Search

Advanced SIMPLIS Training

This section of the documentation comes directly from the training course that SIMPLIS Technologies conducts several times per year in different locations. For information on the next scheduled course, click on this link: Advanced SIMPLIS Training Course. The course material is intended for those with some experience using SIMPLIS. Those taking this course have completed the SIMPLIS Tutorial and are ready to move to an advanced level of proficiency. Experience shows that even advanced users of SIMPLIS can learn quite a bit from this course.

User Requirements for Advanced SIMPLIS Training

Participants are expected to arrive with:

1. The knowledge gained from the SIMPLIS Tutorial.
2. A computer loaded with SIMetrix/SIMPLIS version 8.20d or later. SIMPLIS will provide SIMetrix/SIMPLIS Pro w/ DVM licenses for the training session.

Topics in this module

Topics in this chapter

Advanced SIMPLIS Training

Search

Course Outline

Module 1 - Overview of the SIMPLIS Environment

Module 1 lays the groundwork for the entire course, by whetting the attendee's appetite through examples and exercises.

In the first section, a number of examples compare and contrast SIMPLIS with more common SPICE-based simulators. Models designed for SIMPLIS use Piecewise Linear (PWL) modeling, and the basics of PWL modeling and the accuracy of PWL models are discussed. The POP analysis, which is unique to SIMPLIS is introduced and the AC analysis on the time-domain PWL model is also covered. A brief introduction to the Design Verification Module (DVM) prepares users to automate model testing.

The third section describes how the user interface is constructed. In this section, you will install a few script files which will customize your SIMetrix/SIMPLIS user interface.

By the end of this module you will understand the basic framework on which all of the following modules are based.

To download the examples for Module 1, click Module_1_Examples.zip.

1.0 SIMPLIS Basics

1.1 Introduction to Design Verification Module (DVM)

1.2 The SIMetrix/SIMPLIS User Interface

Module 2 - Advanced SIMPLIS

Module 2 expands on the Module 1 topics, covering the SIMPLIS analyses in greater detail.

The module starts with an in-depth coverage of the SIMPLIS transient analysis, followed by topics that deal with managing simulation output data files, advanced probing, including plotting frequency spectra and arbitrary functions of curves. The two ways that SIMPLIS back-annotates, or loads a previous circuit state, are described and then compared and contrasted. Finally, you will learn how the SIMPLIS POP analysis works in detail.

To download the examples for Module 2, click Module_2_Examples.zip.

2.0 Transient Analysis Settings

2.1 Initial Conditions and Back Annotation

2.2 How POP Really Works

2.3 Managing Simulation Data

2.4 Advanced Probing

Module 3 - Running SIMPLIS Simulations

Module 3 starts with a detailed look under the hood of a SIMPLIS simulation. After this module you will have a firm grasp on the run process, from pressing F9 to viewing the graph output. You will learn about the SIMetrix/SIMPLIS Netlist Preprocessor and how the individual run process steps affect parametrization and model creation. Next, you will learn how to run basic multi-step and Monte Carlo analyses on a model. Finally, how to load circuit templates with values from a text file is covered in detail.

To download the examples for Module 3, click Module_3_Examples.zip.

Module 4 - Introduction to Modeling

Module 4 represents a distinct shift away from how SIMPLIS works to a focus on creating content, both symbols and models. The differences between symbols, models, components, and devices is clearly delineated. The very important topic of subcircuit encryption is introduced, and the different encryption schemes are discussed. Finally, a deeper look under the SIMPLIS hood is covered, including how to troubleshoot slow models.

To download the examples for Module 4, click Module_4_Examples.zip.

Module 5 - Parameterization

Parameterization is one of the most important concepts in modeling. Before you can parameterize a model, you will learn how subcircuits provide a universal interface to electrical models. You will then learn that once the design has been divided into subcircuit blocks, adding parameters which configure those subcircuits will open a realm of modeling flexibility. The importance of the two key concepts of subcircuits and parameters **cannot** be over-emphasized. Finally, you will learn how to add two different parameter editing dialogs to your models.

To download the examples for Module 5, click Module_5_Examples.zip.

Module 6 - Modeling

In this module you will build real, production-worthy models. These models will have the look and feel of any of the models built into SIMPLIS. These models will be robust and will include inherent parameter error checking and will have parameter editing dialogs. Most importantly, you will learn how to exclude schematic component subcircuits from the model based on parameter values. This further opens the realm of circuit parameterization - that parameters can select the schematic configuration.

To download the examples for Module 6, click Module_6_Examples.zip.

- 6.0 Model Requirements
- 6.1 Building a Subcircuit Load
  - 6.1.1 Constant Resistance Subcircuit
  - 6.1.2 Constant Current Subcircuit
  - 6.1.3 Constant Voltage Subcircuit
- 6.2 Assembling the Subcircuit Load

Applications

In this module, you will learn how to use PWL modeling to model several blocks commonly used in switching power converters. This module is under construction, with other topics coming. The first topic describes in detail how SIMPLIS can model MOSFET drivers. The second topic covers how SIMPLIS can simulate the Load Line characteristics of a multi-phase VR regulator.

To download the examples for the Applications Module, click Applications_Examples.zip.

- Application A - Create MOSFET Driver Model
- Application B - Modeling and Measuring Power Stage Efficiency
- Application C - Low-Voltage High-Current Tuned Load Line Techniques
- Application D - Using the Design Verification Module
- Application E - Digital Control: Convert Analog Compensation Network to Digital Compensation Network

1.0.1 SIMPLIS is a Time-Domain Simulator, all the Time, for Every Analysis, Period

SIMPLIS is a time-domain simulator optimized for switching power converters. Most users come to SIMPLIS with some experience using SPICE simulators, which work on an entirely different set of principles. This topic introduces an important difference between SIMPLIS and SPICE. SIMPLIS is exclusively a time-domain simulator, even though it can present its time-domain results very accurately in frequency-domain plots.

In this topic:

Key Concepts

This topic addresses the following key concepts:

- The SIMPLIS simulator is a time-domain simulator which uses Piecewise Linear (PWL) models.
- The AC Analysis requires a periodic operating point (POP) analysis to be run first to find the switching steady-state operating point.
- The Transient analysis run after a POP analysis is initialized to the POP steady-state operating point.

What You Will Learn

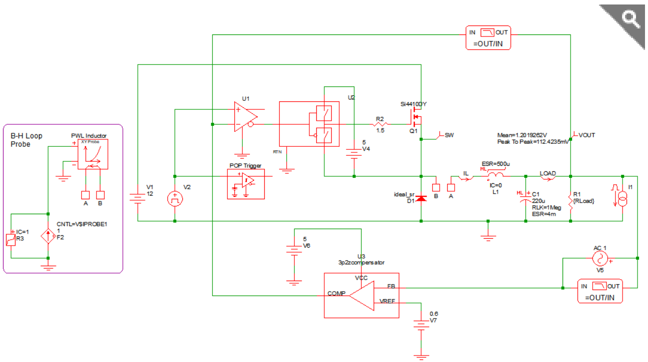
In this topic, you will learn the following:

How SIMPLIS analyzes circuits exclusively in the time domain. This includes the SIMPLIS AC analysis, which is carried out in the time domain.

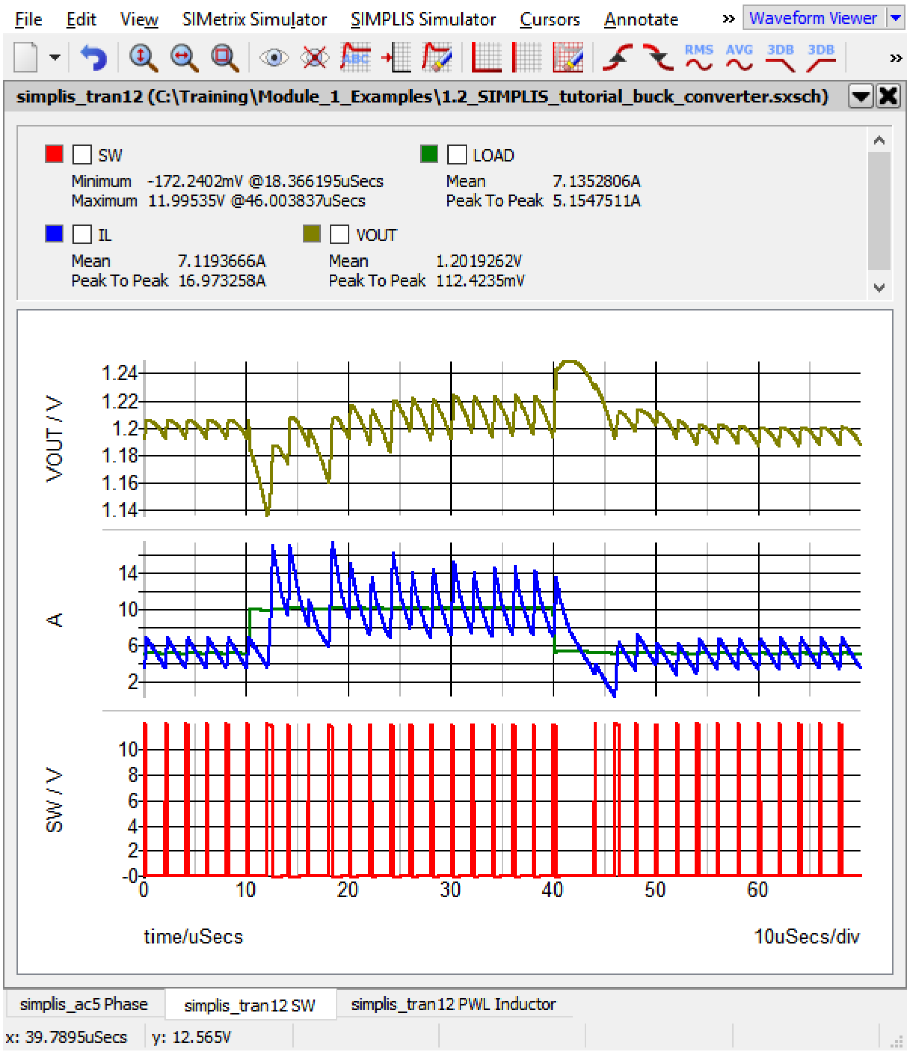
Getting Started: Running SIMPLIS

In this exercise, you will simulate a synchronous buck converter in the each of the three SIMPLIS analyses, Periodic Operating Point, AC analysis and Transient Analysis.

1. Open the schematic titled **1.2_SIMPLIS_tutorial_buck_converter.sxsch**.
   Result: *The buck converter schematic opens:*



2. To simulate the design, press **F9** or from the menu bar, select **Simulator** > **Run Schematic** .
   Result:
   - *The SIMPLIS simulator simulates the same time-domain nonlinear schematic in each of the three analysis modes, Periodic Operating Point (POP), AC, and Transient.*
   - *The SIMPLIS Status window opens when the simulation is first launched, and the waveform viewer displays the simulation results as the simulation progresses.*
   - *The results from the POP analysis are not displayed, as the transient analysis was specified. The transient simulation begins at the operating point found by the POP analysis and only the transient analysis results are displayed.*
   - *The simulation results displayed in the waveform viewer include waveforms plotted versus time as well as time-domain waveforms that are plotted against each other using X-Y plots, where time in an implicit variable.*



After running the simulation, the waveform viewer contains a number of graphs. The left-most tab has the gain and phase of the converter control loop taken from the AC Analysis. The other tabs have the results of the transient analysis.

Discussion

SIMPLIS runs these three analyses in the following order:

1. Periodic Operating Point (POP) Analysis
2. AC Analysis
3. Transient Analysis

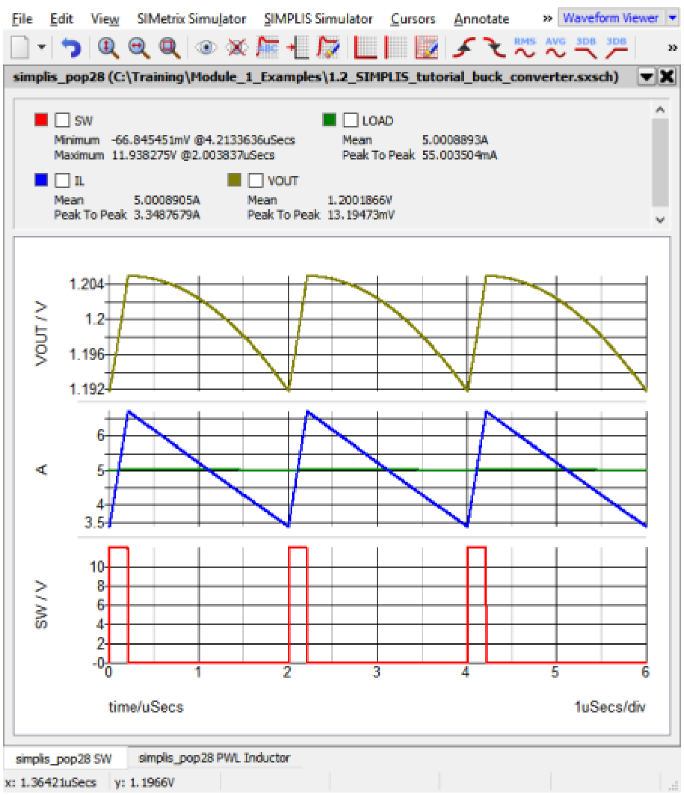
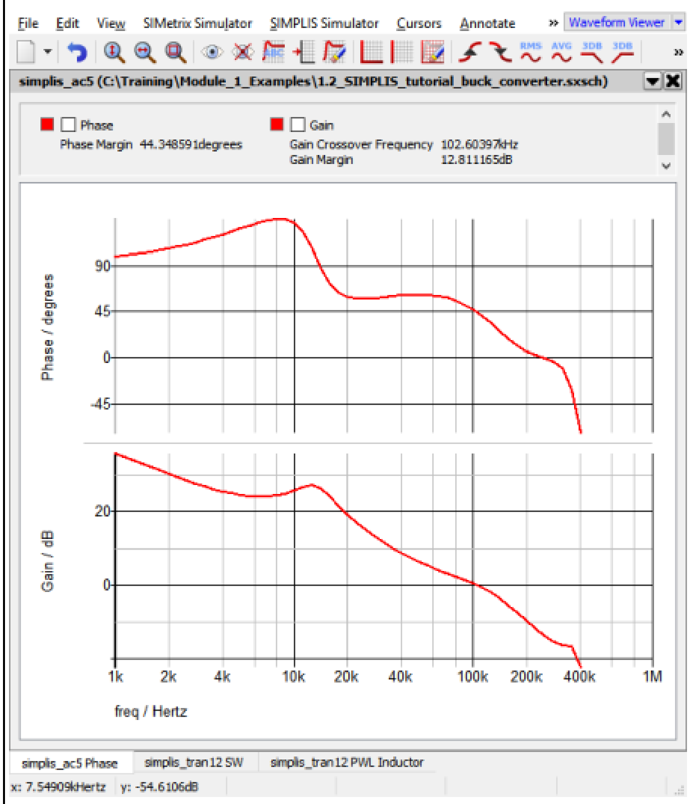
The first analysis run is the POP analysis. The POP analysis finds the switching steady-state operating point of the circuit. This steady-state operating point is then used to:

1. Perform a small signal AC analysis on the circuit at this steady-state operating point.
2. Initialize the circuit for the following transient analysis.

Each of these analyses are executed in the time domain, which is exactly what happens on the lab bench. The Periodic Operating Point analysis is discussed in detail in section 1.0.5 POP Analysis, for now think of the POP analysis as a way to accelerate the process of getting to steady state. A key point to remember is that without the Periodic Operating Point, you cannot run an AC analysis on the circuit.

The AC analysis is carried out on the time domain model by first finding the Periodic Operating Point, then injecting a single time-domain sinusoidal perturbation signal into the circuit. The AC results are then calculated from the time domain response to the perturbation signal. Then the injected signal is stepped to the next frequency to be analyzed and the measurement process is repeated until the entire requested frequency range is covered. No averaged model is used. All AC analysis results are derived from the time-domain response of the full nonlinear system.

The time domain POP waveforms and the frequency-domain loop response of the Synchronous Buck Converter are shown below. The frequency response of the circuit is valid at the steady-state condition found during the POP analysis.

Time Domain Waveforms	Frequency Response of Time Domain Model
	

The transient analysis is similar to a transient analysis in other simulators, except it typically runs much faster.

Conclusions and Key Points to Remember

SIMPLIS operates just like your circuit in the laboratory - in the time domain.

- Your power electronic switching system in the lab has no concept of a DC operating point or an AC small signal model.
- The power switches turn ON and then OFF as determined by a modulator control circuit that senses the output and tries to regulate the circuit performance accordingly.
- If the circuit is not switching, it is not working correctly.
- A switching converter has no DC operating point.
- An averaged AC model is a theoretical construct, it does not exist on the lab bench.
- If you cannot successfully perform a POP analysis on your circuit:
  1. You cannot easily evaluate the AC performance of the circuit. (Caveat: DVM has a Multi-Tone AC analysis, but this takes much longer than the combination of a POP and AC analysis.)
  2. In the vast majority of cases, your simulations will take much longer, since you will have to first wait for the converter to reach steady state before you can perform your intended analysis.

1.0.2 PWL Simulation and Modeling

Every device model used in a SIMPLIS simulation uses Piecewise Linear (PWL) modeling techniques. This includes semiconductor devices such as MOSFETs and Diodes. In this topic you will learn how SIMPLIS models non-linear devices with PWL models.

In this topic:

Advanced SIMPLIS Training

Course Outline

Installing the Training Course License

Getting Started

Module 1 - Overview of the SIMPLIS Environment

Navigating the Course Material

1.0 SIMPLIS Basics

1.0.1 SIMPLIS is a Time-Domain Simulator, all the Time, for Every Analysis,

Key Concepts

This topic addresses the following key concepts:

- Every model used in SIMPLIS is a Piecewise Linear (PWL) model.
- Non-linear characteristics are modeled with PWL primitive resistors, inductors or capacitors.
- Complex devices, such as MOSFETs can be represented by a collection of PWL devices.
- In SIMPLIS, diodes can be nothing more than PWL resistors.

What You Will Learn

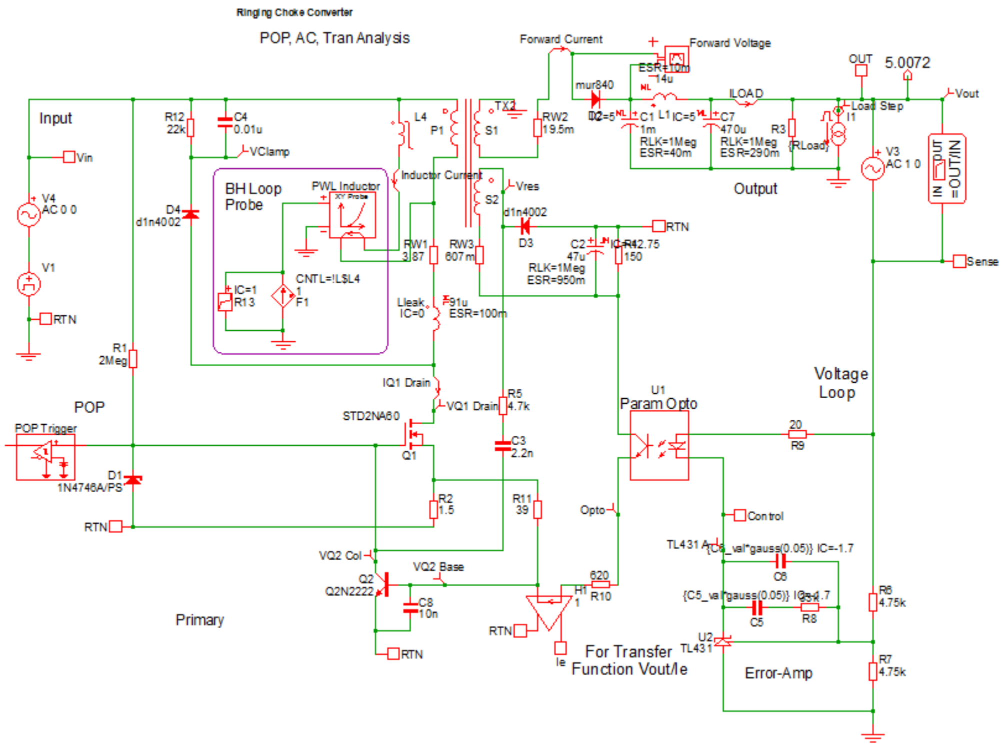
In this topic, you will learn the following:

- How transformer saturation is modeled with PWL inductors.
- How SIMPLIS uses a collection of PWL devices to model a MOSFET.

Getting Started

This topic uses a self-oscillating flyback converter to demonstrate PWL modeling techniques. The converter is intentionally overloaded, causing the converter to enter into a current limited operation. To get started with this example, follow these steps:

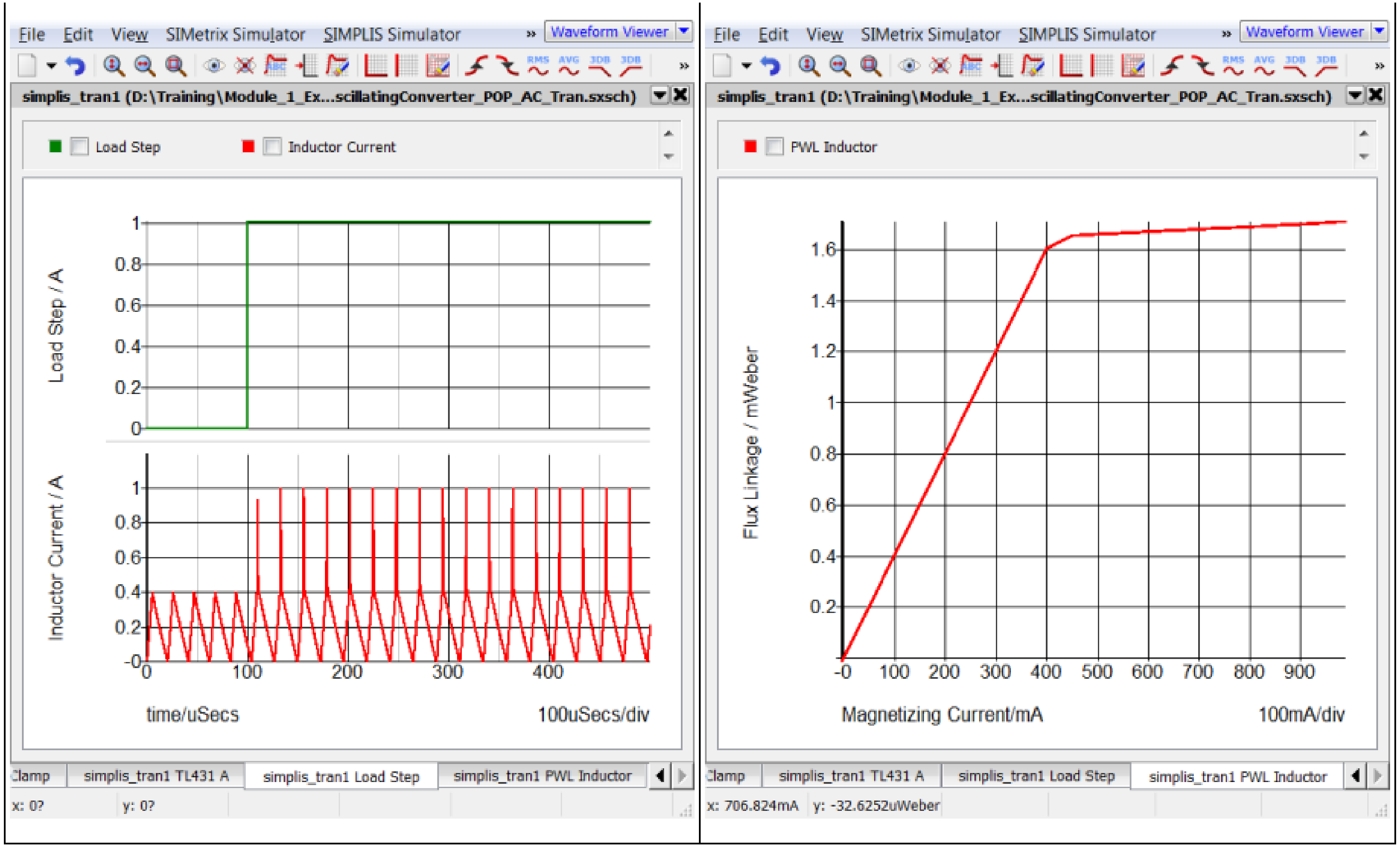
1. Open the schematic titled **1.1_SelfOscillatingConverter_POP_AC_Tran.sxsch**.
   Result: *The flyback converter schematic opens:*



2. To simulate the design, press **F9** or from the menu bar, select **Simulator** > **Run Schematic** .

   Result: *After the simulation completes, the waveform viewer displays multiple graph tabs. If you have closed the waveform viewer or one of the graph tabs, run the simulation again to regenerate the graphs. The right-most two graphs are of interest, these two graph tabs will be similar to:*

Time Domain Waveforms	B-H Loop of Time Domain Model



Discussion

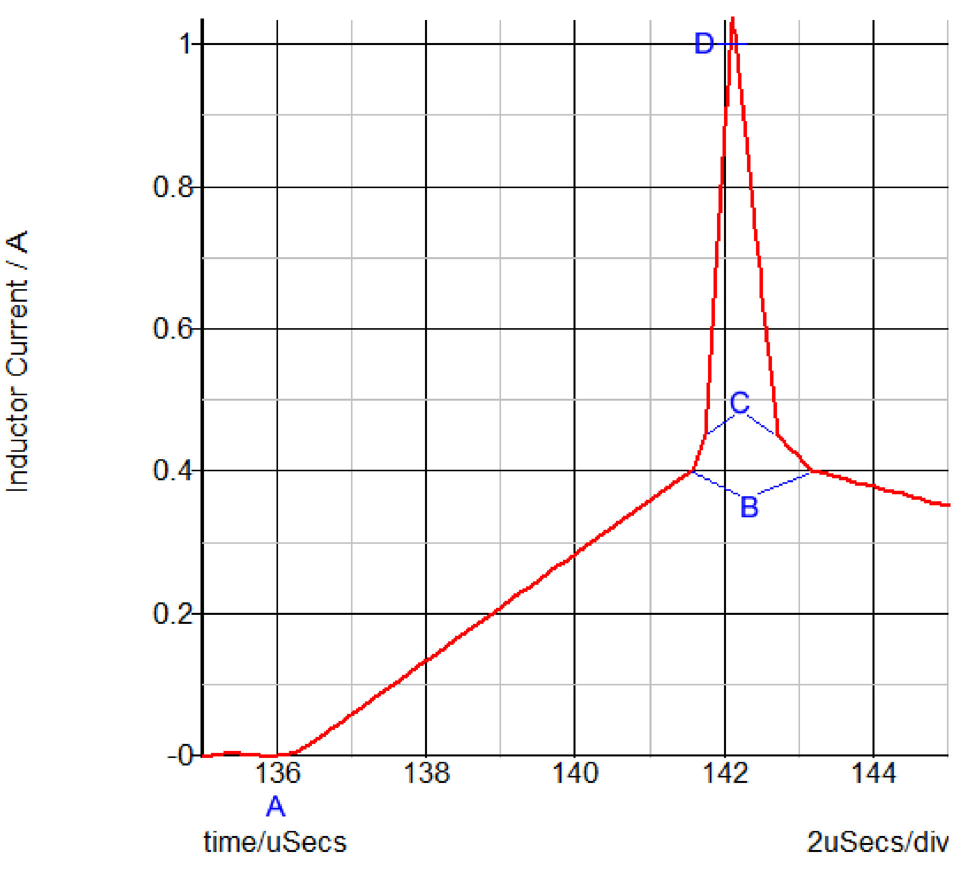
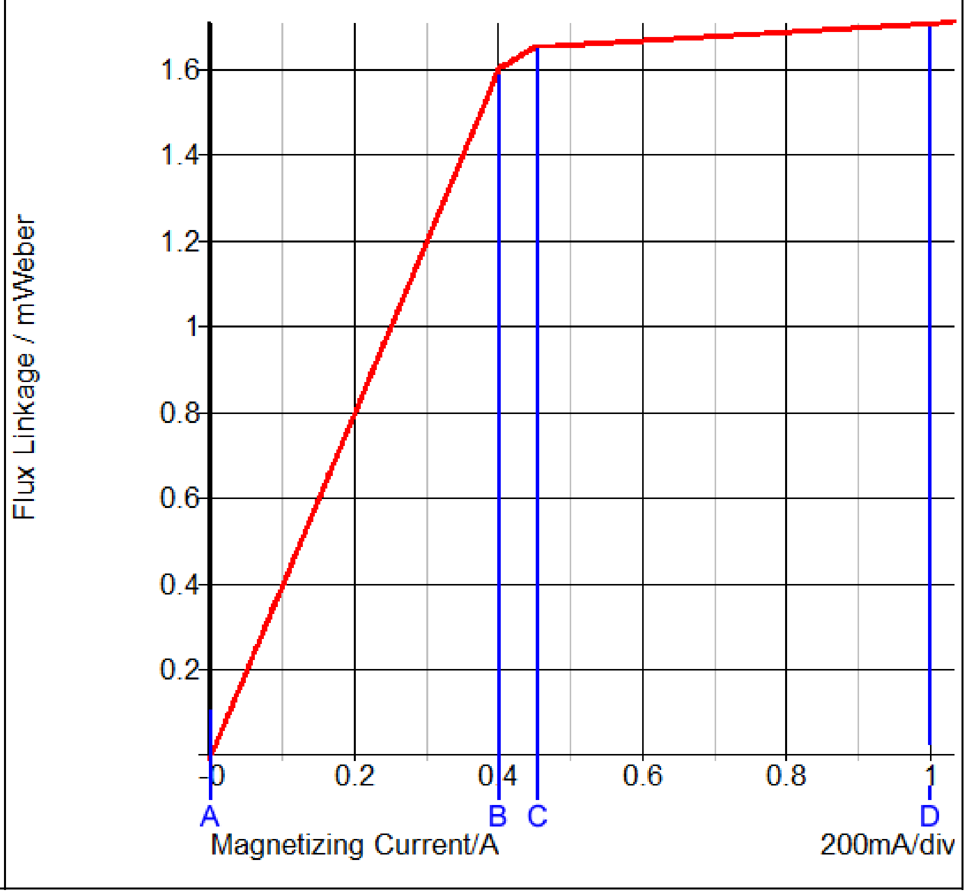
PWL Inductors

The left hand graph contains two curves, the **Load Step** in green, and the **Magnetizing Inductor Current** in red.

During the transient simulation, a one amp load step is applied at 100us. During this load step, the total load current transitions from a 2A full load condition to a 3A overload condition with the consequence that the transformer enters into saturation. The current limit function is triggered and the output voltage drops.

This overload condition demonstrates how a PWL inductor models transformer saturation. The following two graphs show a close-up view of the time-domain magnetizing inductor waveform and the flux linkage versus current plane on which PWL inductors are defined. Each of these three PWL inductor segments can be seen in both the transient simulation results and in the x-y plot of the flux linkage versus current plane show below:

Magnetizing Current and B-H Loop for Self-Oscillating Converter

Saturating Magnetizing Current	B-H Loop
	

The saturation of this transformer is modeled with three PWL segments in the flux linkage versus current plane. Because the slope of this curve is the magnetizing inductance, the magnetizing inductance can take on three distinct values:

- When the magnetizing current is below 0.4A, the normal or unsaturated magnetizing inductance of 1.6mWeber/0.4A equals 4mH is used.
- The knee of saturation occurs when the magnetizing current is between 0.4 and 0.45A. The inductance in this region is (1.65m-1.6m)/(0.45-0.40) which equals 1mH.

- The final PWL segment represents a "hard" saturation. The inductance of this segment is 100uH.

PWL MOSFETs and Diodes

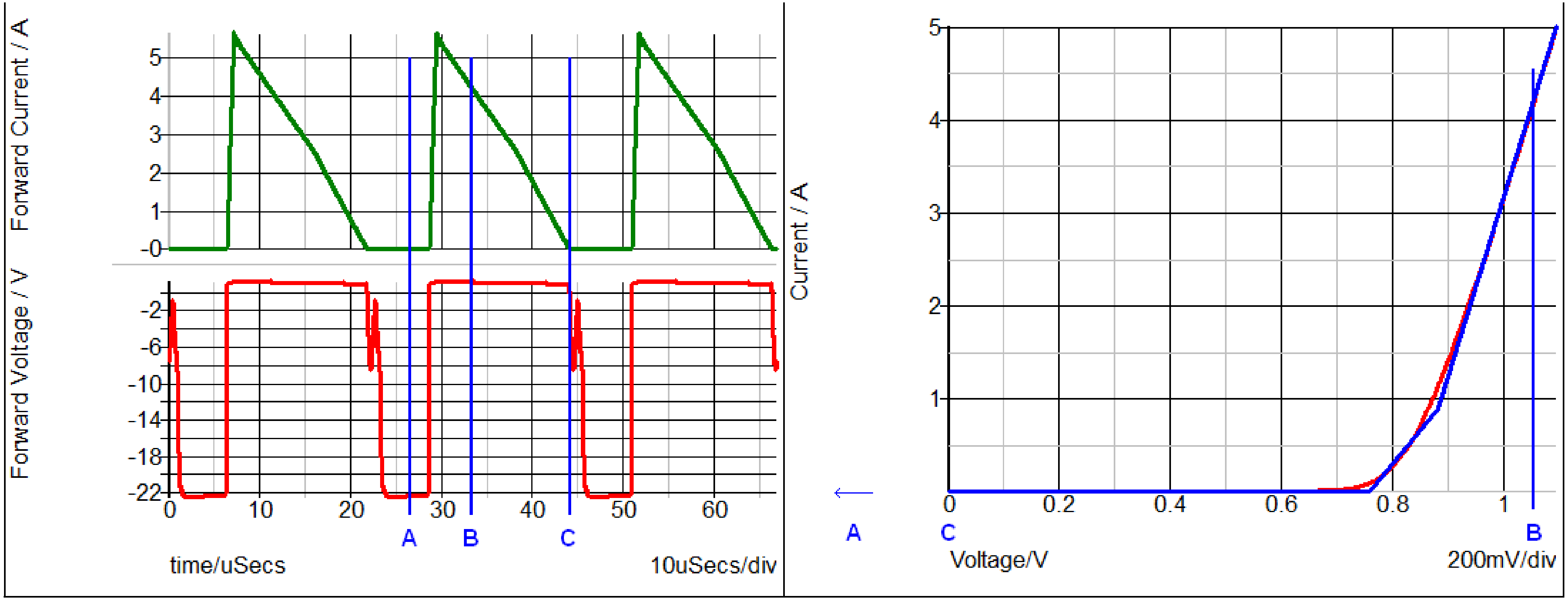
What about the Diodes and the MOSFET on the schematic? Are these PWL models as well? **Yes!**

- In SIMPLIS, the built-in diode models are nothing more than PWL resistors with either 2 or 3 segments. The PWL definition is usually generated with the automatic model parameter extraction routines built into SIMetrix/SIMPLIS.
- The built-in MOSFET models are made from a collection of PWL devices, including:
  - A PWL resistor representing the body diode
  - A transistor switch with a constant forward transconductance gain (Gm)
  - PWL capacitors representing the voltage dependent nonlinear capacitances.

As an example, the output rectifier in the Self-Oscillating Converter has the following Forward Current and Forward Voltage curves during the transient simulation. The left hand graph has the Forward Current and Forward Voltage plotted vs. Time. In the right-hand graph, the Forward Current is plotted versus the Forward Voltage for this diode. The blue curve is the 3 segment SIMPLIS PWL model. The red curve is the SIMetrix simulation results for SPICE model of the same diode.

Output Rectifier Voltage and Current

Voltage and Current vs. Time	Voltage vs. Current



The annotated points describe the following diode states:

- A: Blocking state
- B: Conducting 4A forward current
- C: Transition from conduction to blocking state

An in depth discussion of MOSFET modeling is presented in section 1.0.3 Multi-Level Modeling.

Conclusions and Key Points to Remember

- Every device used in a SIMPLIS simulation is, behind the scenes, a PWL model. This is independent of the symbol. The symbol is merely a graphical representation of the underlying function being modeled.
- The nonlinear characteristics of Resistors, Capacitors, and Inductors are modeled as a series of PWL straight line segments.
- Even complex devices, such as MOSFETs can be represented by a collection of PWL devices.

1.0.3 Multi-Level Modeling

Multi-Level models are a key feature of SIMPLIS. Multi-Level models use a single parameter to configure the model complexity. The multi-level modeling concept allows models to be tailored to the application, where minimum model complexity is used for the simulation objective, which in turn results in the fastest simulations. In this topic you will learn about two types of multi-level models: multi-level MOSFETs and multi-level capacitors.

In this topic:

Key Concepts

This topic addresses the following key concepts:

- Multi-Level models are configured with a single parameter.
- The schematic-view of the model changes based on the model level parameter.
- The model level is chosen based on the desired simulation objective.

What You Will Learn

In this topic, you will learn the following:

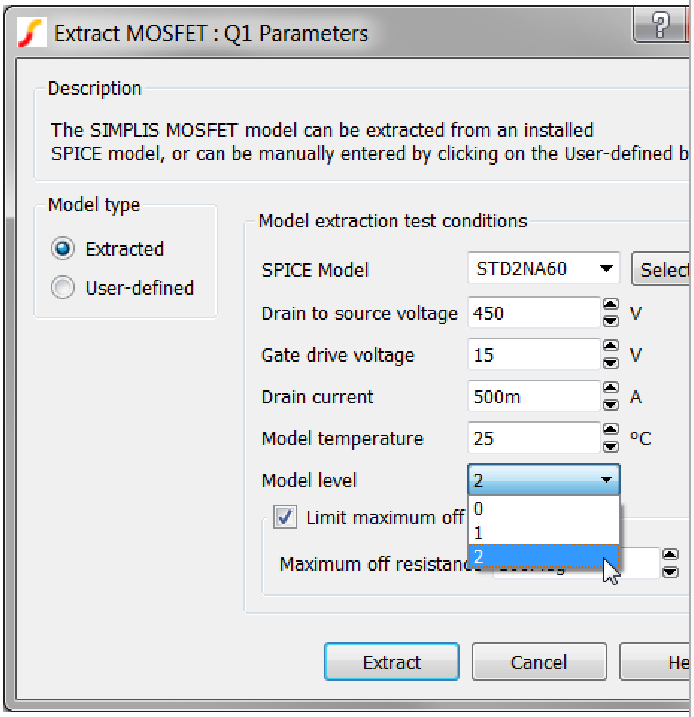
- How models can be configured with different levels of complexity with a single parameter.
- How to choose the appropriate level based on your simulation

Advanced SIMPLIS Training
Course Outline
Installing the Training Course License
Getting Started
Module 1 - Overview of the SIMPLIS Environment
Navigating the Course Material
1.0 SIMPLIS Basics
1.0.1 SIMPLIS is a Time-Domain Simulator, all the Time, for Every Analysis, Period
1.0.2 PWL Simulation and Modeling
1.0.3 Multi-Level Modeling
1.0.4 Accuracy of PWL Models
1.0.5 POP Analysis
1.0.6 AC Analysis
1.1 Introduction to DVM: What is DVM?

objective.

Getting Started: Multi-Level MOSFET Model

1. Open the schematic titled **1.1_SelfOscillatingConverter_POP_AC_Tran.sxsch**.
2. Double click on the MOSFET **Q1**.
   Result: *The Extract MOSFET Dialog opens. The Model level parameter control is shown below:*



3. Click on the **Help** Button in the lower right corner of the dialog.
   Result: *The Help system opens to the SIMPLIS MOSFET Models topic.*

Discussion: Multi-Level Modeling

At this point, you should have the Extract MOSFET Dialog open in SIMetrix/SIMPLIS and the SIMPLIS MOSFET Models help topic open in a browser window.

MOSFETs, in common with the other semiconductors such as Diodes, Zener Diodes, IGBTs and JFETs have very nonlinear behavior. For example, the drain-to-source capacitance of a MOSFET can radically change as the voltage across the MOSFET changes from the blocking to the conducting state. If you are interested in the switching behavior of this device, it is important to model this capacitance change; However, if you primarily interested in the Bode Plot of the converter, the details of the switching transition are typically not important.

SIMPLIS has the ability to change both the underlying schematic structure and the parameters of a model based on a single parameter value. In the Extract MOSFET Dialog, and indeed, in many SIMPLIS built-in models, the "Level" or "Model Level" parameter controls the schematic view of the model which is used in the simulation.

Multi-Level Model Example #1: The SIMPLIS MOSFET

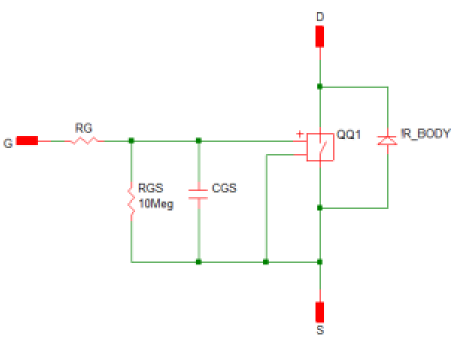
The MOSFETs used in SIMPLIS have four levels of complexity. Each level is described in detail in the currently open help topic. Below are the schematic views of the level 0 , 1, and 2 models. The level 3 model is intended for user-customized models, and is not supported by the internal model extraction routines.

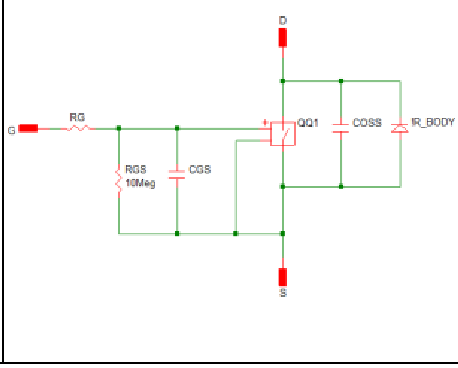
Level 0 Model	Level 1 Model	Level 2 Model

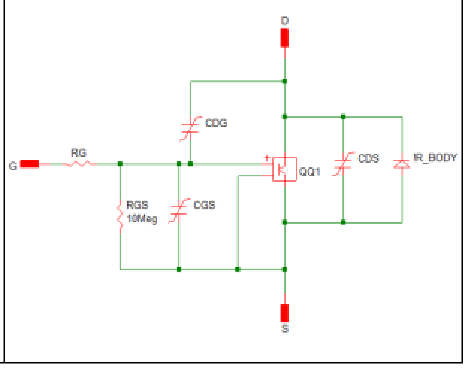
QQ1:	Switch with On and Off Resistance
CGS:	Linear Capacitance
RGS:	10Meg Ω Resistor
RG:	Internal Gate Resistor
!R_BODY:	Body diode modeled by PWL Resistor

QQ1:	Switch with On and Off Resistance
CGS:	Linear Capacitance
RGS:	10Meg Ω Resistor
RG:	Internal Gate Resistor
!R_BODY:	Body diode modeled by PWL Resistor
COSS:	Lumped linear output capacitance

QQ1:	Switch with forward transconductance
CGS:	PWL Capacitance
RGS:	10Meg Ω Resistor
RG:	Internal Gate Resistor
!R_BODY:	Body diode modeled by PWL Resistor
CDS:	PWL capacitance
CDG:	PWL capacitance







The Level 0 MOSFET is used whenever the detailed switching action of the MOSFET is not important. The Level 2 MOSFET, which models the nonlinear capacitances, is typically used when the switching transitions are important, such as when measuring efficiency. The Level 1 MOSFET is used for power stage development when the converter topology relies on the MOSFET output capacitance.

Model Parameter Extraction

1. If you have closed the Extract MOSFET dialog, reopen it by double clicking on the MOSFET **Q1**.
2. Click **Extract**.
   Result: *A progress bar briefly displays the progress as SIMetrix/SIMPLIS extracts the SIMPLIS model parameters from the SPICE model.*
3. Look in the SIMetrix/SIMPLIS command shell window.

   Tip: You can press the space bar to bring the command shell into view.

   You should see the following message:

```
Extracting SIMPLIS model for STD2NA60. Please wait..
Complete
```

You have just executed several SIMetrix SPICE simulations on the SPICE model for the STD2N60 MOSFET, curve-fit the SPICE simulation data to a SIMPLIS PWL model, and written 66 model parameters to the symbol. Congratulations! Nice work!

The Multi-Level Modeling concept is at the core of this process - that models can have varied complexity based on the application. You can maximize simulation speed by using the minimum model level required for your analysis.

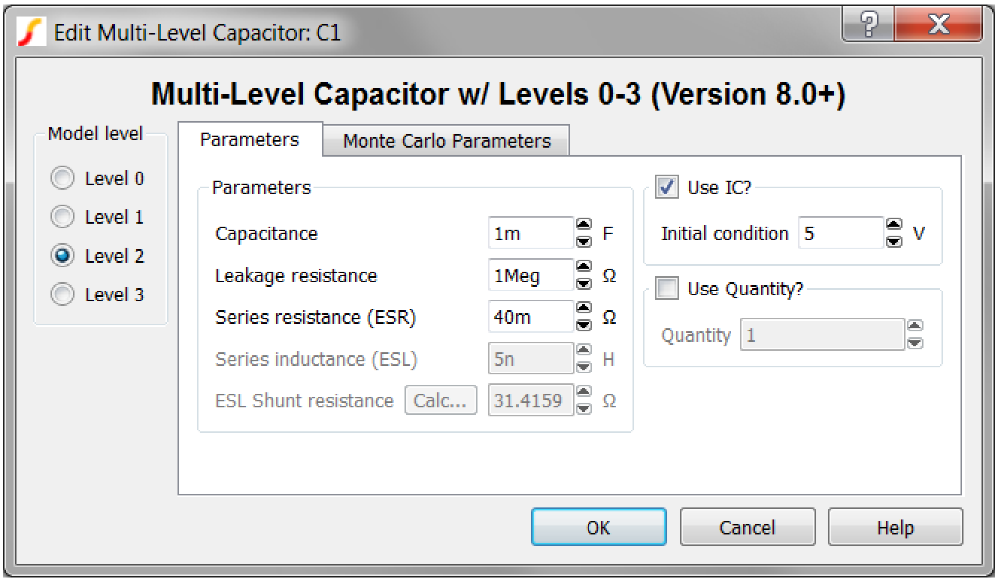
As you have just seen, SIMetrix/SIMPLIS has the unusual ability to simulate SPICE semiconductor models and from these results to extract a PWL SIMPLIS model. This capability becomes especially powerful when you combine it with the Multi-Level Modeling concept. Now critical device models can have the appropriate level of complexity based on the simulation objective of a particular simulation run.

Multi-Level Model Example #2: The Multi-Level Capacitor

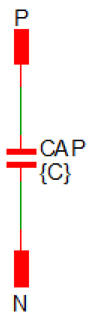
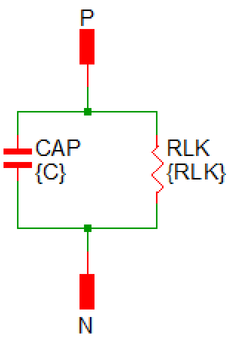
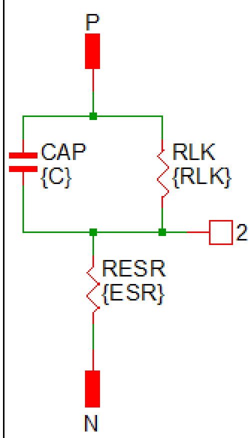
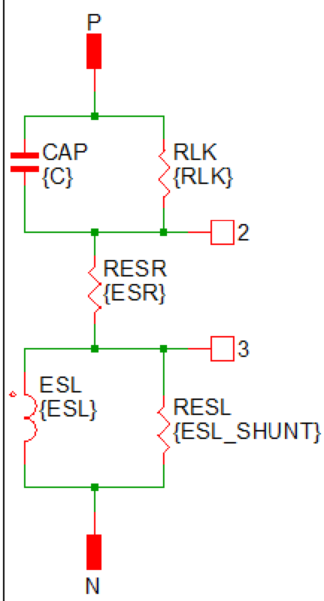
Passive components, such as inductors and capacitors are also available as multi-level models. The model level in this case determines the parasitic elements which are included in the model. In this section the new Multi-Level capacitor is used as an example. This capacitor was introduced with SIMetrix/SIMPLIS version 8.0, and replaces the old electrolytic capacitor.

Exercise #2: Multi-Level Capacitor Model

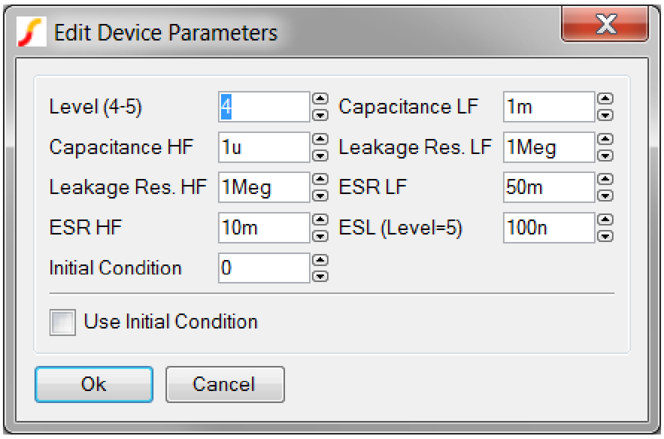
1. Double click on the output capacitor **C1**. This is the first capacitor symbol to the right of the transformer output.
   Result: *The Edit Multi-Level Capacitor dialog opens:*



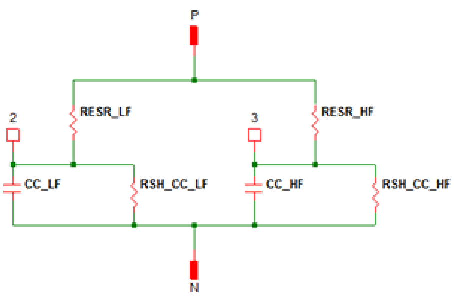
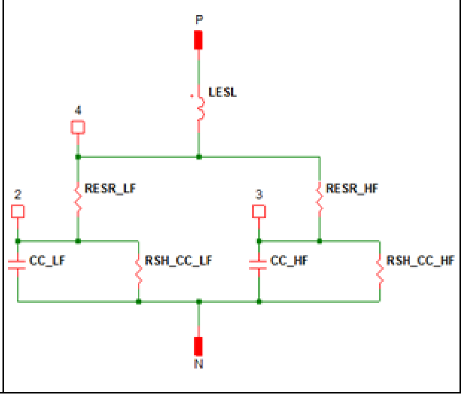
SIMetrix/SIMPLIS has two multi-level capacitor models; the one used here has model levels 0-3, and a second, more detailed model, has levels 4 and 5. The three multi-level capacitors on this schematic all use the new multi-level capacitor model with levels 0-3 and have the model level set to 2. Equivalent schematic images of the level 0-3 models are shown below:

Level 0	Level 1	Level 2	Level 3
CAP Capacitance	**CAP** Capacitance **RLK** Leakage Resistance	**CAP** Capacitance **RLK** Leakage Resistance **RESR** Equivalent Series Resistance	**CAP** Capacitance **RLK** Leakage Resistance **RESR** Equivalent Series Resistance **ESL** Equivalent Series Inductance **RESL** ESL Shunt Resistance
			

The Level 4-5 model for the electrolytic capacitor models the low and high frequency components separately. The model is essentially two electrolytic capacitors in parallel. The edit dialog for the Level 4-5 electrolytic capacitor is shown below:



The schematic views of the level 4 and level 5 models are shown below. The Level 5 model adds a Equivalent Series Inductance (ESL) to the Level 4 model.

Level 4		Level 5	
CC_LF	Low Frequency Capacitance	**CC_LF**	Low Frequency Capacitance
RSH_CC_LF	Low Frequency Leakage Resistance	**RSH_CC_LF**	Low Frequency Leakage Resistance
RESR_LF	Low Frequency Equivalent Series Resistance	**RESR_LF**	Low Frequency Equivalent Series Resistance
CC_HF	High Frequency Capacitance	**CC_HF**	High Frequency Capacitance
RSH_CC_HF	High Frequency Leakage Resistance	**RSH_CC_HF**	High Frequency Leakage Resistance
RESR_HF	High Frequency Equivalent Series Resistance	**RESR_HF**	High Frequency Equivalent Series Resistance
		L_ESL	Equivalent Series Inductance
			

For future reference, you can place the Multi-Level Capacitors from the SIMPLIS parts selector:

- **Commonly Used Parts>Multi-Level Capacitor (Level 0-3 w/Quantity) (Version 8+)**
- **Commonly Used Parts>Electrolytic Capacitor (w/ HF ESR and ESL) (Level 4 - 5)**

Conclusions and Key Points to Remember

- Multi-Level modeling is extremely powerful because:
  1. A multi-level model is a flexible model. The model can range from simple to extremely complex based on the level parameter. The schematic view of the model can change based on the level parameter.
  2. The user can select the minimum complexity which meets the current simulation objective.
  3. By enabling one component symbol to have different underlying models the user can have a single schematic serve multiple simulation objectives. This can save a lot of time and confusion compared with having to manage multiple versions of the same schematic.
- All semiconductors used in SIMPLIS are modeled with PWL devices. To determine what level of model is used, double click on the symbol and note the value of the model level parameter. Using the lowest

complexity model level consistent with the required accuracy of the simulation objective will increase simulation speed.

- In Module 6 - Modeling you will learn how to create your own multi-level models. During the training course, think about what kind of models you have used which would benefit from the multi-level modeling concept.
- In SIMPLIS, the parameter name "Level" is used to describe the level of complexity of any given model. There is nothing special about this name, and you can build your own models with any parameter name you would like. For example, an IC model may have a "Process" parameter or a "Corner" Parameter. The Process might have three string values. The model could then have the string value configure the model:
    - 'Slow'
    - 'Typical'
    - 'Fast'

  The model could use the string value to configure the model. Using string values as variables is covered in 6.2 Assembling the Subcircuit Load.

Search

Home / Advanced SIMPLIS Training / Module 1 - Overview of the SIMPLIS Environment / 1.0 SIMPLIS Basics /
1.0.4 Accuracy of PWL Models

1.0.4 Accuracy of PWL Models

PWL device models are accurate for switching power converter applications because these devices spend most of a switching period in either their conduction or blocking states and transition very rapidly these two states. The areas where the PWL models have higher error occur during switching transitions where the model spends a very small portion of each switching cycle.

In this topic:

Key Concepts

This topic addresses key concept that PWL models are accurate both the time and frequency domains.

What You Will Learn

In this topic, you will learn why PWL models can accurately model the behavior of switching power systems.

Getting Started

The three examples in this section compare simulation results of different switching power supplies with measured lab data.

Discussion

When used to model switching power systems, Piecewise Linear (PWL) models can provide a high degree of accuracy. The inherent approximations of SIMPLIS time-domain device models need not seriously degrade the accuracy of the simulation results. A well constructed PWL model is designed to spend very little time in the areas where these PWL approximations have higher error. In section 1.0.2 PWL Simulation and Modelling, the forward transfer characteristic of a diode is depicted. In a switching power supply, the diode is often used as a passive switch. Consequently, a rectifying diode spends the vast majority of the time either blocking current flow or conducting. Very little time is spent in the transition between these two states. With a simulation objective of modeling the closed-loop behavior of a power supply, the PWL approximations used to model the knee region of a rectifying diode contribute an insignificant error to the accuracy of the system behavioral model.

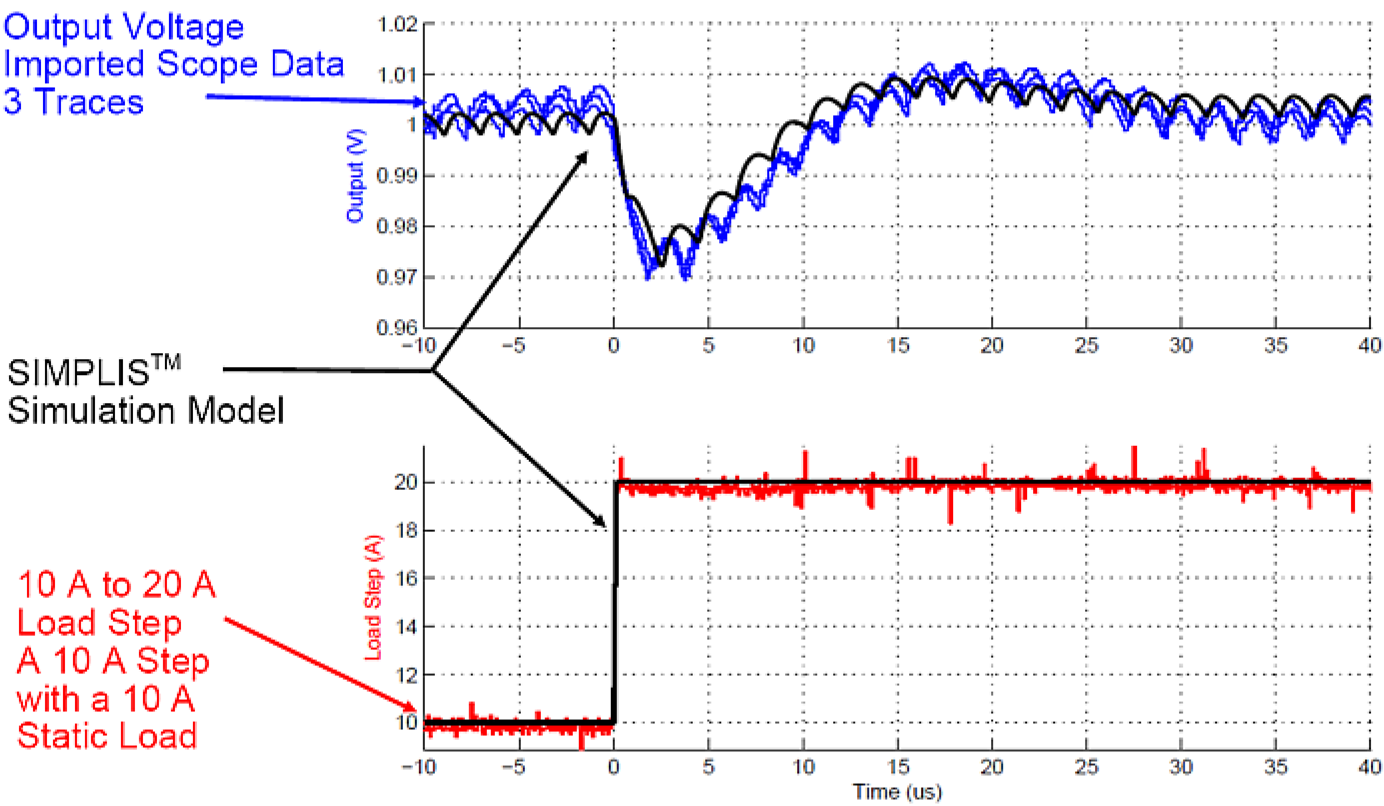
The three examples described below compare the SIMPLIS results with actual hardware testing.

Transient Example: Single Phase SyncBuck

This example comes from a single-phase, digitally-controlled, synchronous Buck converter. These oscilloscope images of the output voltage and output current were taken with three triggers of the hardware measurement. The SIMPLIS simulation results were overlaid on the hardware data. The maximum error during both a step load increase and step load decrease is 0.5%.

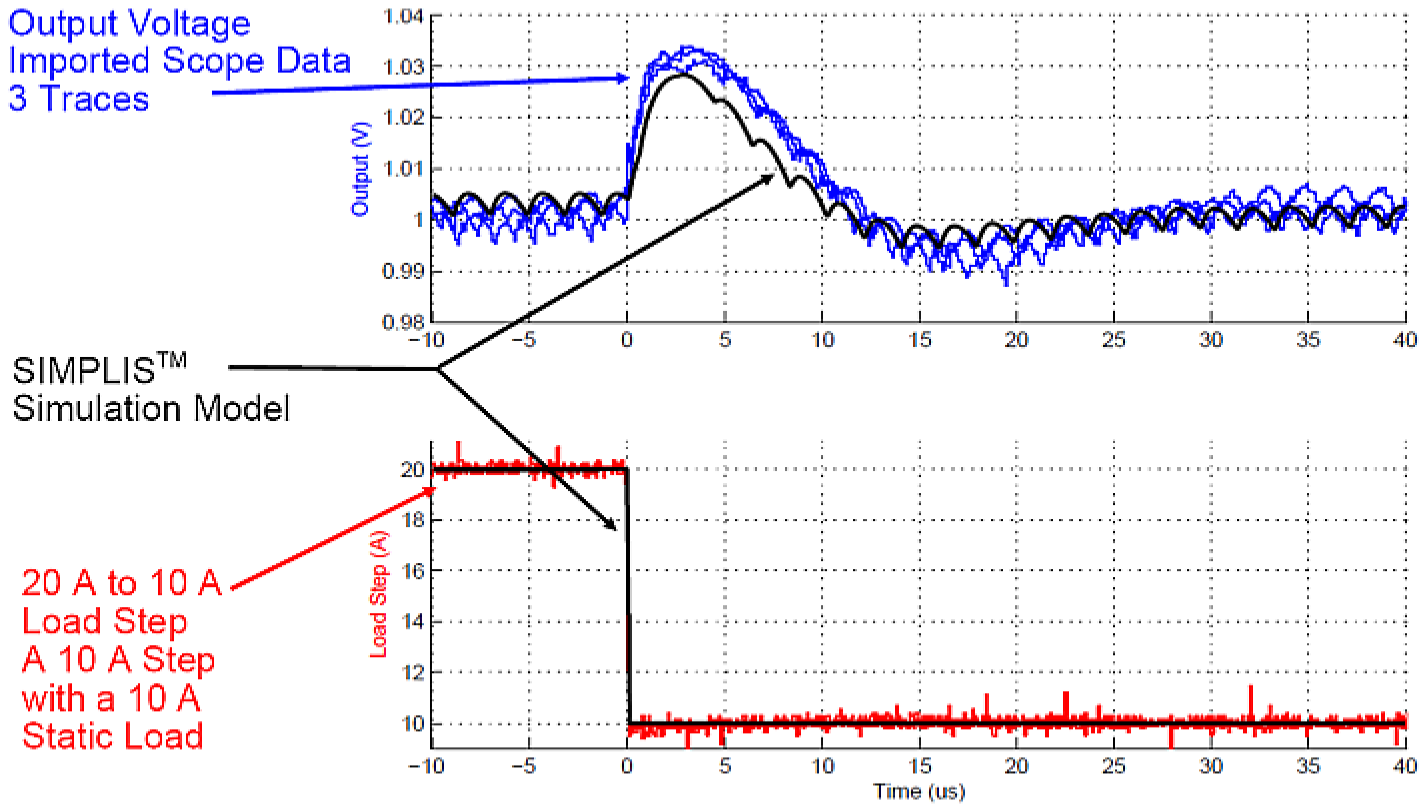
Step load increase from 10A to 20A

Time-domain simulation vs. experimental results



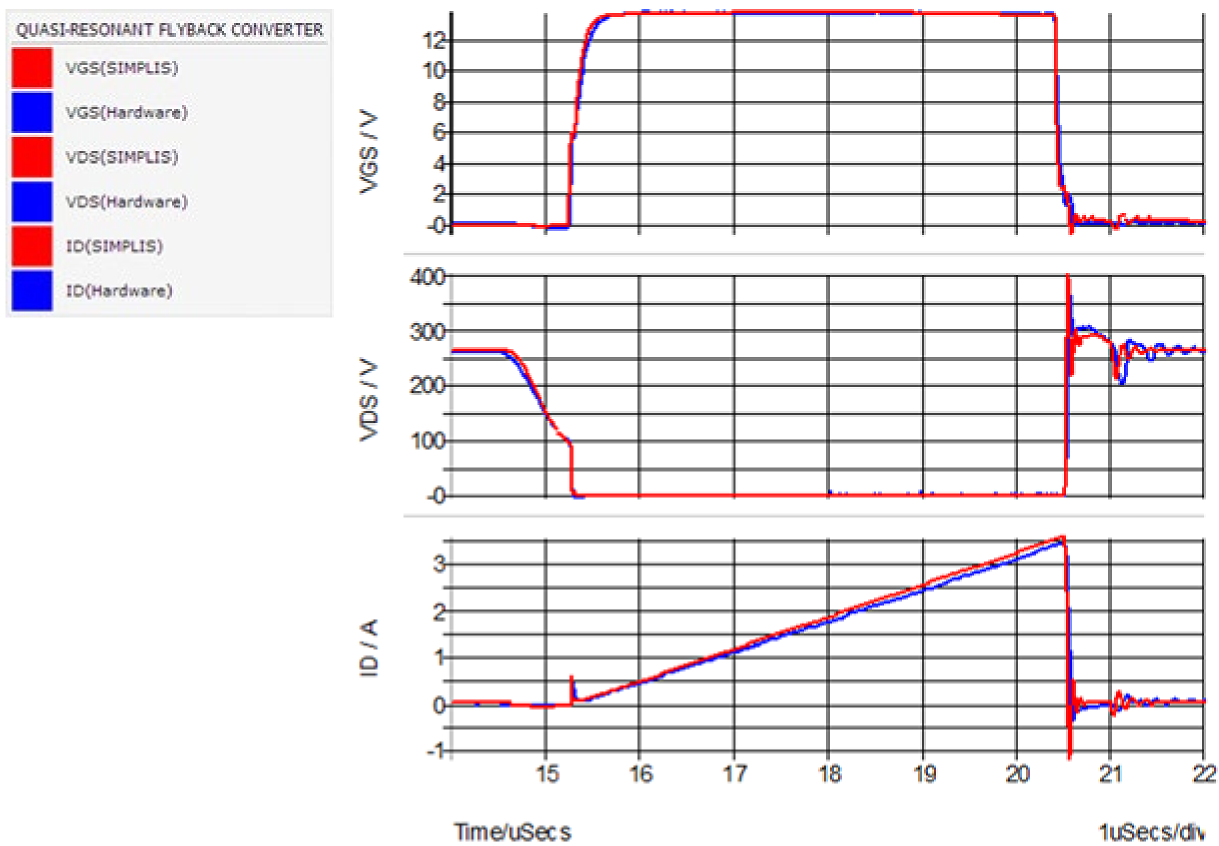
Step load decrease from 20A to 10A

Time-domain simulation vs. experimental results



Transient Example: Quasi-Resonant Flyback Converter

By including PWL capacitors to model the non-linear capacitances of the MOSFET, SIMPLIS can also do a good job of modeling the device switching transitions. This example is from a Quasi-Resonant Flyback Converter used in an AC adapter application. The main MOSFET and output rectifiers in the example were automatically converted from the SPICE models with the Model Parameter Extraction algorithms. The resulting SIMPLIS model correlates well to the hardware measurements of the gate-to-source voltage, drain-to-source voltage and drain current.

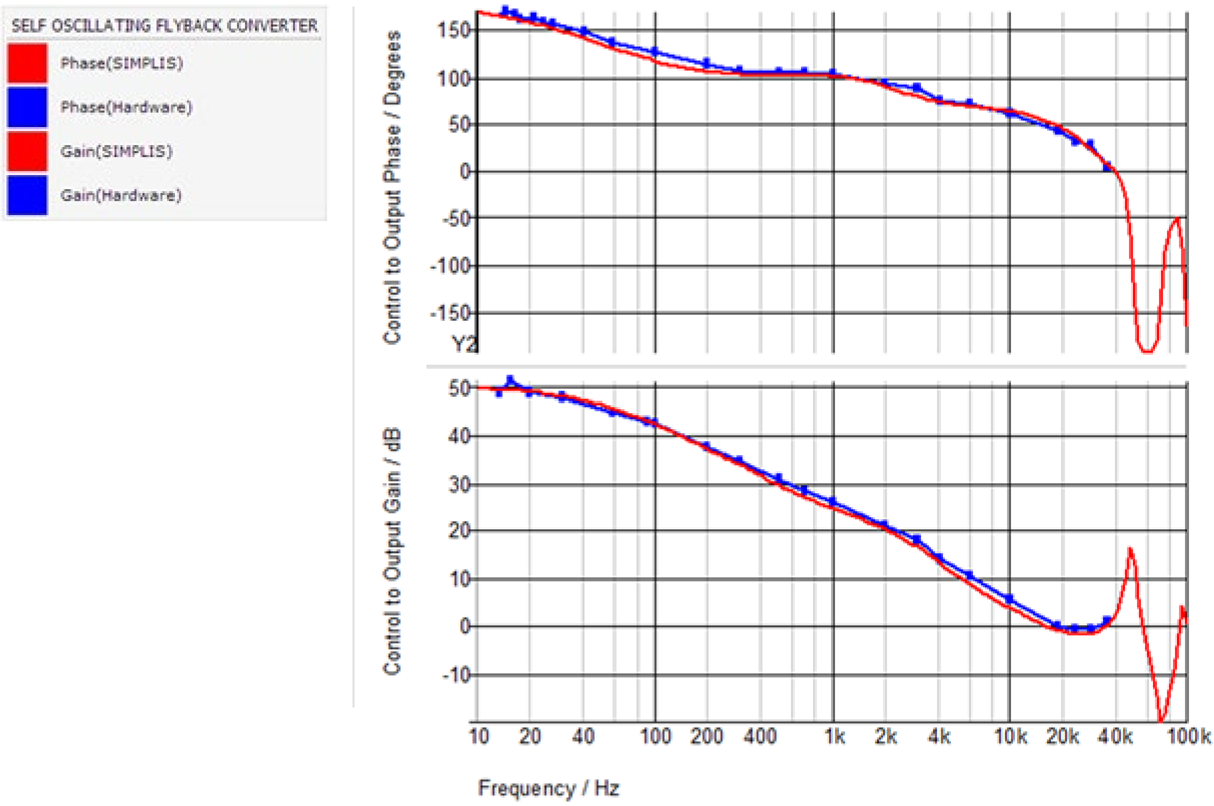


AC Example: Self Oscillating Flyback Converter

Numerically, SIMPLIS finds an extremely accurate solution to its system of PWL circuit equations that describe the time-domain behavior of a switching power supply. As discussed later in much more detail, SIMPLIS exploits this capability to find a very accurate steady-state periodic operating point of a switching system. Once the system is in steady state, SIMPLIS can inject an extremely small AC signal at a particular frequency into the system and then measure the time-domain response to that injected signal. By repeatedly performing this analysis at different frequencies, and then presenting the results as a function of the injected frequency, SIMPLIS is able to perform very accurate AC analysis using only the full nonlinear time-domain circuit model. No average AC modeling is required. There is also no requirement that the system employ constant-frequency control. The only requirement is that the system be in stable steady-state operation.

Because SIMPLIS is always simulating the nonlinear time-domain response, the AC analysis includes any effect of ripple voltages in the control signal path, which are generally ignored with averaged model techniques. As a result, with a carefully constructed model, the SIMPLIS AC analysis can closely match experimental data even when minute details of the converter operation are included.

The following graph compares the SIMPLIS AC analysis in red with bench measurements in blue on a hardware prototype. These simulation results were taken from the Self-Oscillating Converter circuit example from the 1.0.1 SIMPLIS is a time-domain simulator, all the time, for every analysis, period topic.



Conclusions and Key Points to Remember

- PWL modeling techniques can be quite accurate in the modeling of switching power systems.
- A PWL model should be designed to spend a very small proportion of the switching cycle in the area where the PWL model has higher approximation error.
- SIMPLIS can analyze circuits which do not have a closed form small signal model. The variable frequency Self-Oscillating converter is an example, as is the LLC converter.

Search

1.0.5 POP Analysis

The POP analysis is one of the most powerful capabilities of SIMPLIS. The POP analysis is a specialized transient analysis which quickly finds the switching steady-state operating point of a circuit. Once the steady-state operating point is found, an AC analysis at the periodic operating point can be performed on the circuit.

The POP analysis can also be followed with a transient analysis, in which case the transient simulation will start at the operating point found in the POP analysis. This is very useful for tests such as a pulse load transient where the circuit starts in steady-state.

In this topic:

Key Concepts

This topic addresses the following key concepts:

- The POP analysis is a specialized transient analysis.
- The POP analysis literally forces the circuit into a steady-state condition by putting an extra control loop around the converter.
- The POP analysis solves the steady-state operating point to a high level of precision, much higher than the RELTOL of a SPICE simulator.
- A successful POP analysis is required to run an AC analysis.
- SIMPLIS looks at your circuit in terms of **topologies**, or unique circuit configurations.

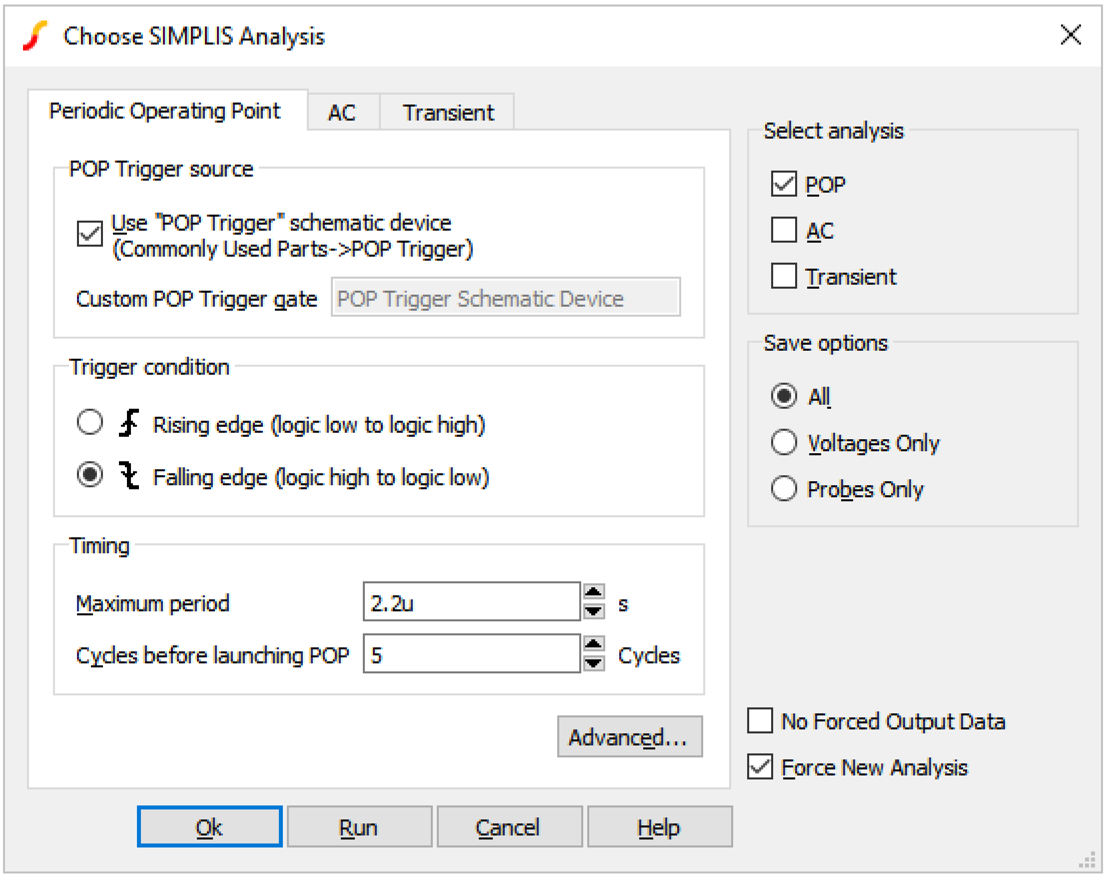
What You Will Learn

In this topic, you will learn the following:

- The basics of how a SIMPLIS Periodic Operating Point analysis works.
- Why POP is so important when simulating switching power circuits.
- What a new topology is.

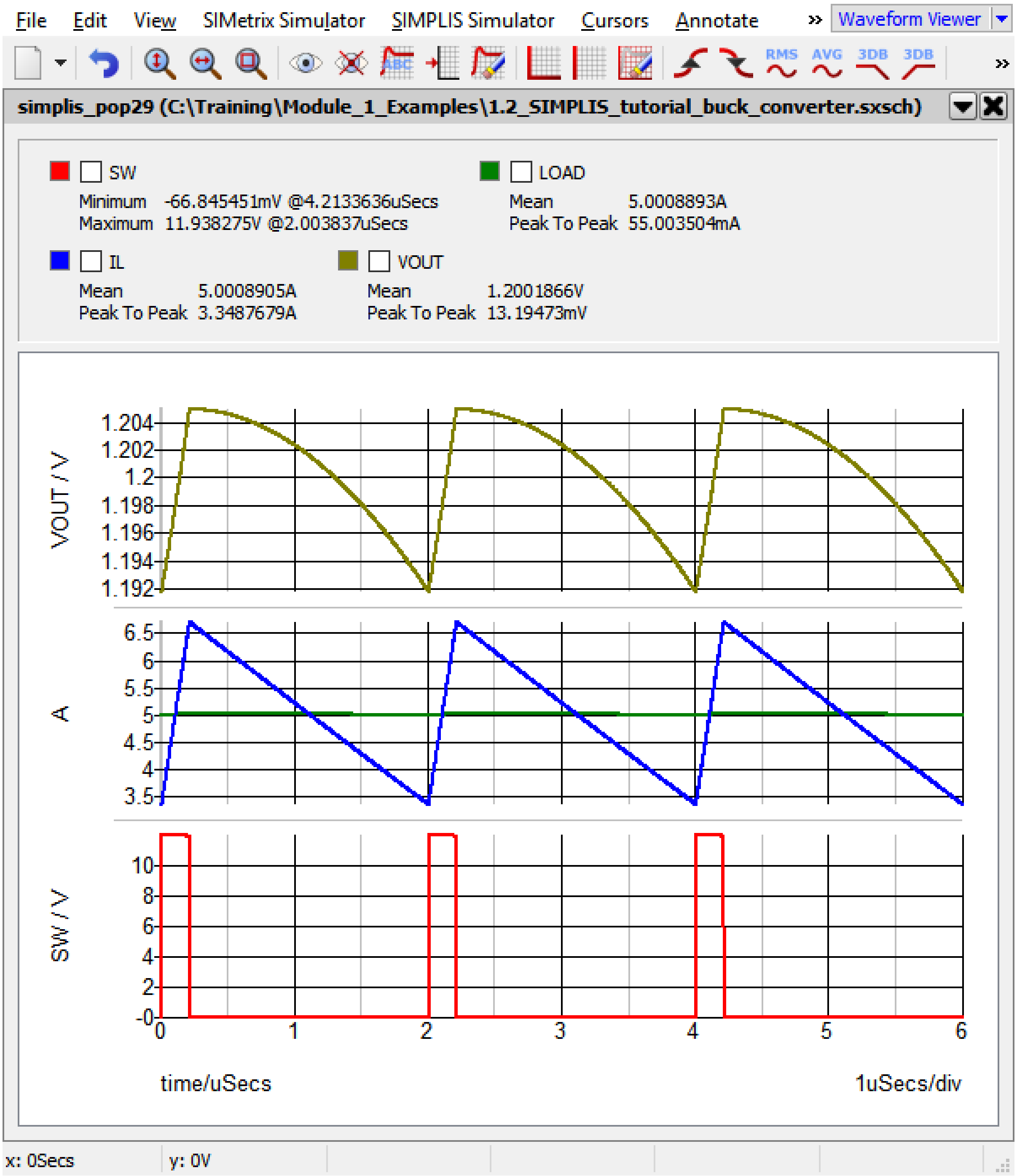
Getting Started: Running a POP Analysis

1. If the waveform viewer is open, close it.
2. If the SIMPLIS Status window is open, select the window (**Ctrl+Space**), and click the **Clear Messages** button to clear all messages from the window.
3. Open the schematic titled **1.2_SIMPLIS_tutorial_buck_converter.sxsch**.
4. From the menu bar select **Simulator** > **Choose Analysis...** .
5. Un-check the **AC** and **Transient** checkboxes, leaving the **POP** analysis checkbox checked.
6. The dialog should appear as follows:



7. Click **Run**.

   Result: *The POP analysis runs on the Synchronous Buck Converter, finding the switching steady-state operating point of the circuit. The waveform viewer opens with 3 switching cycles of data.*



Discussion

When you go into the lab and power up a switching power circuit, it has several seconds to settle into steady state before you view or capture your first oscilloscope image. Even the slowest PFC control loop with a bandwidth of a few Hertz will settle in the time between when you power up the circuit and when you first probe the circuit. Life in the simulator is a little bit different - we need a way to accelerate the time required to get to steady-state. This is exactly why the Periodic Operating Point was developed.

How POP Works

POP is essentially a software control loop around your power supply control loop. POP monitors each switching cycle of the converter. The POP Trigger device detects a waveform edge signaling the beginning of the next switching cycle, much like the oscilloscope trigger captures waveforms in the lab. At each edge, the POP algorithm takes a number of actions:

1. Samples and records each capacitor voltage and each inductor current.
2. Records the current operating segment of each PWL device, whether that device is a resistor, capacitor, or inductor.
3. Records the state of each switch in the circuit.

Armed with this information, POP then simulates the circuit for another switching cycle. POP then re-samples the capacitor voltages and inductor currents, and makes a calculation to determine if the values are essentially the same from one switching edge to the next switching edge. If the percent error is less than the POP convergence specification, the POP algorithm decides the converter is in steady state and exits. The simulation time is reset to zero, and a user specified number of switching cycles, three in this case, are simulated and plotted on the waveform viewer.

What if the sampled values from one switching edge to the next are greater than the convergence specification? POP will take another pass through the loop, during each pass:

1. POP will predict what the capacitor voltages and inductor currents should be for the converter to be in a steady-state condition.
2. POP loads the circuit with these initial conditions and re-starts the simulation.
3. At the next switching edge, the process repeats.

SIMPLIS Status Window Displays POP Progress

The SIMPLIS simulator outputs the simulation progress directly to the SIMPLIS Status Window. The data output includes:

- The percentage completion for each analysis.
- Elapsed and CPU times for each analysis.
- The POP convergence found for each pass through the POP process.
- New topology information. A new topology is a unique circuit configuration, for example, in this buck converter, there is a new topology when the MOSFET turns on, and another when the MOSFET turns off. You will learn more about new topologies in section 2.0 Transient Analysis Settings.

The SIMPLIS Status window offers a peek into how the POP algorithm works. Shown below is the output from the POP simulation run.You can view the status window text as a file in a new browser window by clicking 1.0.5_simplis_status_window_pop_analysis.log:

```
********************************************************************************
********************************************************************************
simplis VERSION 8.10, RELEASE Rel-17.10.3, Mar 21, 2017
Checking syntax of ``1.2_SIMPLIS_tutorial_buck_converter.deck''

New topology #1
New topology #2
New topology #3
New topology #4
New topology #5
New topology #6

A starting operating point located.
Elapsed time   :  0 hr  0 min  1    sec
CPU time       :  0 hr  0 min  0.06 sec
Simulation time: 0.000000000000e+000 sec
```

```
PERIODIC OPERATING-POINT ANALYSIS




New topology #7
New topology #8
New topology #9
New topology #10
New topology #11
New topology #12
New topology #13
New topology #14
New topology #15
```

After each pass through the POP algorithm, the pass number and the measured convergence is output to the SIMPLIS Status Window. Each pass is a complete loop through the POP algorithm as described above. The final convergence for this circuit is 2.45E-13%. SIMPLIS routinely solves circuits to this level of accuracy, which as you will see in the next section, allows you to run an AC analysis on the time-domain model.

This topic is an overview of the POP analysis. You will learn the details of the POP algorithm in 2.2 How POP Really Works.

Conclusions and Key Points to Remember

- The reduction in time to reach steady-state using the POP analysis greatly reduces the time required in the design iteration process.
- The POP algorithm only works if the circuit is switching in a periodic fashion.
- The SIMPLIS PWL circuit equations are solved to a very high degree of accuracy. The POP convergence spec is many orders of magnitude smaller than the relative tolerance (RELTOL) of a SPICE simulator.

Advanced SIMPLIS Training

Search

1.0.6 AC Analysis

The SIMPLIS AC analysis analyzes the small signal response of a circuit at the operating point found in the 1.0.5 POP Analysis. This is analogous to SPICE simulators, which finds an AC analysis around a ***DC*** operating point. Switching power converters don't have a DC operating point, so the SPICE AC analysis cannot be used on a time-domain switching power converter.

In this topic:

Key Concepts

This topic addresses the following key concepts:

- The SIMPLIS AC Analysis is a time-domain analysis.
- The AC analysis results are valid at the switching operating point found in the POP analysis.

What You Will Learn

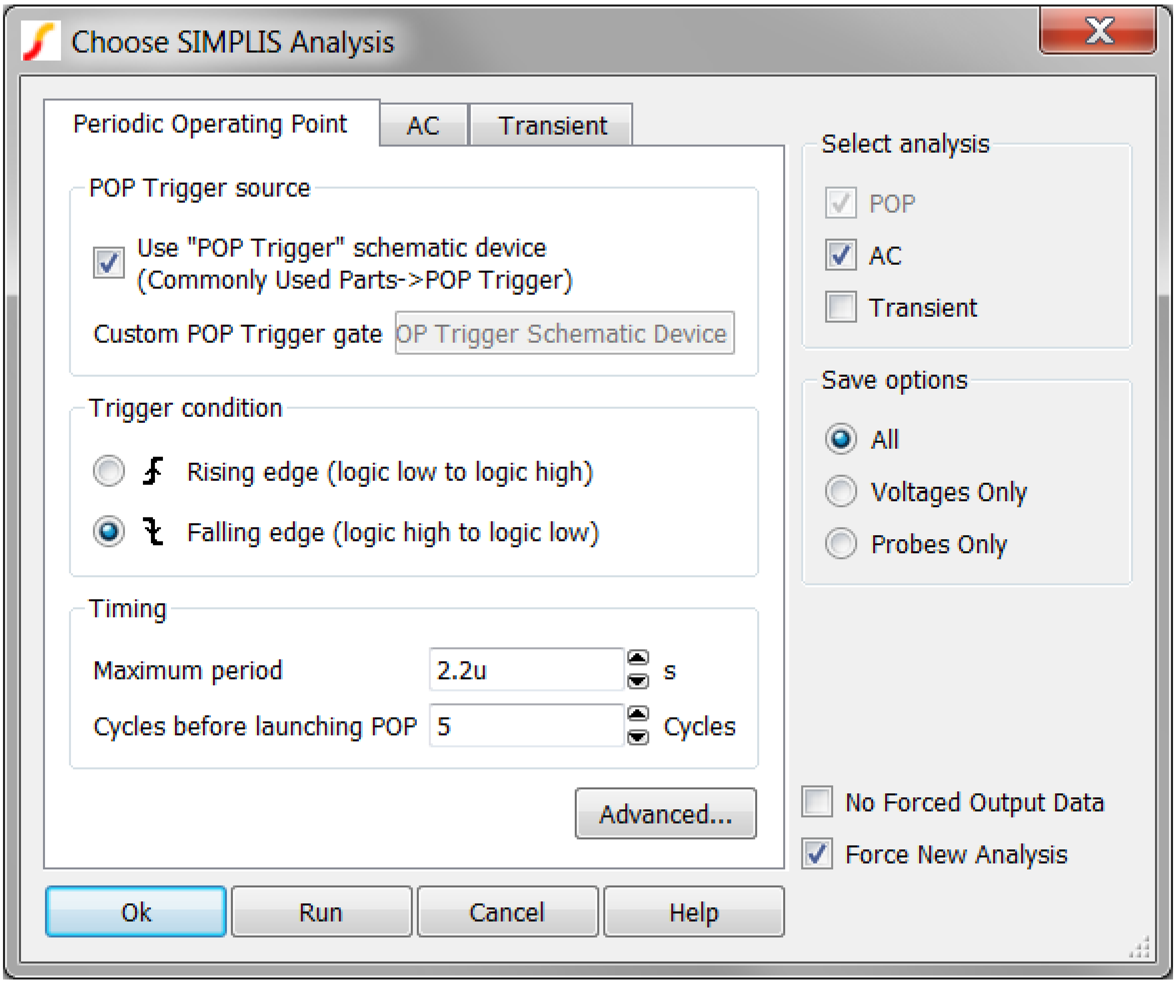
In this topic, you will learn the following:

- How SIMPLIS simulates the AC response of a time domain model.
- The difference between the AC results on a time-domain model and an averaged model.

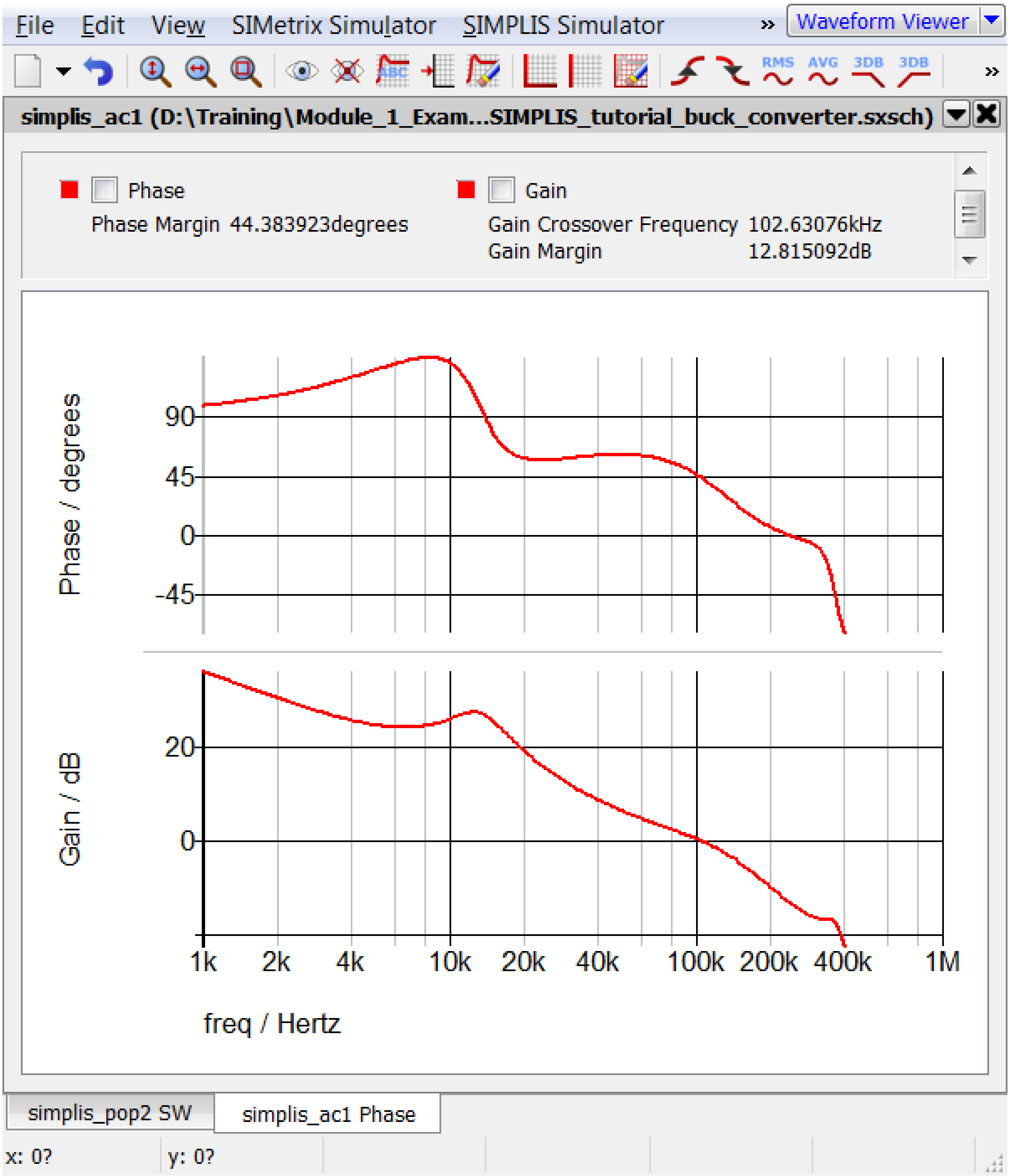
Getting Started: Running an AC Analysis

1. If the waveform viewer is open, close it.

2. Open the schematic titled **1.2_SIMPLIS_tutorial_buck_converter.sxsch**.
3. From the menu bar select **Simulator** > **Choose Analysis...** .
4. Un-check all analysis check boxes, and check the **AC** analysis checkbox.
   Result: *The POP Checkbox is also checked, but disabled - indicating you must run a POP analysis before every AC analysis.*
5. The dialog should appear as follows:



6. Click **Run**.
   Result: *SIMPLIS runs a POP analysis followed by an AC analysis on the Self Oscillating Converter. As with the last section, the waveform viewer opens with 3 switching cycles of data, but also with the AC response of the control loop.*



Discussion

When you go into the lab and connect your switching power system to a network analyzer, you are measuring the AC response of the circuit in the time domain. Your circuit doesn't suddenly change to a averaged small signal model when the network analyzer is connected to it. The SIMPLIS AC analysis works exactly in the same way; it performs an AC analysis on a time-domain simulation model.

How the SIMPLIS AC Analysis works

1. You run a POP analysis on the circuit. This analysis finds the large signal steady state operating point of the circuit.
2. You run an AC analysis on the circuit using the POP analysis as the operating point. For each frequency in the sweep, the AC analysis does the following:
    a. Sets all AC sources (for AC analysis) to a common frequency - the analysis, or perturbation, frequency.

The analysis frequency is **not** the switching frequency where the POP analysis took place - it is the frequency where the small signal analysis is run.

b. These AC sources are time-domain sinusoidal, and the amplitudes of each source is set to an infinitesimally small number.
c. SIMPLIS simulates the time domain response of the circuit to the AC perturbation.
d. Using Fourier methods, SIMPLIS extracts the small signal response of the circuit from the time-domain data.

During an AC sweep, steps a-d above are repeated for each frequency in the sweep. The circuit is perturbed by a single frequency, and that frequency is stepped.

What Can Go Wrong?

1. If the circuit does not POP successfully, in other words, if SIMPLIS cannot find a stable steady-state periodic operating point, the AC analysis will not be run. A warning message appears in the command shell.
2. Your circuit may converge during the POP analysis, but to a periodic operating point which you are not expecting. A common example occurs when the POP analysis results in the circuit operating in a current limit or other fault condition. Since during current limit operation, the voltage loop is essentially open, the gain of the voltage loop is greatly attenuated compared to what it would be in normal operation..

Conclusions and Key Points to Remember

- The SIMPLIS AC analysis works just like a network analyzer in the lab.
- Every AC analysis must be preceded by a POP analysis.
- The AC results are totally and completely dependent on the operating point found during the POP analysis.
- The AC analysis is performed on the time domain model - including all ripple effects.
- Since a small signal averaged model is not required, design time is reduced and only one time-domain model is required.