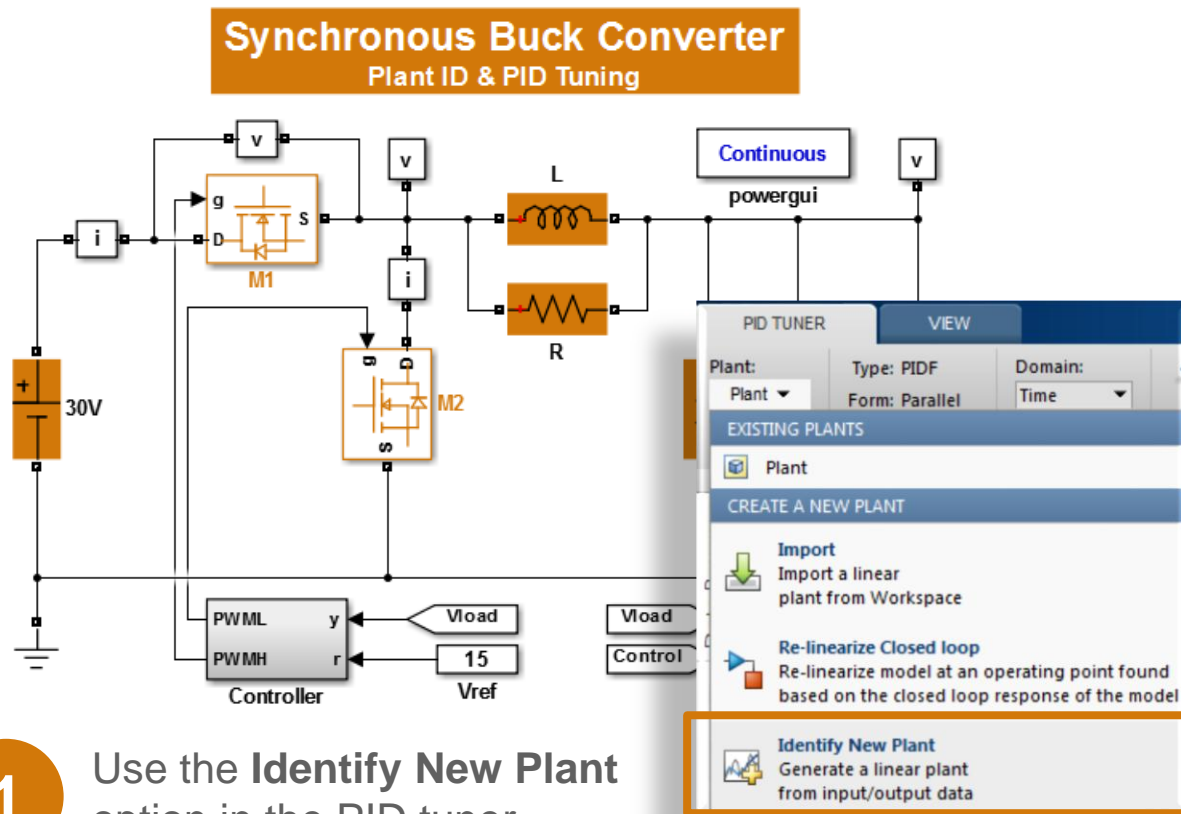
4

Run simulation with the computed operating condition object as initial states

```
>> mosfetbuckboost_avg
```

Control design and linearization

Plant identification from simulation data



```
>> sps_powersupply_sid
```


Control design and linearization

Plant identification from simulation data

3 Set up a Step Response test with the following characteristics:

- Sample Time = 5×10^{-6}
- Offset = 0.5
- Onset Lag = 0.003
- Amplitude = 0.4

2 Choose **Simulate Data** to obtain I/O results from the simulation model

```
>> sps_powersupply_sid
```

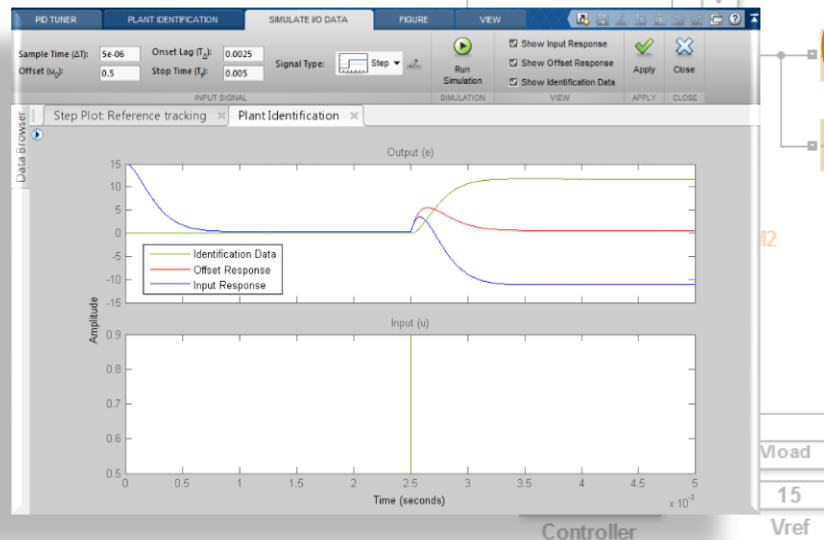

Control design and linearization

Plant identification from simulation data

4

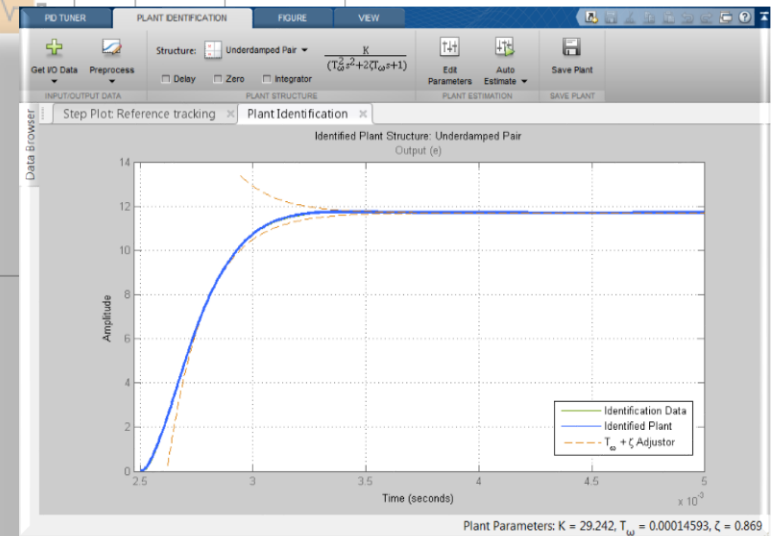
Run the simulation to obtain the identification data

Synchronous Buck Converter



5

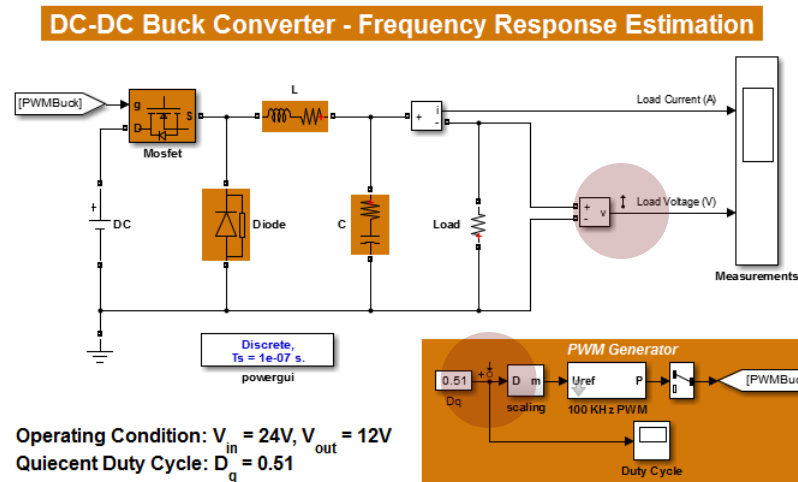
Select the model structure and use the interactive handles to match the model to the data



```
>> sps_powersupply_sid
```

Control design and linearization

Frequency response estimation



1

Open the **Frequency Response Estimation** tool

Analysis → Control Design → Frequency...

2

Create a new input **Fixed Sample Time Sinestream** with $T_s = 1e-7$

Use the pre-computed analytic transfer function of the converter **GDBuck** as a reference for selecting the test frequencies and increase the amplitude of the test sine waves to **1e-1**

```
>> mosfetbuck_fre
```

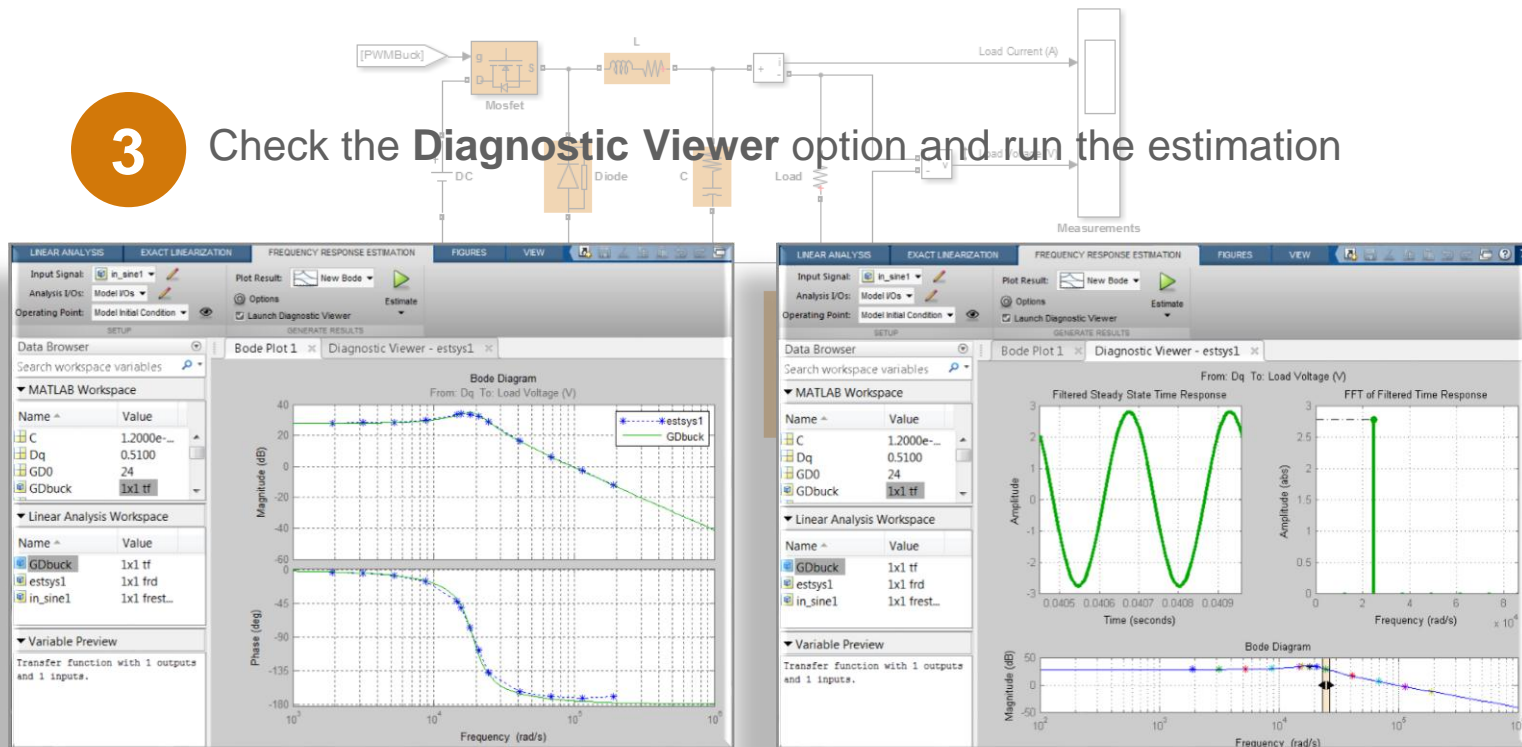
Control design and linearization

Frequency response estimation

DC-DC Buck Converter - Frequency Response Estimation

3

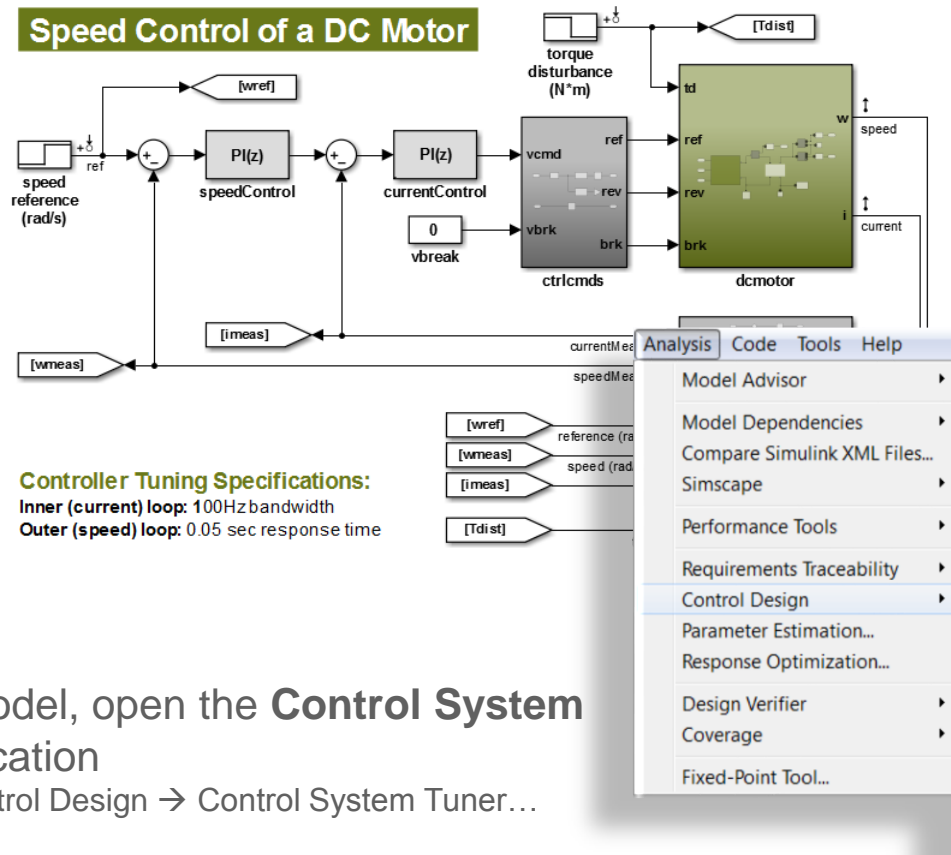
Check the **Diagnostic Viewer** option and run the estimation



```
>> mosfetbuck_fre
```

Control design and linearization

Multi-input, multi-output control tuning



- From the model, open the **Control System Tuner** application
 Analysis → Control Design → Control System Tuner...

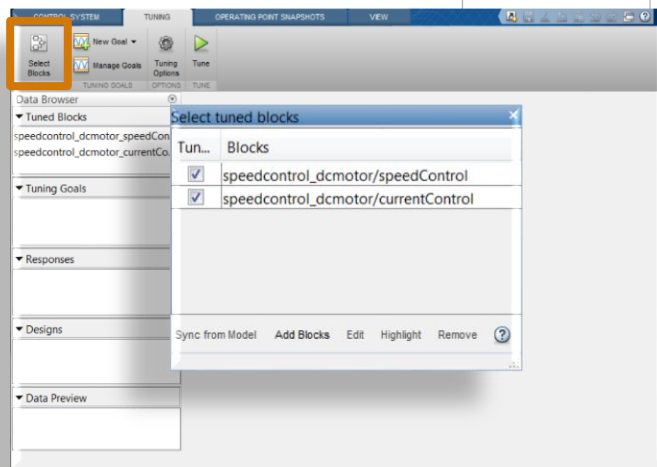
```
>> speedcontrol_dcmotor
```

Control design and linearization

Multi-input, multi-output control tuning

2

In the **Control System** tab, capture a new operating point by taking a simulation snapshot at 10 seconds

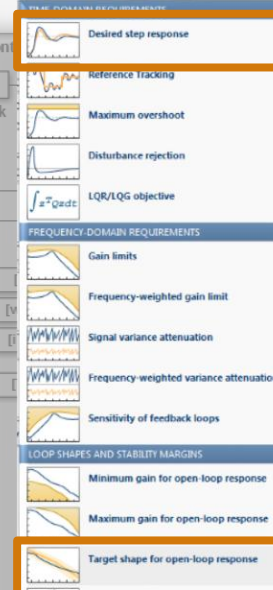


3

In the **Tuning** tab, select the controller blocks to be tuned

4

In the **Tuning** tab, select the two new desired tuning goals



Goal #1: Desired step response

Input: speed reference

Output: speed measurement

Time constant = 0.05 seconds

Goal #2: Target shape for open loop response

Response at current measurement

Open loop at the speed measurement

Crossover frequency = $100 \cdot 2 \cdot \pi$ rad/s

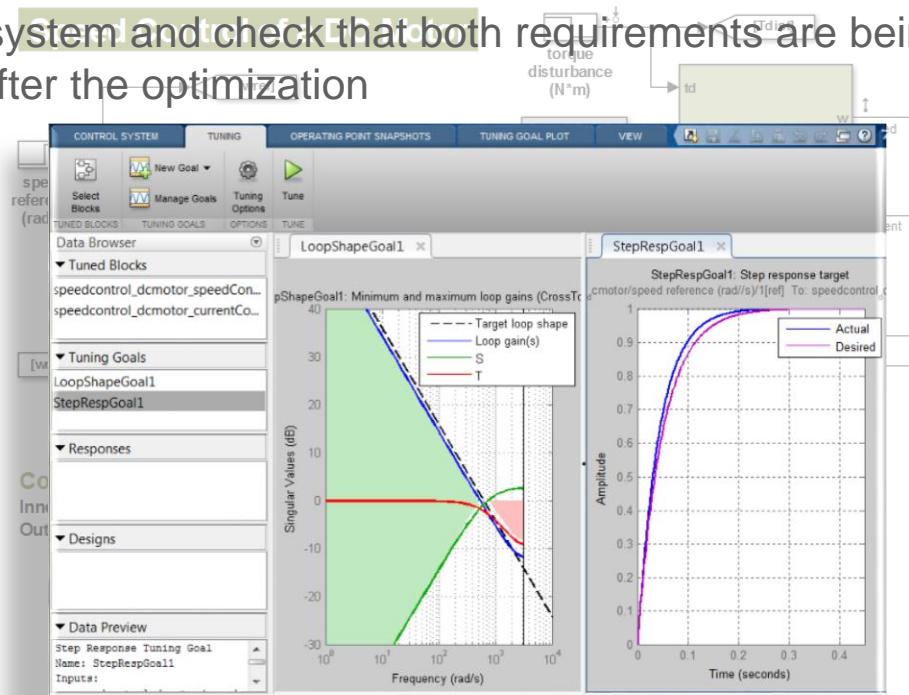
```
>> speedcontrol_dcmotor
```

Control design and linearization

Multi-input, multi-output control tuning

5

Tune the system and check that both requirements are being adequately satisfied after the optimization



6

In the **Control System** tab, select **Update Blocks** and verify the performance of the controller against the nonlinear model in simulation

```
>> speedcontrol_dcmotor
```

