A Tutorial Introduction to Elements

The Free SIMPLIS Demonstration Version

Christophe Basso

Business Development Manager

IEEE Senior Member

November 2022 – Rev 0.34

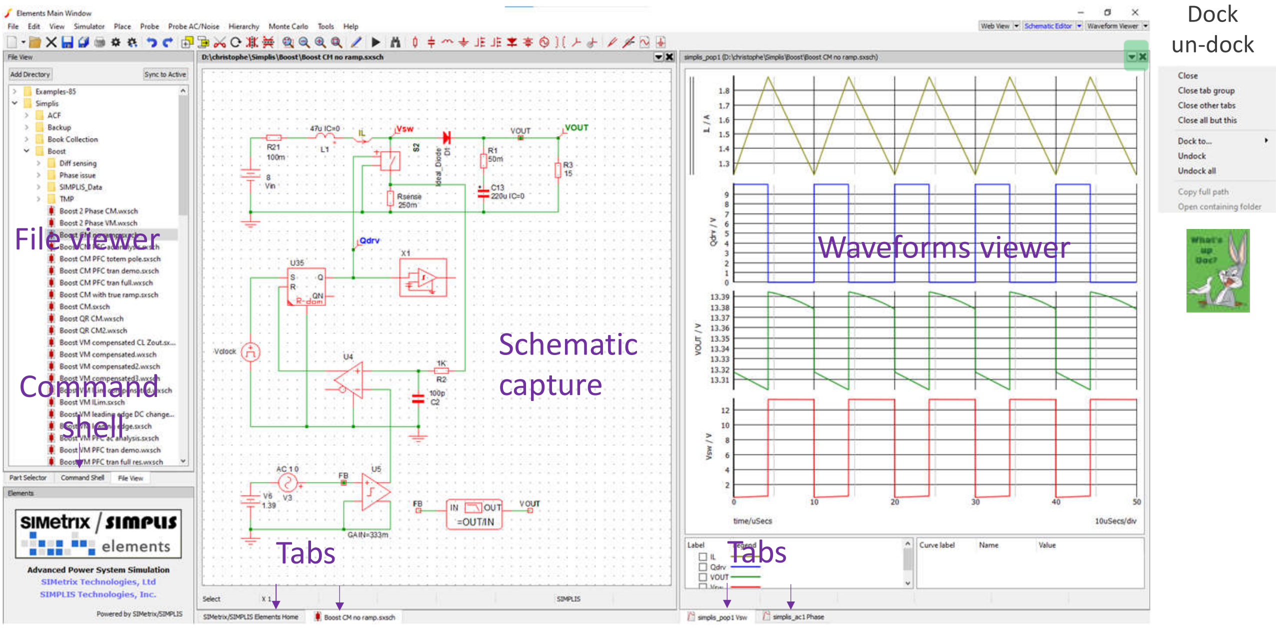
This presentation offers an overview on how you can make your first steps with SIMPLIS Elements, the free demonstration version

Agenda

- What is Elements?
- Running a Basic Simulation
- A Buck Converter
- Ac Linear Analysis with SIMPLIS
- Importing a SPICE Model
- The Ready-to-Use Template

This tutorial will give you basic information on how you can take advantage of the SIMPLIS demonstration version Elements.

Elements is the Demonstration Version of SIMPLIS



The environment consists of different windows all aggregated under a common interface. You can of course undock the schematic viewer and the capture program to have them individually located in different screens. The file viewer gives a direct access to your files and examples and can be configured to make it point where you want to. The command shell is the place to read messages from the engine like syntax or simulation errors.

Download and install the demonstration version here:

https://www.simplistechnologies.com/product/elements

Home

SIMetrix/SIMPLIS Elements

SIMulation of Piecewise LInear Systems

SIMetrix/SIMPLIS Elements is a free-to-download version of our software that offers full schematic capture and waveform viewing/analysis capability along with a host of documentation and training materials designed to help users get up-to-speed quickly with SIMetrix/SIMPLIS' simulation capabilities.

The latest version of SIMetrix/SIMPLIS Elements is **v8.50d**, released on October 5th, 2021.

SIMetrix/SIMPLIS Elements Installer (~109MB) DOWNLOAD

Elements is a free version of the program with no license or copying restrictions. Virtually all features are enabled but a circuit size limit applies. The limits for the Elements versions are generous enough for them to be used for real work and we are happy for users to do so.

System Requirements

SIMetrix/SIMPLIS Elements requires Windows 10 **64 bit edition** (Home, Pro or Enterprise).

SIMetrix/SIMPLIS Elements Limitations

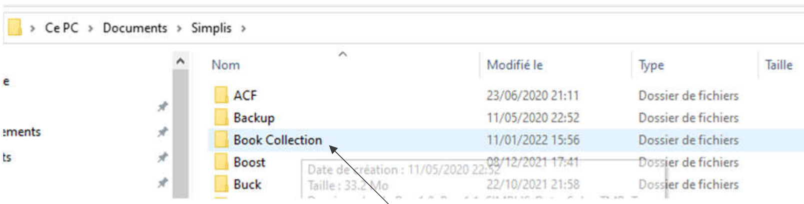
The program can be downloaded from the SIMPLIS website and installed on your computer. The program occupies less than 300 Mo of space on the disk.

FUTURE ELECTRONICS

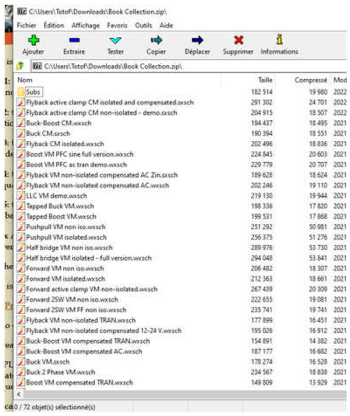
Then go to my webpage and download the ready-made templates:

http://powersimtof.com/Downloads/Book/Book%20Collection.zip

Extract the files and place them, for instance, in a SIMPLIS directory created in <My Documents>



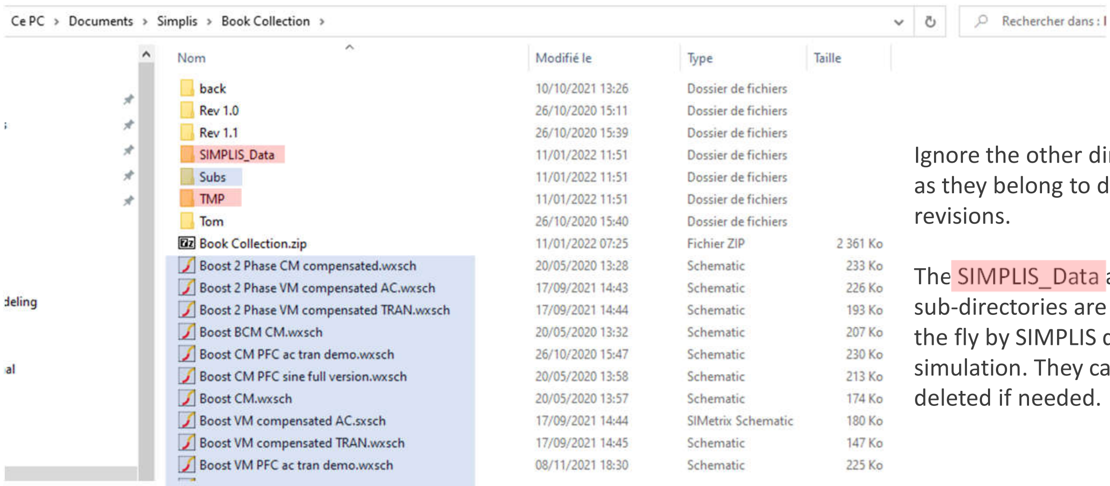
Extract the ZIP here



These files are examples I documented in my last book
Transfer Functions of Switching Converters – Faraday Press

Once the program is downloaded and installed, you can obtain the demonstration files from my webpage. These files can be extracted anywhere but having a SIMPLIS sub-directory under the Documents directory is a good option. Theses files are ready-made templates for switching converters, working for most of them with the demo version.

Once installed, you have this structure:



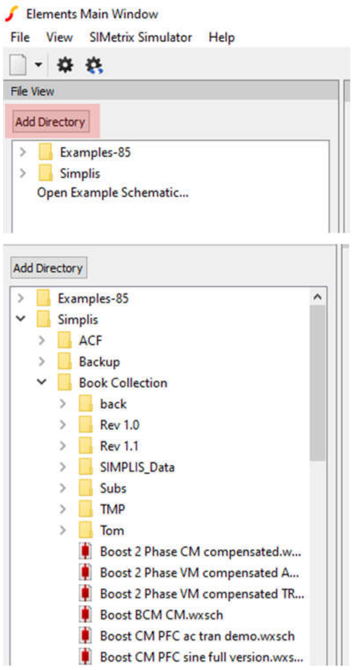
Ignore the other directories as they belong to different revisions.

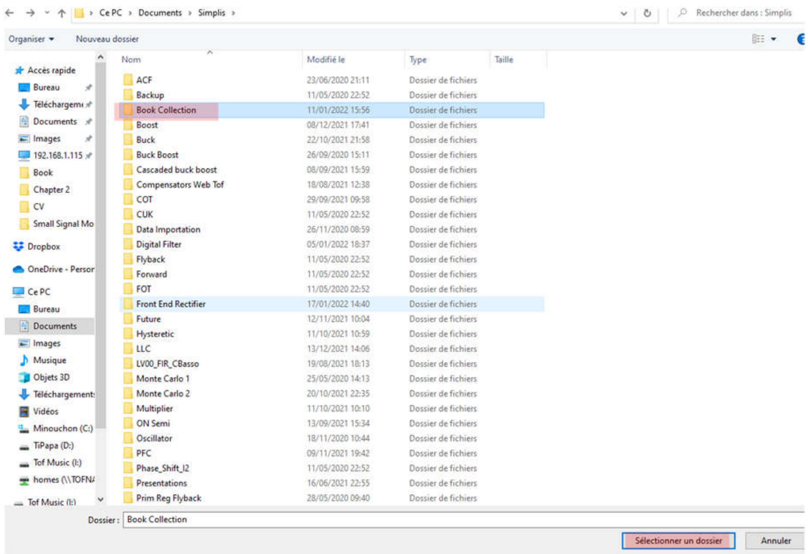
The SIMPLIS_Data and TMP sub-directories are filled on the fly by SIMPLIS during simulation. They can safely be deleted if needed.

I do not recommend to launch SIMPLIS by double-clicking on the file of interest. Open SIMPLIS and select the file of your choice instead.

SIMPLIS uses two directories in which it stores simulation data files, SIMPLIS_Data and TMP. The directory Subs stores the subcircuits intended to be used with the examples.

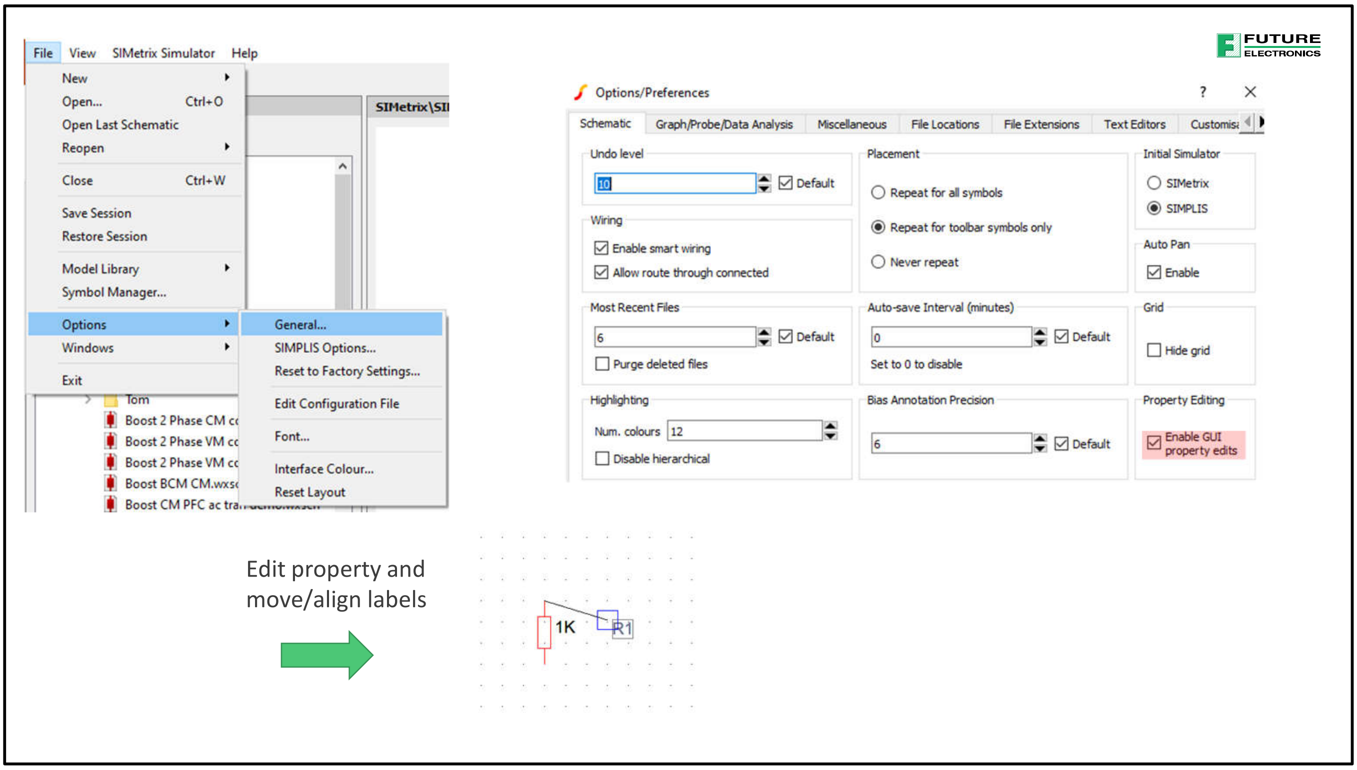
Once SIMPLIS Elements is installed, instruct the file viewer with the directory location:





Then you can pick the file of your choice via this File View tab

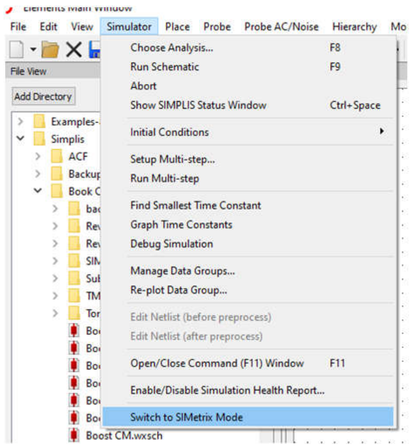
To have a quick access to your simulation files, press the Add Directory button and select the directory or directories you would like to see added to the left pane for an easy and quick access.



I like to properly move and align components labels so the first thing to do is instruct Elements that you want to do this. Go to File>Options>General and enable the GUI (graphic user interface) edits.

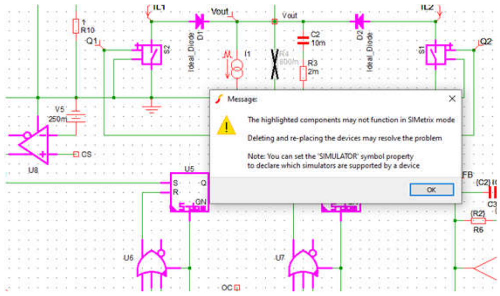
The program Elements gives access to two simulators:
SIMPLIS and SIMetrix

1. SIMetrix is a classical SPICE-based program and is compatible with many SPICE engines. It can perform ac and transient analyses as any SPICE package would do.
2. SIMPLIS is exclusively a time-domain simulator which can also simulate in ac and transient modes. SIMPLIS can extract the ac response from a switching circuit what SPICE cannot (easily) do.



You can switch between the two engines at any time

Some components are not compatible with both engines and the program warns you

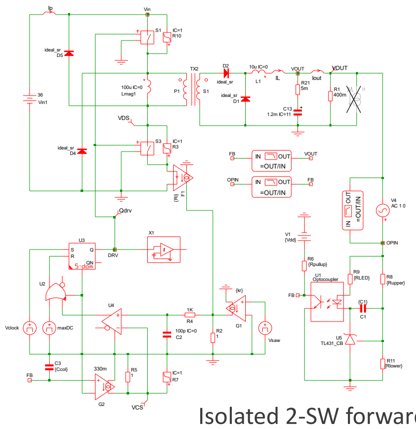


The program embeds two simulation engines, SIMetrix and SIMPLIS. SIMetrix is a SPICE-based engine and uses a different syntax than SIMPLIS. You can toggle between the two engines, but differences or incompatibility exist between components or syntax.

SIMPLIS Circuit Size Limits

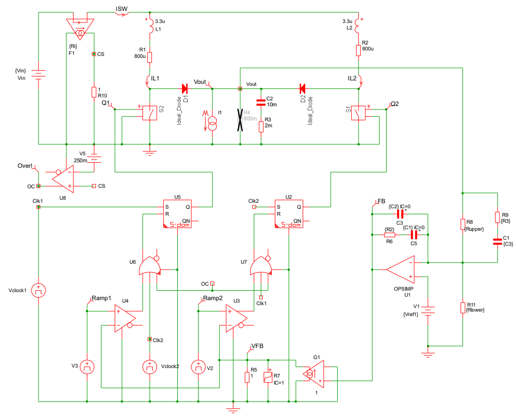
The exact limits for the SIMPLIS simulator are:

1. A total of no more than 15 state variables. A capacitor or an inductor each requires one state variable. Each time-varying or small-signal AC source requires one state variable, with the exception of SINusoidal or COSinusoidal sources, which require two state variables per source.
2. A total of no more than 10 capacitors or inductors combined.
3. A total of no more than six switches, simple or transistor.
4. A total of no more than six logic gates.
5. A total of no more than 26 "states." Each PWL element requires one state. Each switch requires one state. Each time-varying source requires one state. Each logic gate requires one state.
6. A total of no more than 100 new topologies. 100 topologies will be enough for simple switching circuits that use simple models only. More complex circuits or circuits that have more complicated models may exceed this limit



Isolated 2-SW forward

These are comfortable limits which let you simulate quite complex structures



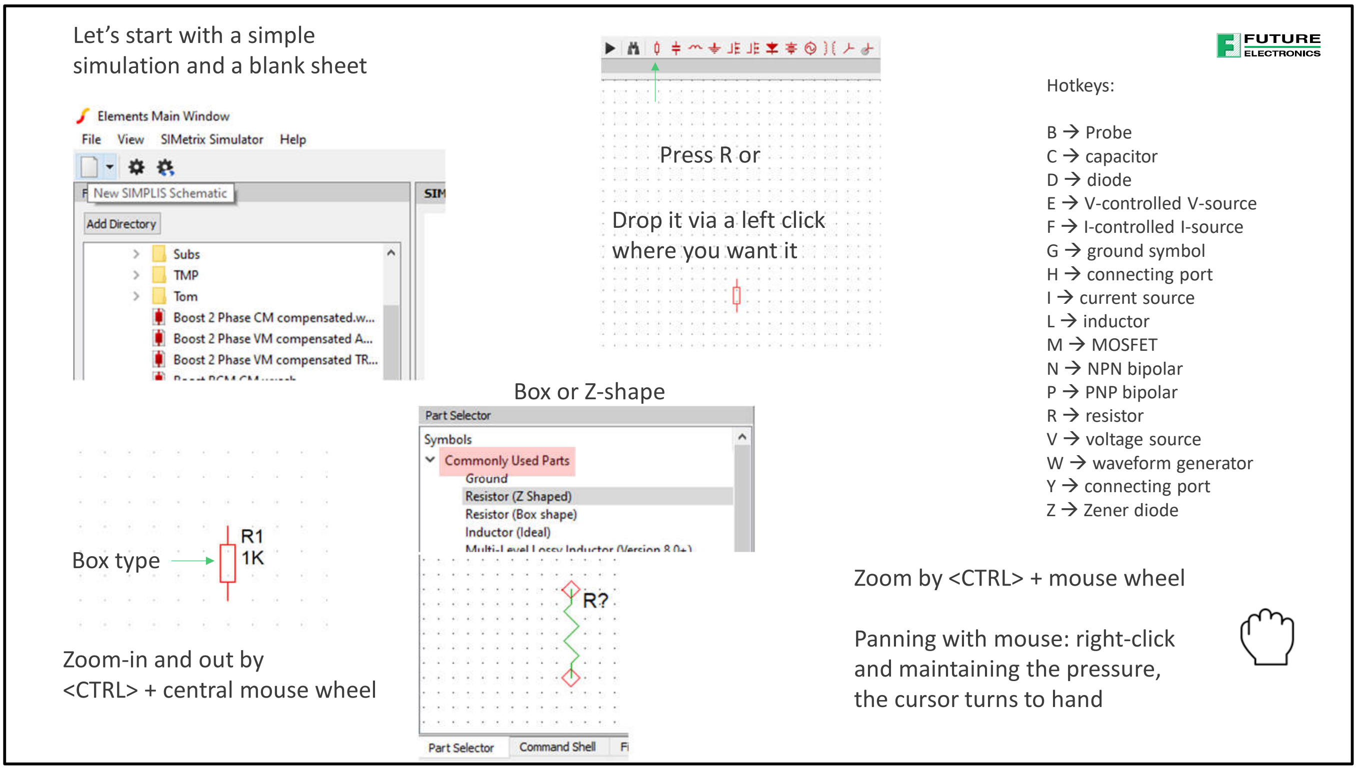
Compensated 2-phase boost

The demonstration version is limited in capabilities but, nevertheless, many interesting switching circuits can be simulated with, or without isolation as shown in the above examples.

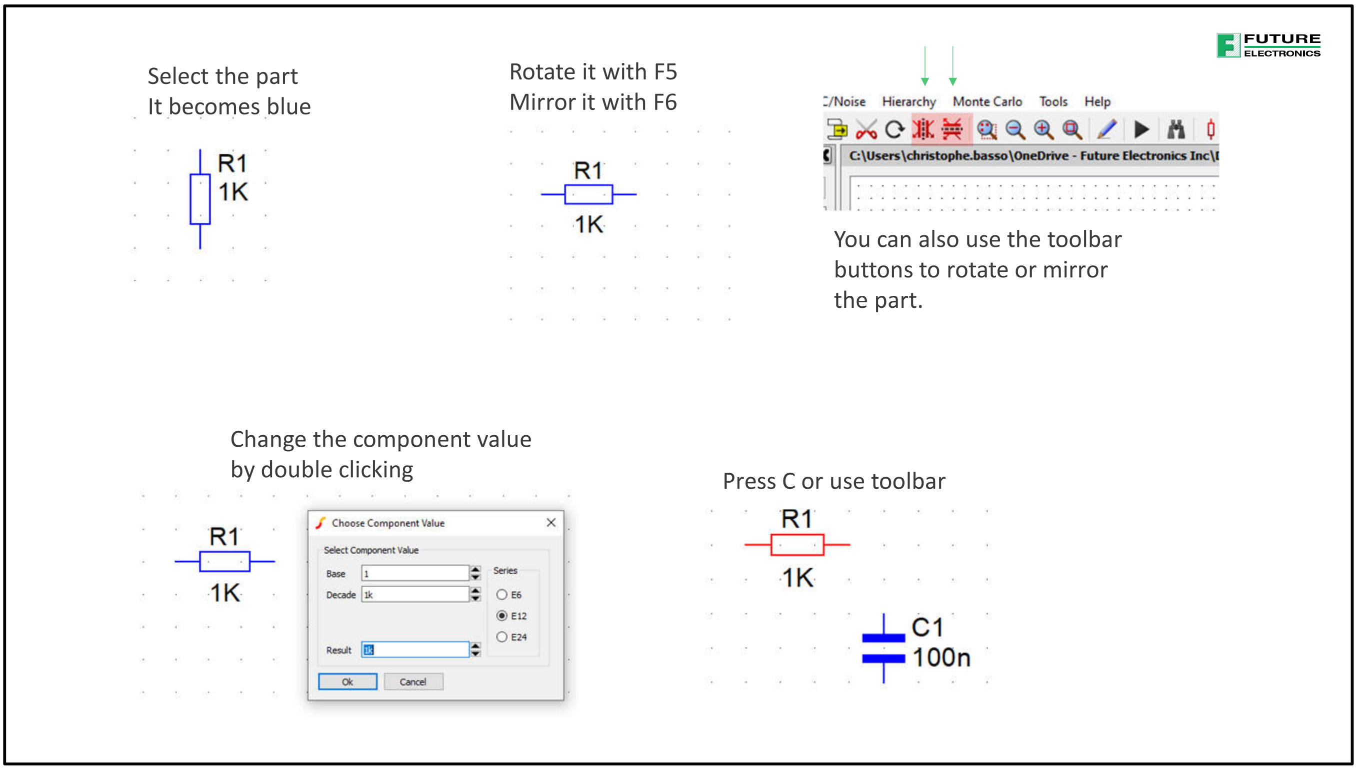
Agenda

- What is Elements?
- Running a Basic Simulation
- A Buck Converter
- Ac Linear Analysis with SIMPLIS
- Importing a SPICE Model
- The Ready-to-Use Template

Let’s now see how to smoothly start using Elements

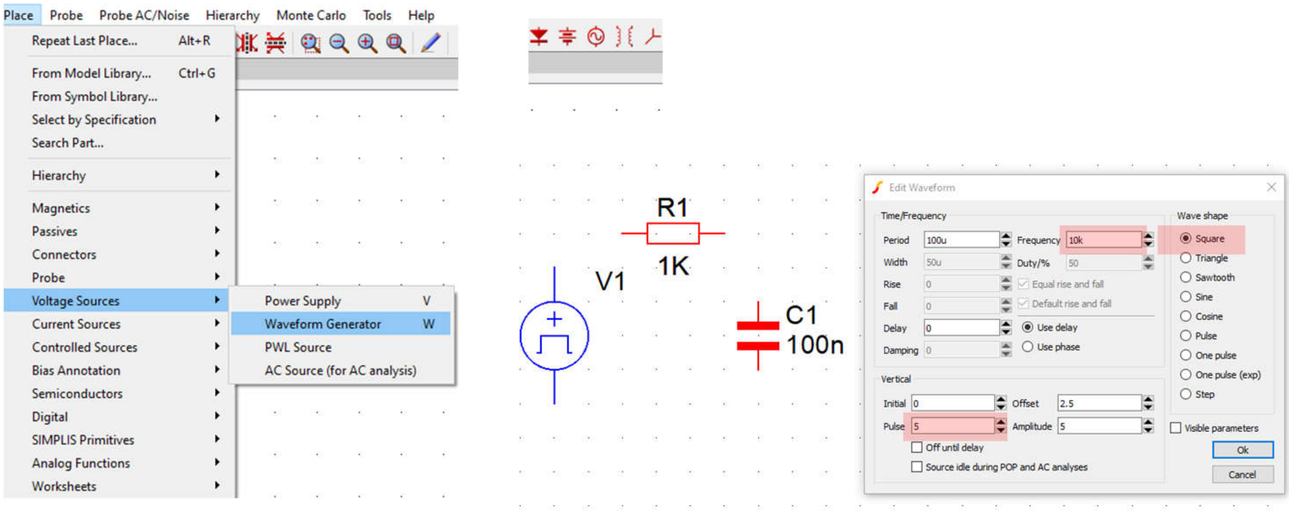


We start by opening a blank sheet and place a resistor. Go to the ribbon or press R on the keyboard. You obtain a boxed-shaped resistor. If you want a Z-shape instead, go to Part Selector in the left panel and in the Commonly-Used Part menu, pick the Z-shaped symbol. Zooming on the schematic is done by pressing the CTRL key and moving the mouse central wheel.



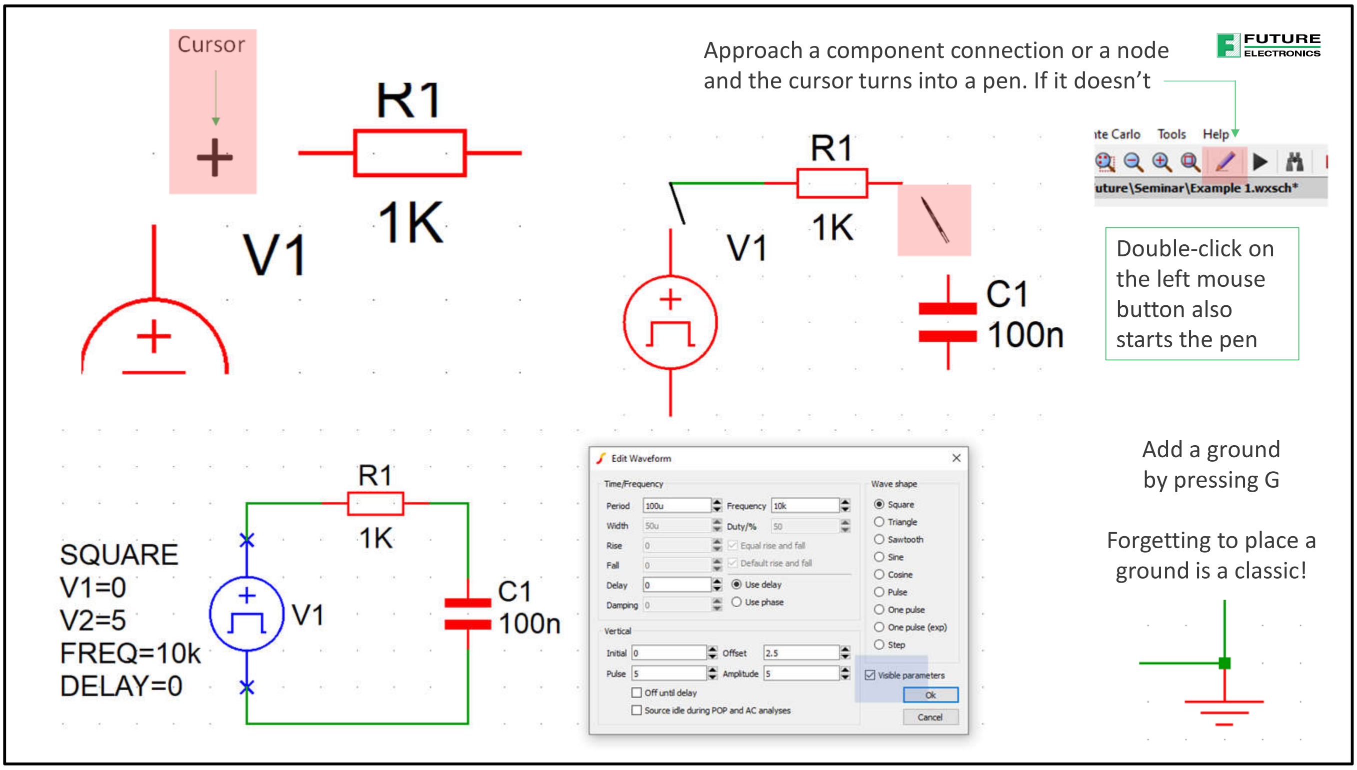
When you hold the component while placing it, before dropping it on the schematic, you can adapt its position via F5 and F6 to rotate or mirror it.

Add a waveform generator – press W or pick-up in the toolbar

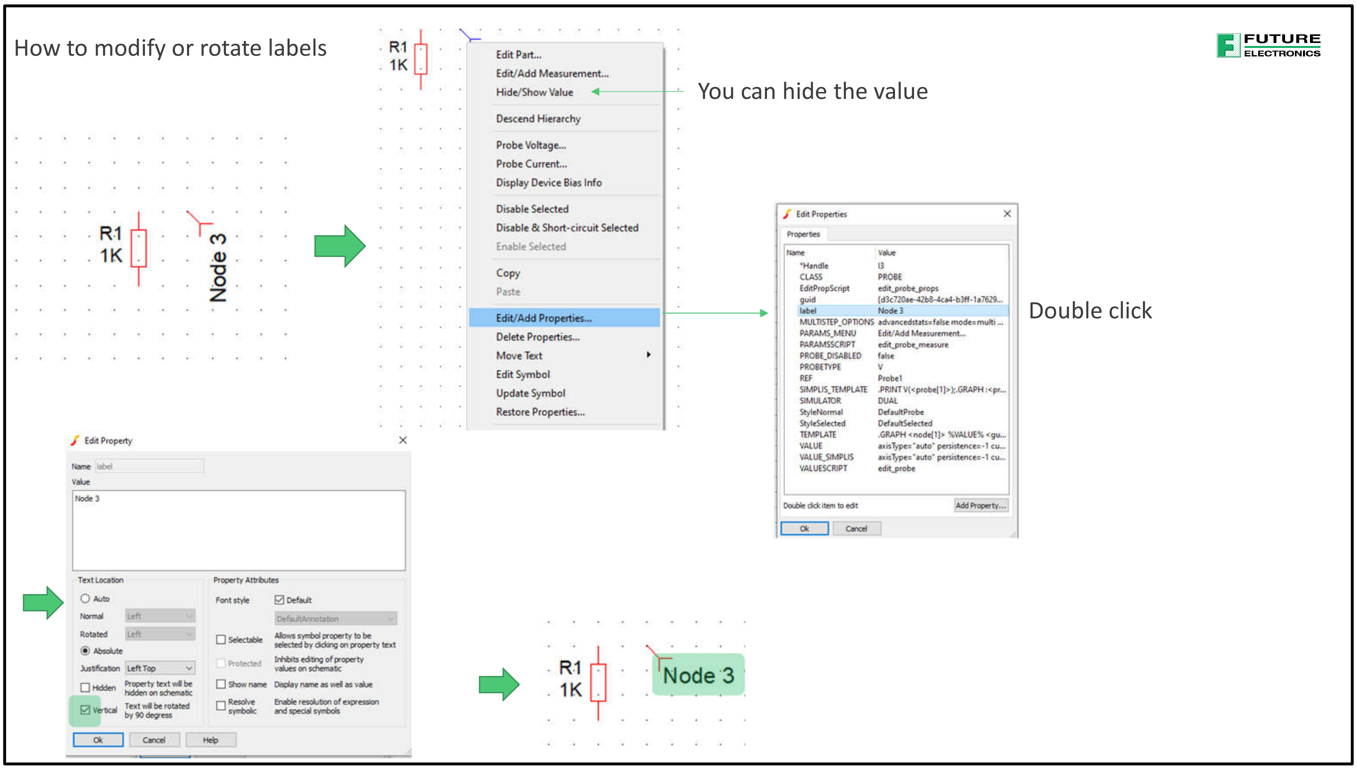


Double click

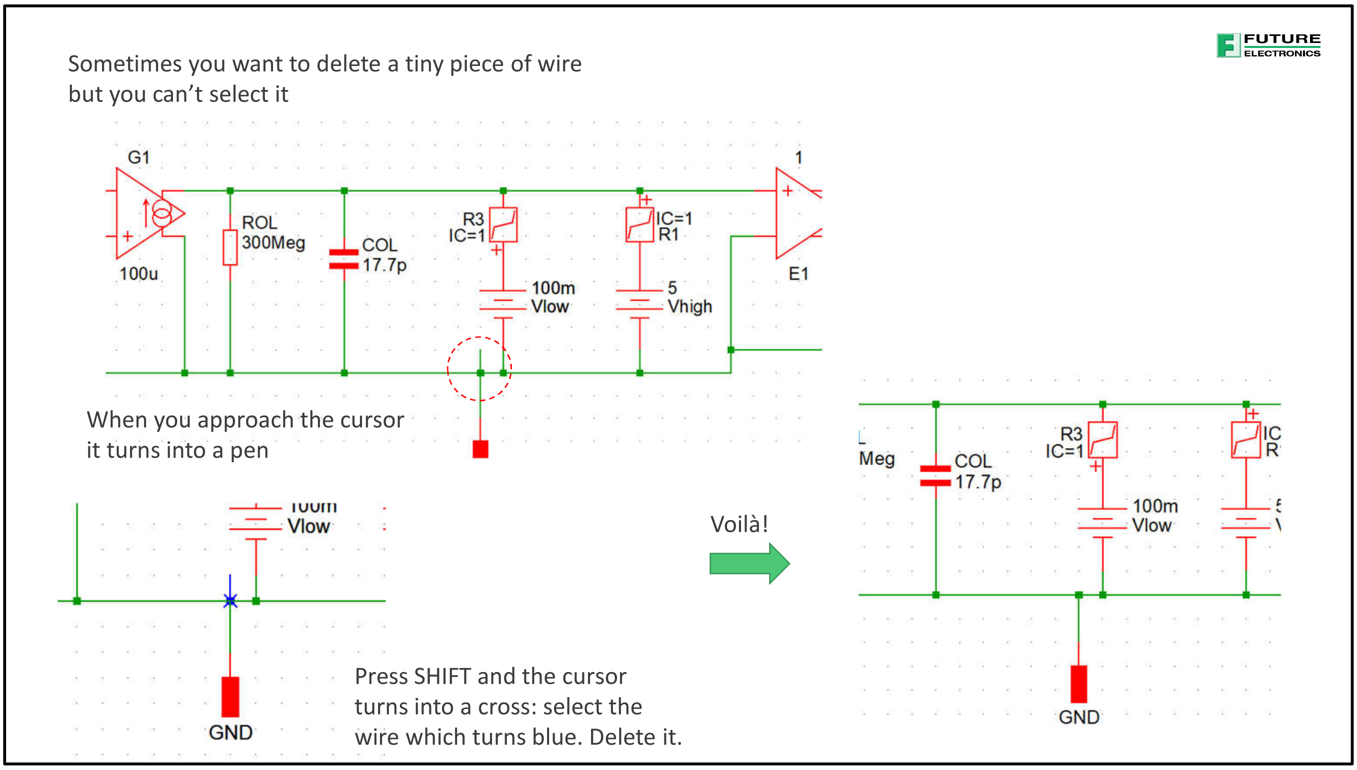
Waveform generators can be added with different profiles. Press W from the keyboard or activate the pull-down menu then drop the source on the schematic. By double-clicking it, you have access to the parameters. Enter a 10-kHz frequency (no unit, k is enough) and a 5-V amplitude.



To activate the wiring tool, either approach the cursor from a node and the cursor will turn into a pen or press the pen button on the toolbar. Connect all parts together and do not forget to add the ground (0-V node) by pressing G on the keyboard. Then edit the waveform generator and check "visible parameters" to have the source parameters displayed in the schematic.

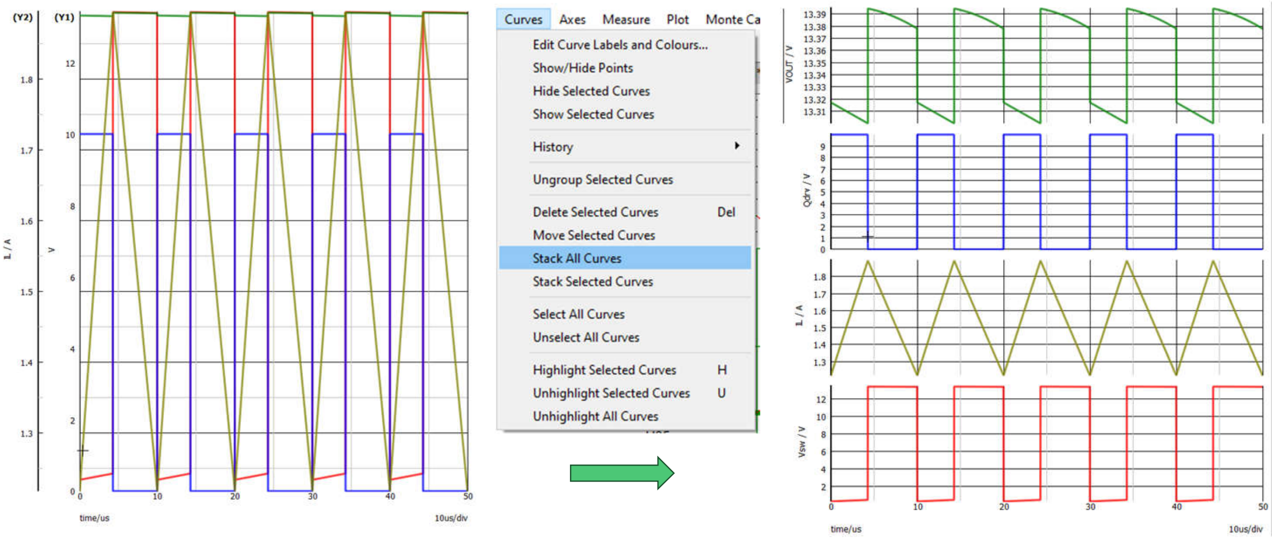


Sometimes, when the part has been rotated, the label is also in a difficult-to-read position and we want to place it back in a horizontal position: right-click and select Edit/Add Properties then locate the corresponding label and tick « vertical » then ok.

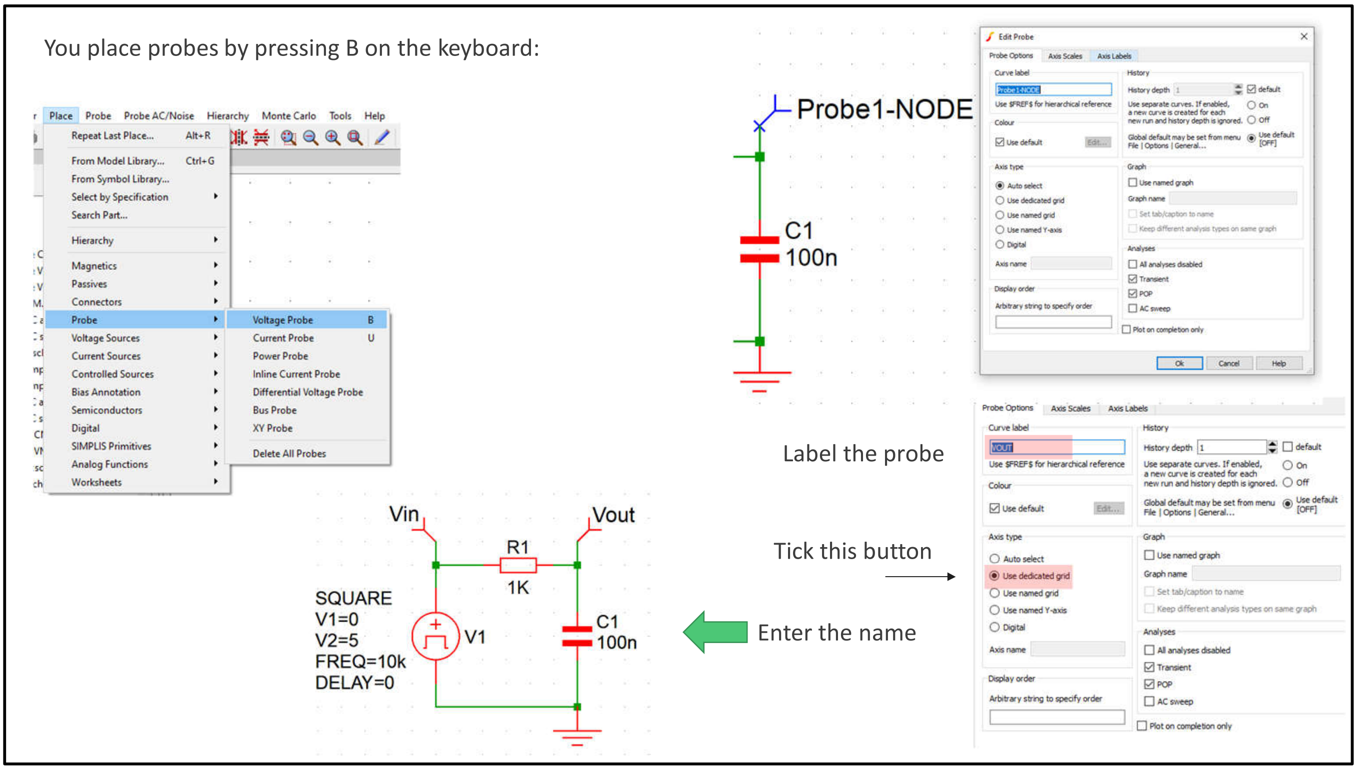


The rubber-banding feature is sometimes difficult to handle when it comes to trim bits of wires: should you want to remove some annoying pieces, the cursor turns into a pen and does not offer you to select the extra wire. Press SHIFT and easily select the wire then delete it.

If you place probes without specifying axis information, by default, the viewer stacks up all the curves, ending in an ugly representation as below. To have them go to their own axis – what it should do by default in my opinion – open the Curves menu and choose “stack all curves”:

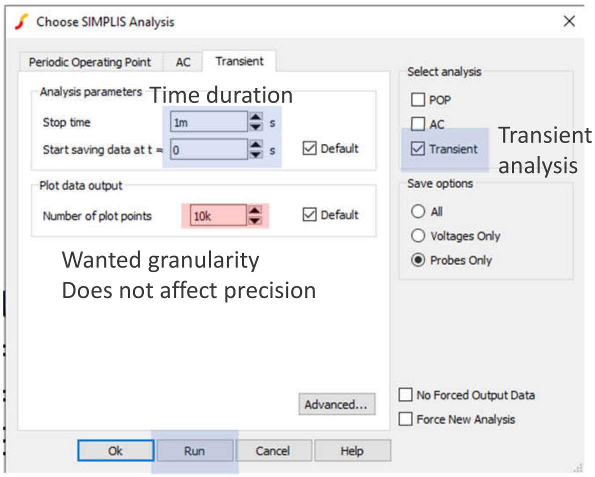


By default, SIMPLIS collects all the generated vectors and stacks them up with a common x-axis, making the final result a complete mess. You can solve this by going to the Curve menu and choose “stack all curves”. Otherwise, the next slide explains how to do it automatically.

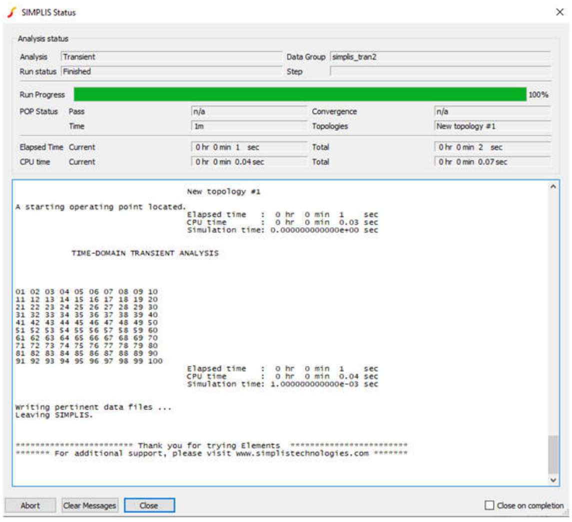


To add classical ground-referenced probes, either use the drop-down menu and select a Voltage Probe or press B on the keyboard. Once placed, double-click on the probe and enter a label to identify the curve when plotted. Then tick Use Dedicated Grid which will name the grid according to the label. Using this approach will assign a grid to each probe and will let you display nicely-arranged waveforms. Repeat this operation for V_{in}. Should you want to purposely plot the curves on the same axis, choose a common axis name. Another option is to select Use Dedicated Grid in case you don't want to bring other curves on the same grid. I tick Use Name Grid for all my simulations.

Choose the simulation parameters – press F8

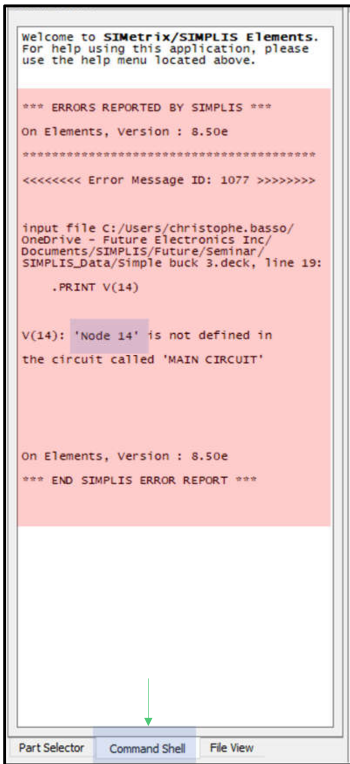


Once the circuit is saved, go!



When everything is ok, the complete windows pops-up

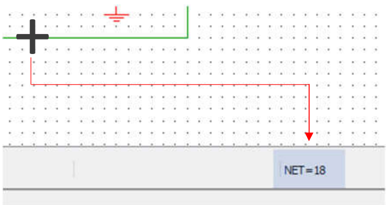
It is now time to invoke the simulation panel where you will select the simulation length. Enter a 1-ms duration with a granularity of 10000 points as a start. Please note that the granularity only affects the display and not the simulation resolution which is always the highest.



In case simulation issues occur, look at the Command Shell window.

You can erase the window with <CTRL+A> and SHIFT+DEL

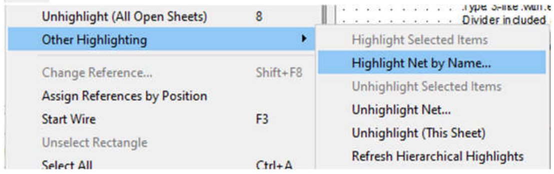
In case the error refers to a net or node number you have to find its position on the schematic diagram.

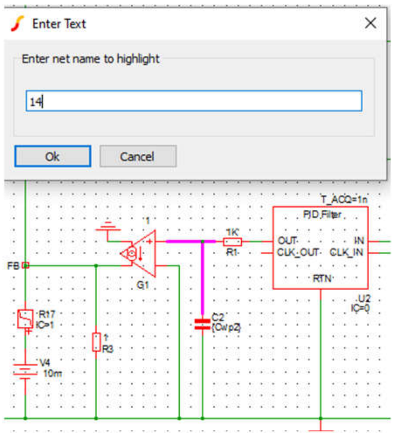


Fly over a (cuckoo's) net and see the number displayed in the low-side bar.

Edit

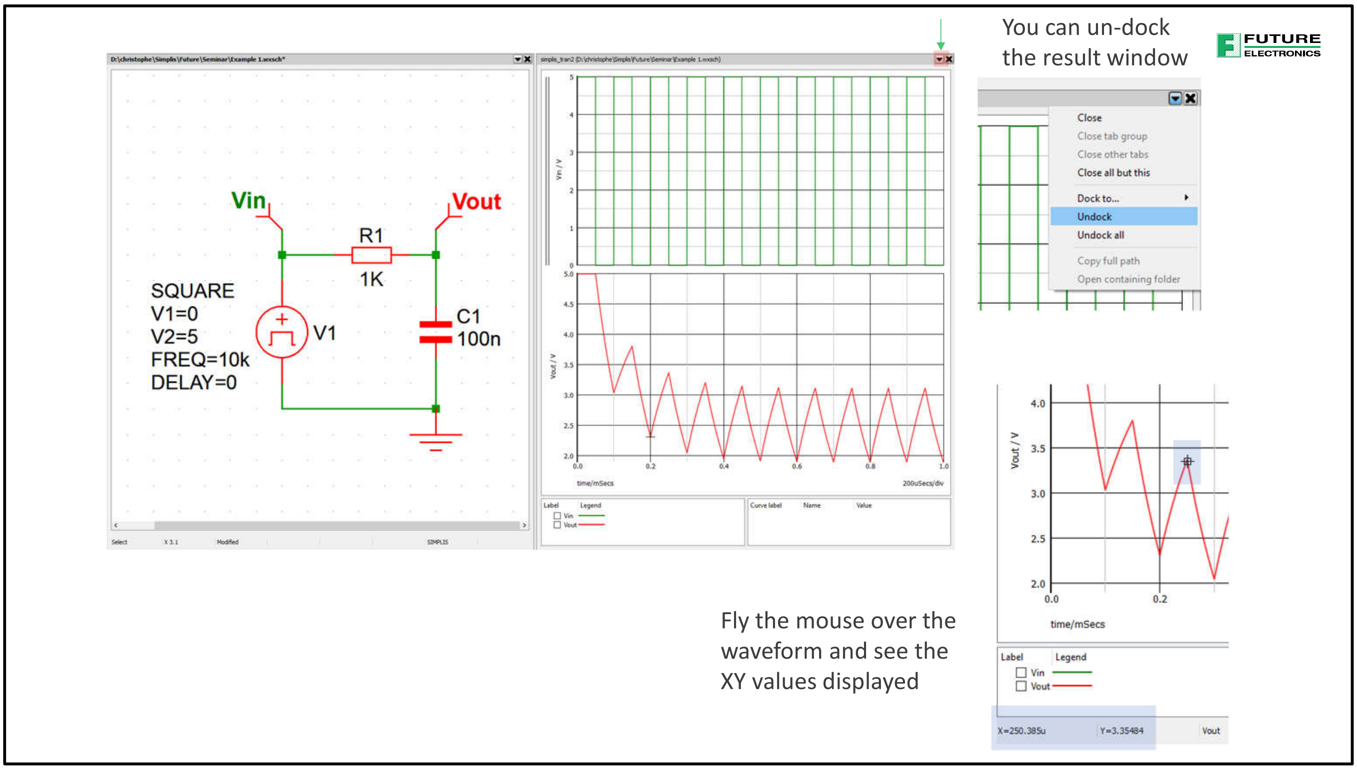
In the Edit submenu



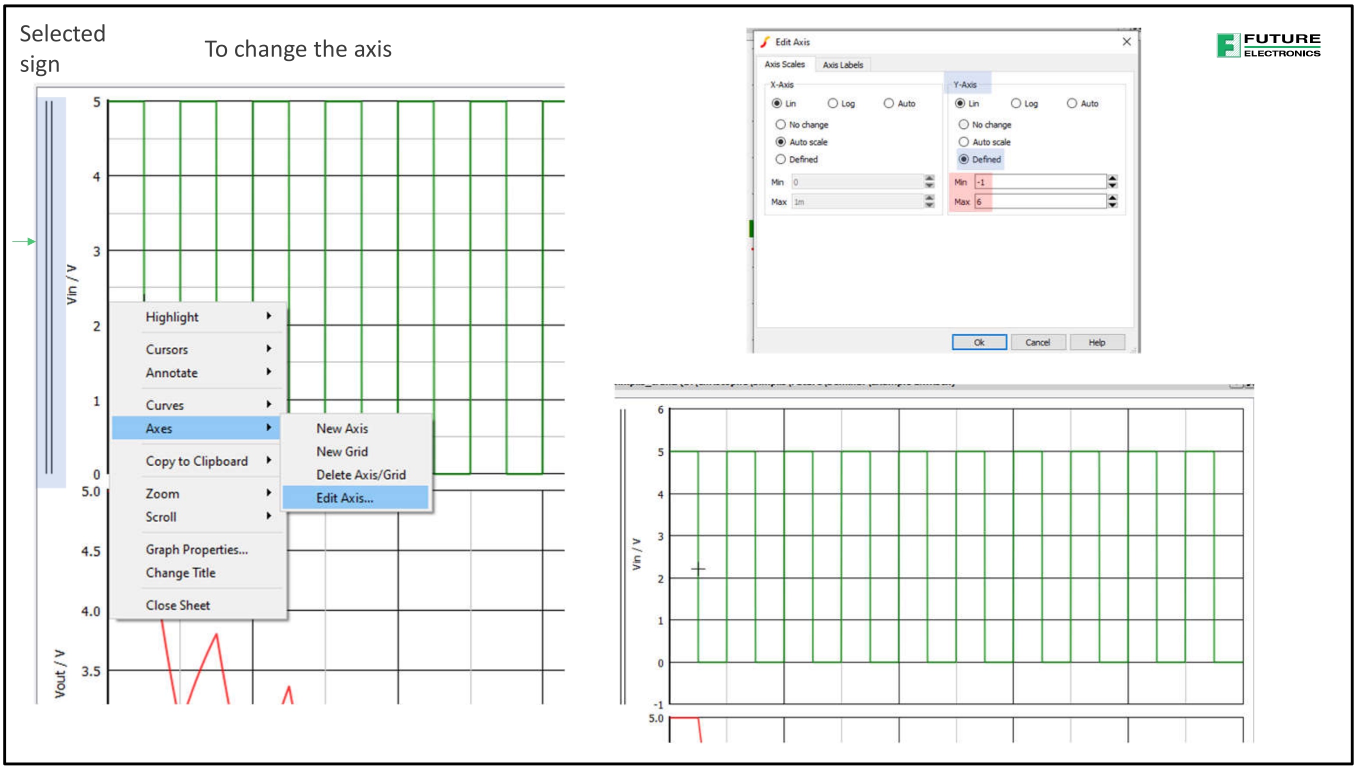


To erase the highlight, press 8 on the keypad

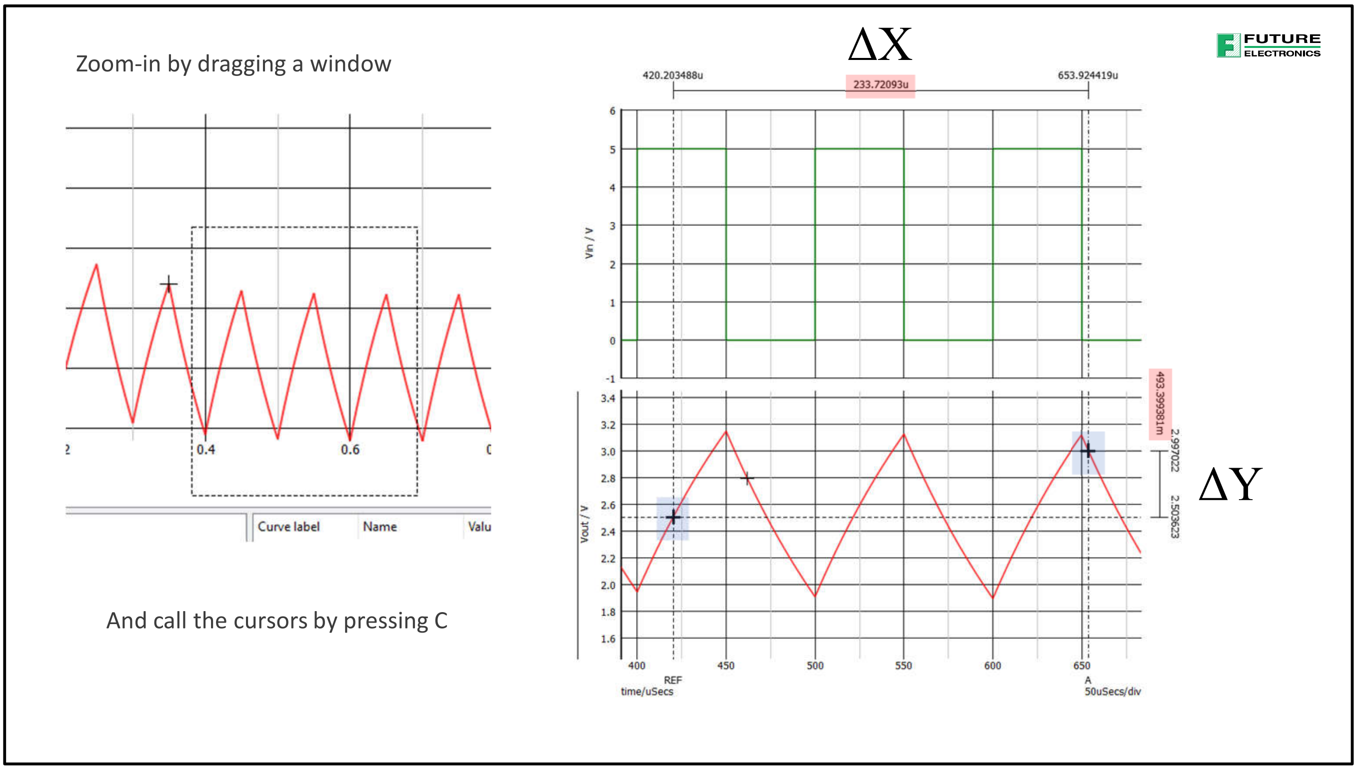
In case simulation errors appear, they will show up in the Command Shell window and tell you what the problem is. If it refers to a node number, you will have to identify the connection on the schematic. Flying the mouse cursor over a net displays the net number in the low-side bar. You can also ask the program to highlight the net corresponding to the node you search: Edit | Other Highlighting | Highlight Net by Name and enter the number. The net is highlighted in the schematic. You can turn the highlight off by pressing 8 on the keypad.



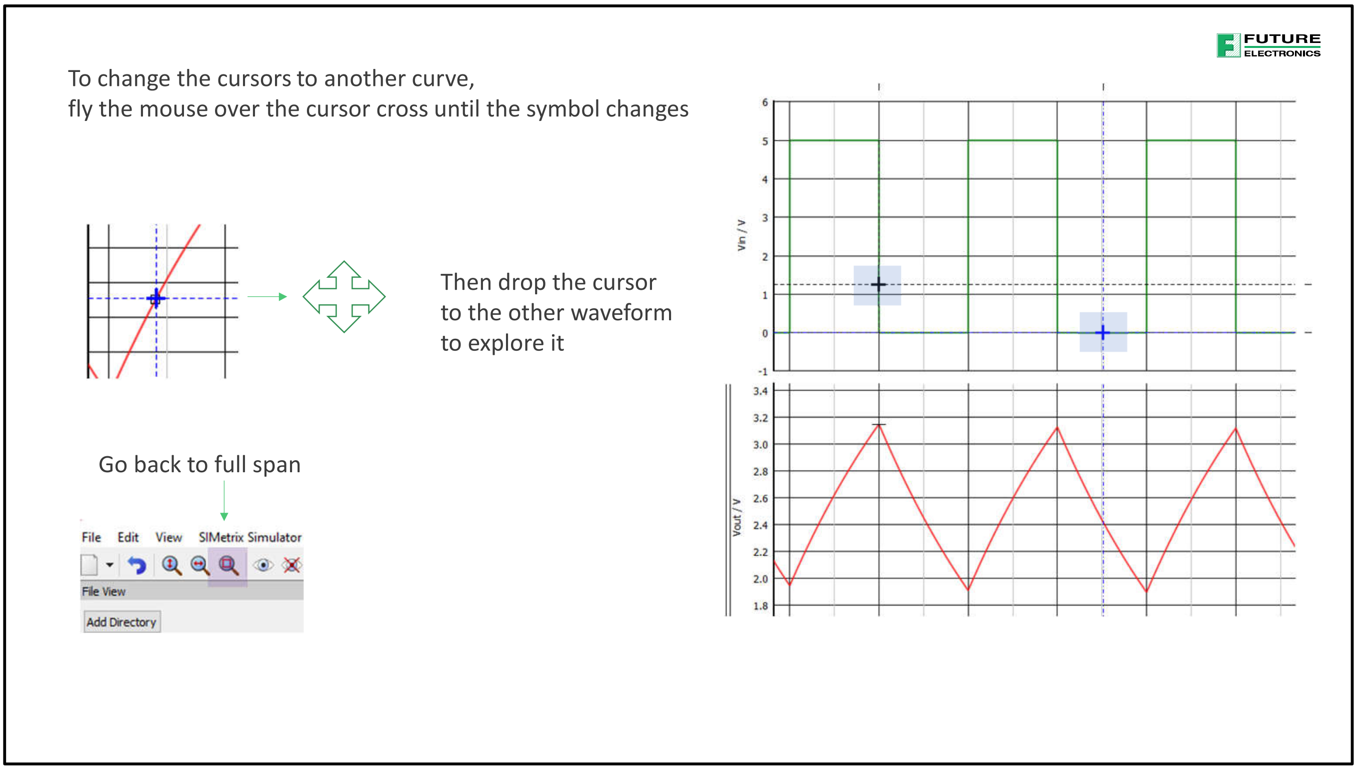
Once the simulation is run, you can see the waveforms nicely stacked in the waveforms viewer. You can undock the viewer and move it to a separate screen if needed. If you fly the cursor over the curves, you see the x-y values displayed in the low-side bard.



You can change the axis to a different scale by right-clicking on the axis and selecting the y section. Enter new values for the min and max fields then press ok.

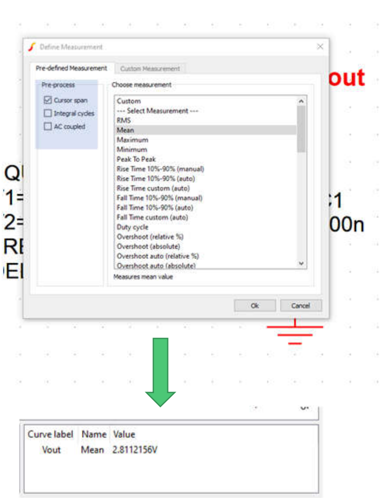


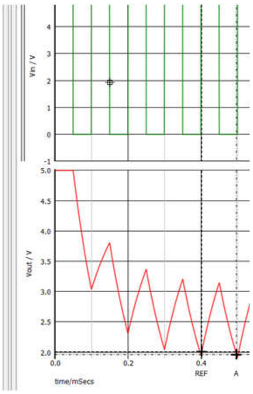
If you keep pressing the left button and open a window on a curve, you zoom in the waveform. Pressing c on the keyboard invokes cursors that display the delta X and delta Y values. You toggle cursors off by pressing c one more time.

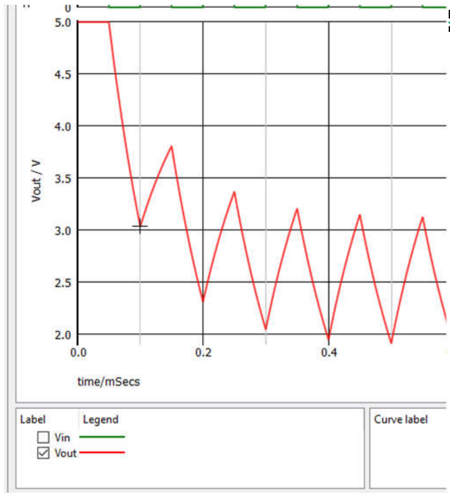


To change cursors to another grid and curve, you'll have to seize each cursor by pressing the left mouse button and turn it into a larger cross. Then drag and drop the cursor to the desired grid. Do it for one or both cursors.

To run a measurement on the waveforms, press F3





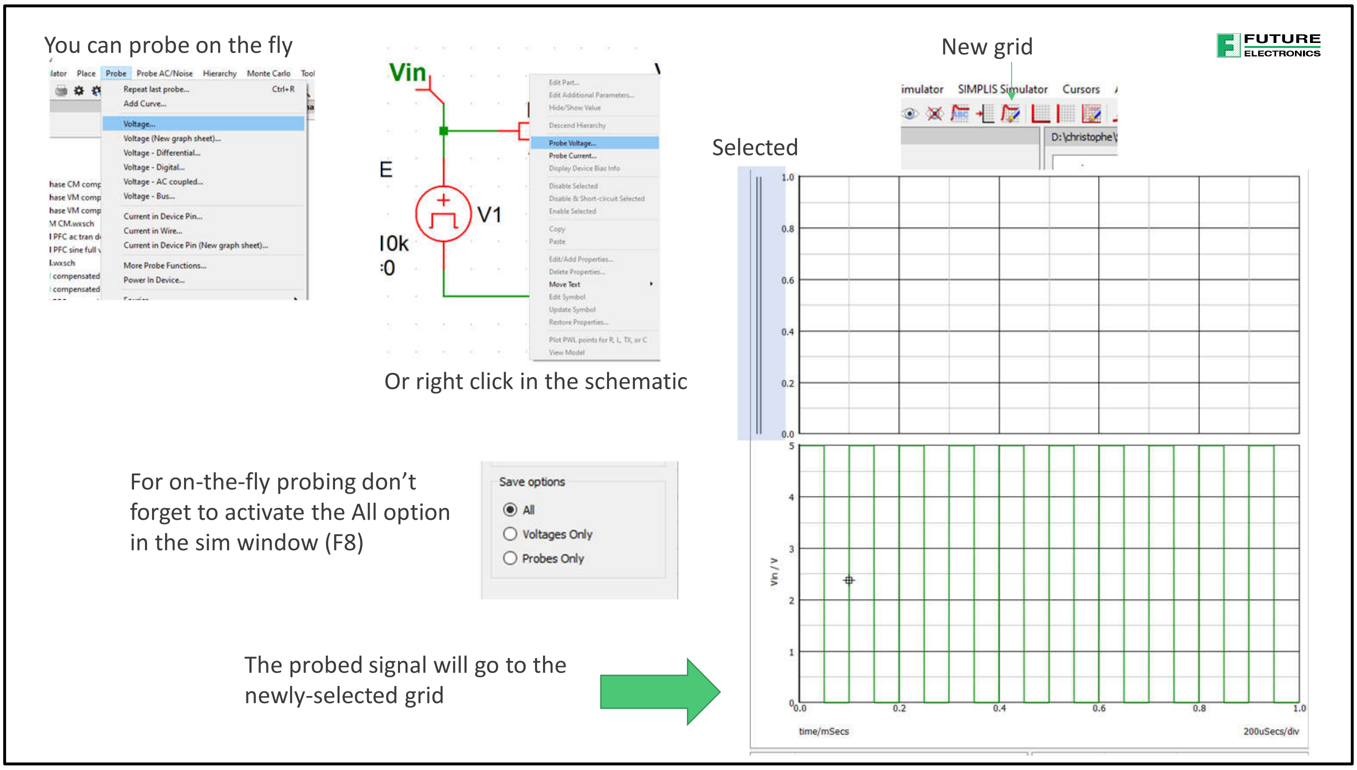


FUTURE ELECTRONICS

To delete a waveform, select it and press delete

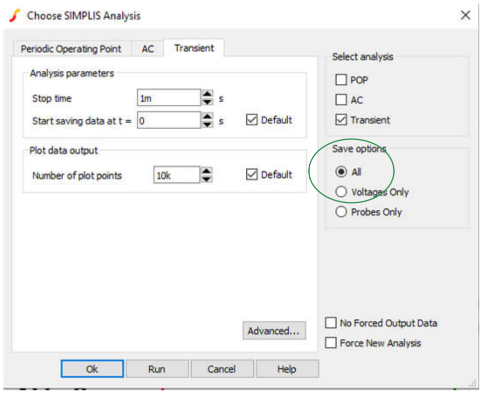
Cursor span	Truncates the waveform data to the span defined by the current positions of the cursors. In other words, the measurement is performed on the range defined by the cursor positions.
Integral cycles	Truncates the waveform data to an integral number of whole cycles. This is useful for measurements such as RMS which are only meaningful if applied to a whole number of cycles.
AC coupled	Offsets the data by the mean value. This is equivalent to 'ACcoupling' the data.

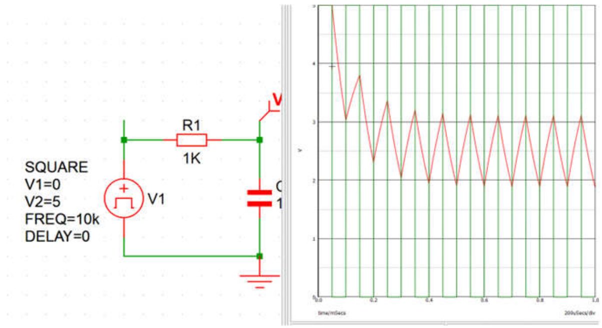
To run a measurement on a curve, press F3 and access the measurement menu to select which data you want like rms, peak, average etc. The measurement is carried either over the truncated simulated points to obtain an integral number of cycles or between the cursors.



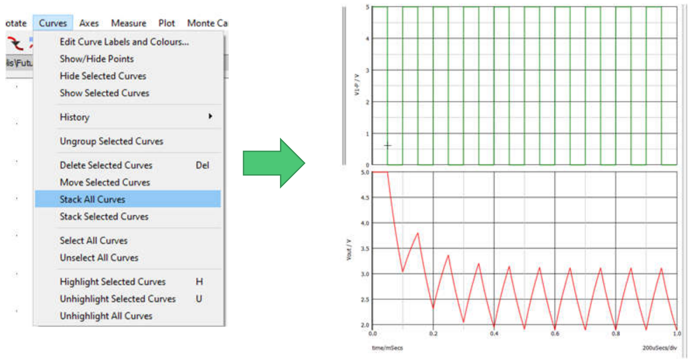
One very convenient option is to probe the schematic on the fly, without associating a fixed probe as we did. Go to the schematic window and right-click to select Probe Voltage or select Probe | Voltage in the pull-down menu. Now, you either create a blank grid to welcome the to-be-probed signal or add the curve to the already-selected grid.

The waveforms are placed on a common grid





You can also select the All option and dynamically probe all main schematics nodes without placing probes first. Ok for debugging but I preferred well-labeled probes. Ticking All does not give access to subcircuits nodes, another statement is necessary.



As shown in the illustration, if you do not create a new grid, then all curves are gathered in the same graph, and it can quickly be messy. Once you have accumulated all the waveforms you want, Stack all Curves in the Curves pull-down menu. Please note that only curves displayed from fixed probes are updated when a new simulation is run. It is not the case – unfortunately – with the dynamically-probed waveforms.

In case you need to cross-probe voltages and currents in a subcircuit, you need to insert one of the below .KEEP combination in the main simulation window. Check the documentation here and watch out when using this command as the data file can quickly be gigantic!

The Concise Guide to .KEEP Statements

The following guidelines will be helpful to select .KEEP statements

Command	Action
.KEEP *V	Keeps all voltages at this hierarchical level.
.KEEP **V	Keeps all voltages at this hierarchical level and all lower levels.
.KEEP *I	Keeps all currents at this hierarchical level.
.KEEP **I	Keeps all currents at this hierarchical level and all lower levels.
.KEEP *I **V	Keeps all currents at this hierarchical level and all voltages at this hierarchical level and all lower levels.
Other combinations are possible using a space separated list of voltage and current declarations.	

CAUTION:

For large circuits, placing a .KEEP **V at the top level will save an enormous amount of data. Using .KEEP **V **I will save even more data, as all currents will be added to the data group.

CAUTION:

When you package your circuit for encryption and distribution, make certain you do not have .KEEP statements in the hierarchy. Any .KEEP statements will output simulation vectors to the data group, possibly exposing some internal intellectual property. This will be discussed in more detail in the 4.4 Protecting Your Intellectual Property - Model Encryption topic.

Press F11 while on the main schematic:

```
.param Rlower={2.5/Ib}
.param fc=40
```

```
1 .simulator SIMETRIX
2 *.tran 1m
3 .ac dec 100 10m 1k
4 .simulator DEFAULT
5
```

Insert as shown

SIMetrix

```
1 .simulator SIMETRIX
2 *.tran 1m
3 .ac dec 100 10m 1k
4 .KEEP **I **V
5 .simulator DEFAULT
6
```

SIMPLIS

```
1 .simulator SIMPLIS
2 .ac DEC 200 10 100k
3 .KEEP **I **V
4 .print
5 + ALL
6 .options
```

Saves all currents and all voltages at all levels

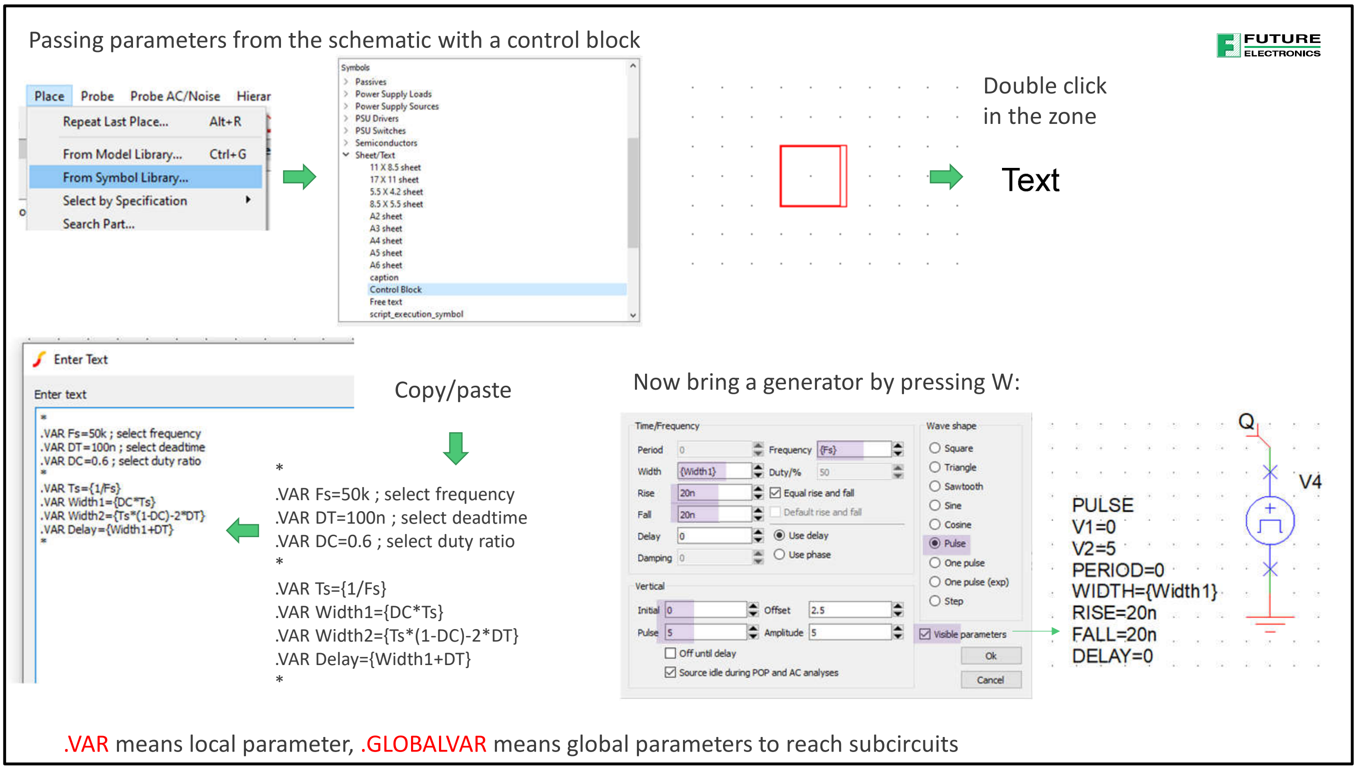
When debugging subcircuits or simply to understand how some parts behave, it can be interesting to cross-probe some of these internal nodes. By default, the engine does not give access to these nodes, located below the main schematic level. You have to insert a specific keyword for storing these data. Be wary with this mode as the engine saves a tremendous amount of data which can saturate the disk quite quickly according to the SIMPLIS folks.

Agenda

FUTURE ELECTRONICS

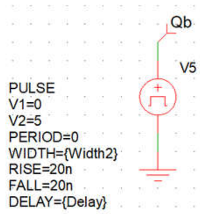
- What is Elements?
- Running a Basic Simulation
- A Buck Converter
- Ac Linear Analysis with SIMPLIS
- Importing a SPICE Model
- The Ready-to-Use Template

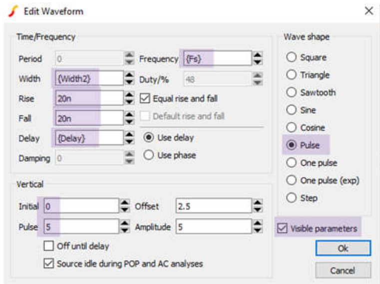
We will now see how we can simulate a buck converter.



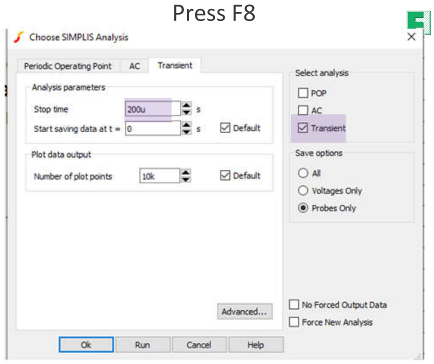
A very convenient feature are control blocks which let you pass parameters directly from the schematic. You access a block – which is not a text block – by Place | From Symbol Library… and Control Block. The principle is to describe values by a label later passed as a parameter. You use the keyword .VAR for a local variable and .GLOBALVAR for a global variable, e.g. to pass parameters to a subcircuit. Here, .VAR Fs=50k passes the value 50k to Fs which can now be passed as a parameter via the curly braces. Enter {Fs} in the Frequency window as well as the other highlighted parameters. Display the variables to see them in the schematic.

Do the same for a second source

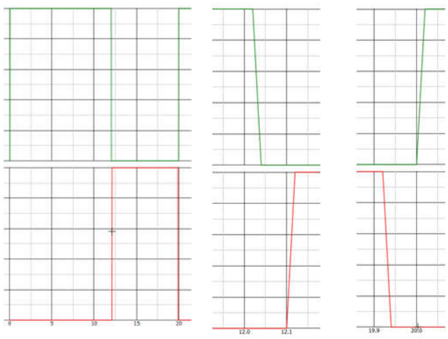




Press F8



Then press F9 to run:



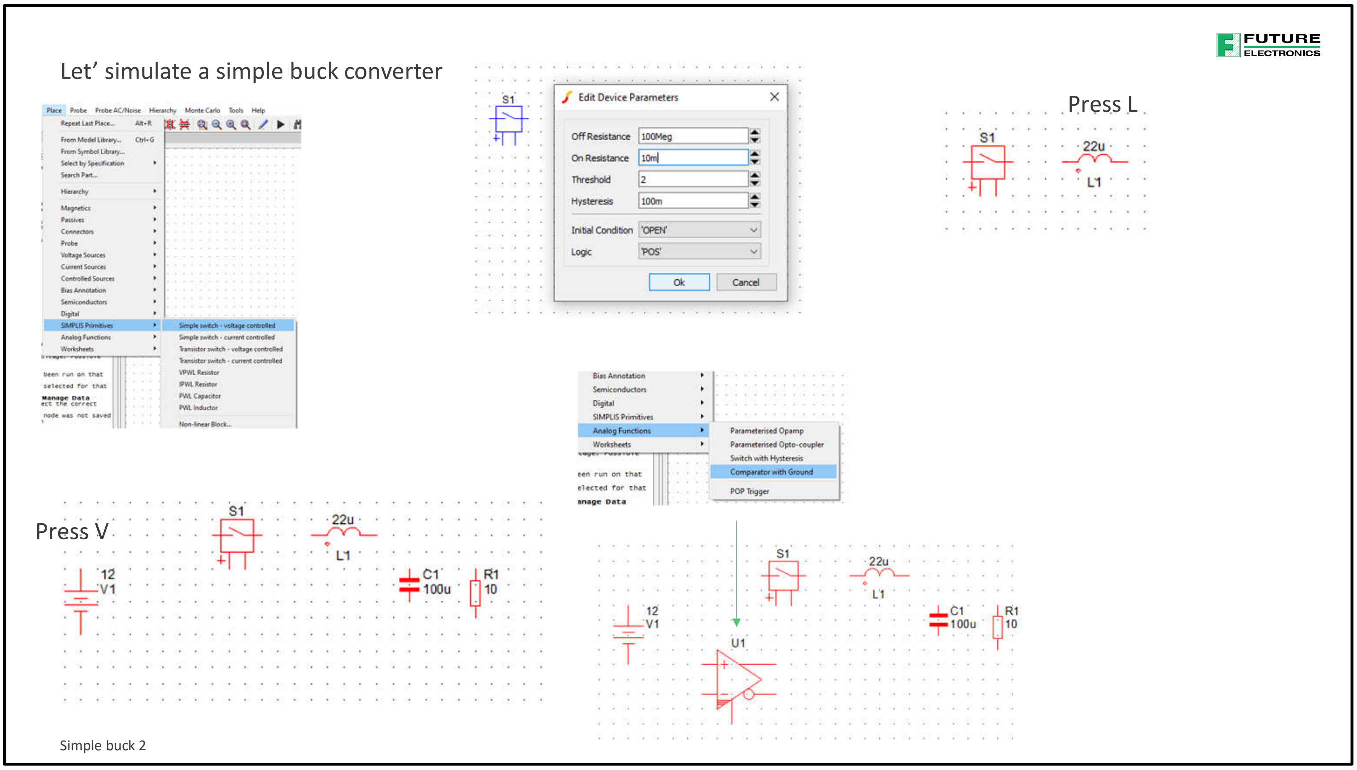
You could also pass parameters directly in the command window: press F11 to open the window, press F11 again to hide it

```
.simulator SIMPLIS
.options
+ PSP_NPT=10001
+ POP_ITRMAX=20
+ POP_OUTPUT_CYCLES=5
+ SNAPSHOT_INTVL=0
+ SNAPSHOT_NPT=11
+ MIN_AVG_TOPOLOGY_DUR=1a
+ AVG_TOPOLOGY_DUR_MEASUREMENT_WINDOW=128
.tran 200u 0

*
.VAR Fs=50k ; select frequency
.VAR DT=100n ; select deadtime
.VAR DC=0.6 ; select duty ratio
*
.VAR Ts={1/Fs}
.VAR Width1={DC*Ts}
.VAR Width2={Ts*(1-DC)-2*DT}
.VAR Delay={Width1+DT}
*

.simulator DEFAULT

```

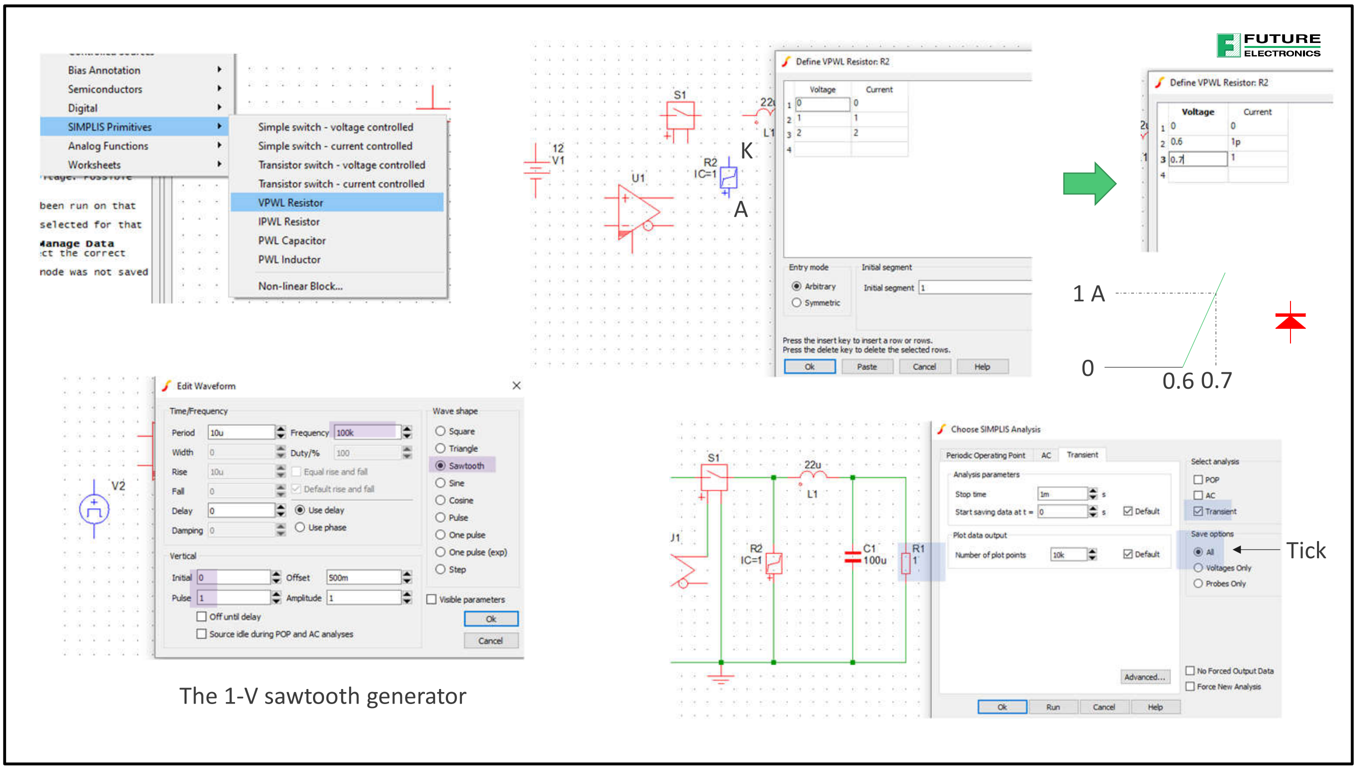
But passing controls from from the schematic is clearer and faster in my opinion.

You've created two complementary waveforms with deadtime, well suited for sync buck of ½-bridge control.

Repeat this operation for the second source and by pressing F8, select a 200-µs simulation length. Run the simulation, zoom in the waveforms with different deadtime values.

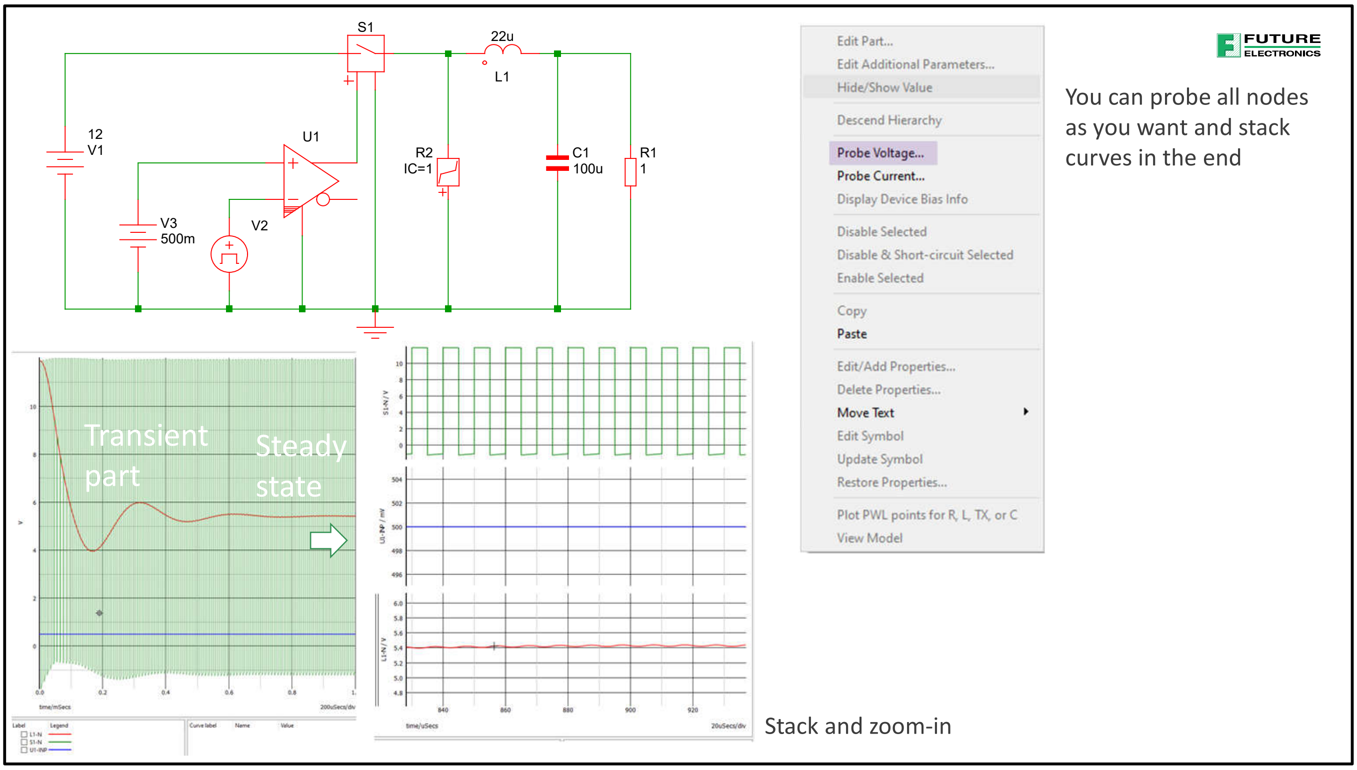


Let’s increase the complexity with a buck converter. Follow the steps in the picture to pick all the discrete components necessary to assemble the dc-dc: a power switch, a voltage source, an inductor etc. All these elements are accessible from the Place submenu or via the hotkeys. Once the switch is placed on the schematic, double-click it and enter the values for the on-off resistances and threshold values.

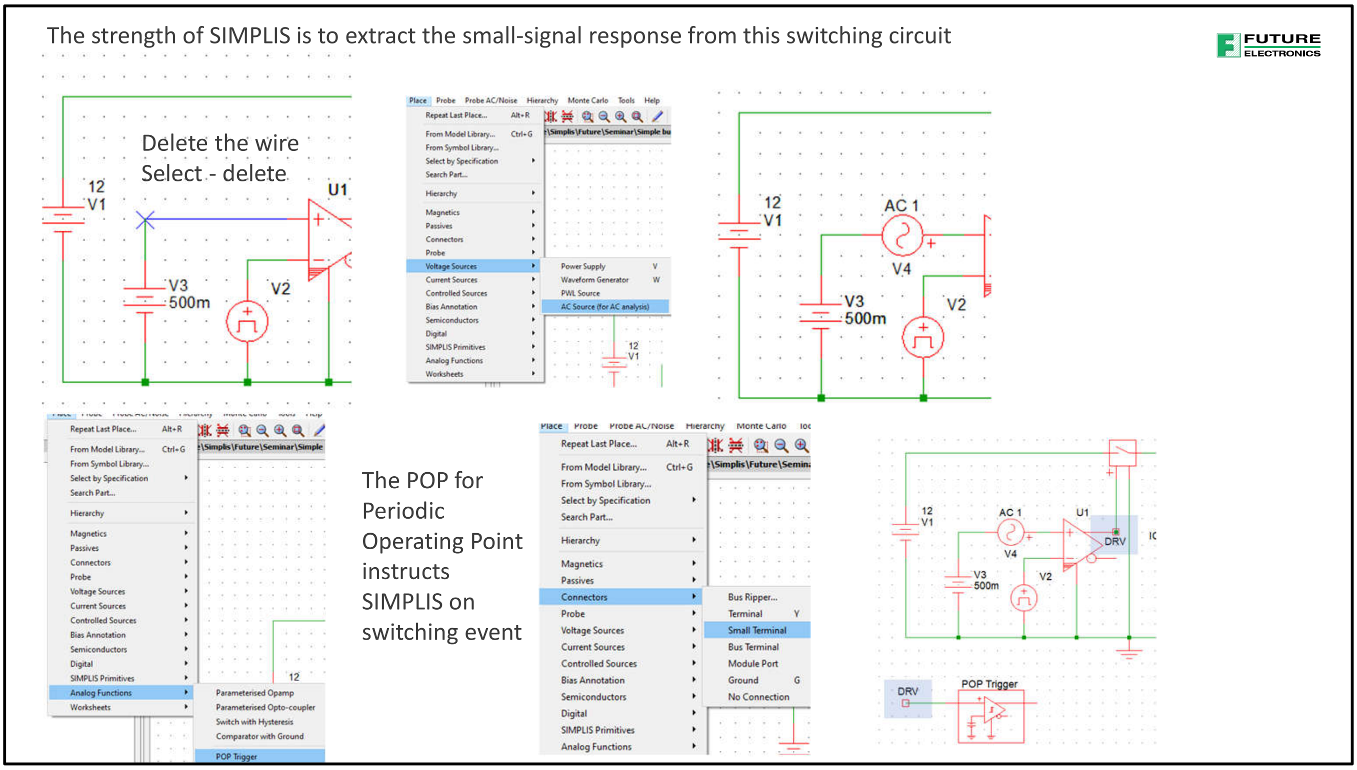


The 1-V sawtooth generator

The diode will use a very popular element in SIMPLIS, a piece-wise linear (PWL) resistor. The idea is to model a resistance depending on the voltage across its terminals: when the voltage is below the threshold – let's say 0.6 V – then the current is very small (1 pA for instance) and when you pass the threshold, the current increases. We set it to 1 A in this example, defining the slope of the diode small-signal resistance. There are two segments, but more can be added of course. The + on the symbol designates the anode. A sawtooth generator is then added with a 1-V amplitude.

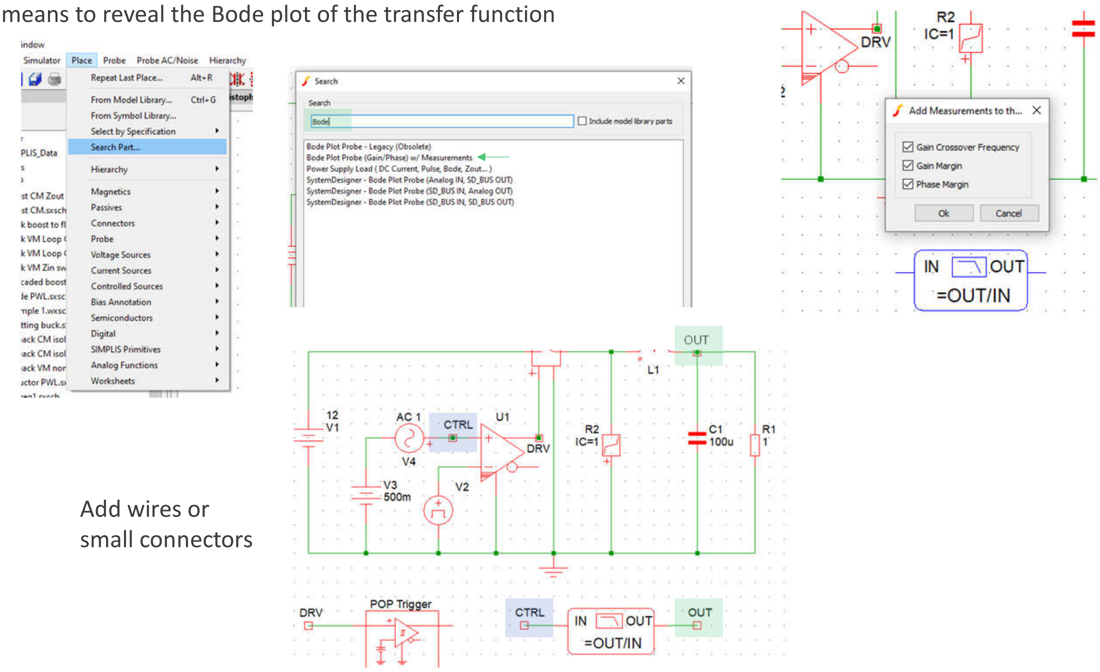


Once the simulation setup is programed for a 1-ms total duration, you can press F9 and run the simulation. As we did not install probes, we have to interactively probe the schematic. Remember, this option is available if you ticked All in the save Save option field. After exploring several waveforms, you can stack the curves to send them to a dedicated axis. Please note the time taken by the transient period before variables stabilize.



Now that we have a switching circuit, we can extract its small-signal response showing the strength of SIMPLIS in this type of application. We need an ac source that we obtain from the Place menu. The value of this source is 1 but it is more a flag for the engine than a value: SIMPLIS permanently internally adjusts the modulation amplitude to keep the circuit in small-signal excitation at all frequencies and you don't need to care about it. Ac analysis in the lab with a frequency response analyzer (FRA) can only be carried if the converter has reached steady-state. SIMPLIS determines steady-state by checking the difference between the value of a current or a voltage at the beginning of a cycle and at the end. When the difference lies in the femto range, the simulator displays the steady-state waveforms and runs the ac sweep. To instruct the simulation engine about the switching events, a trigger input is needed and that is the POP Trigger circuit.

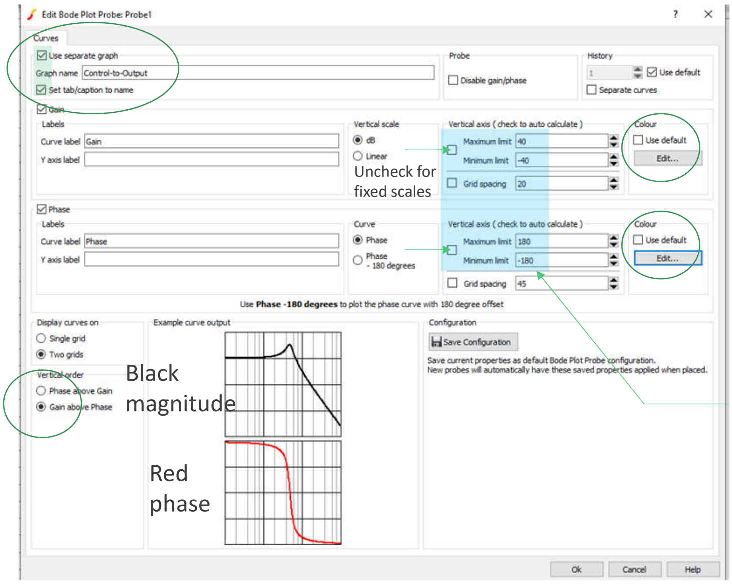
We now have a means to excite the converter, we need a means to reveal the Bode plot of the transfer function

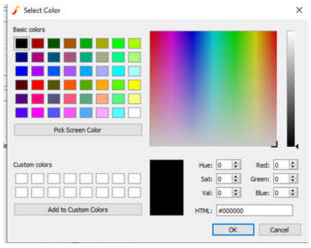


Add wires or small connectors

To plot the magnitude-phase response, we can either use specific probes or place a Bode box obtained from the Place | Search Part. Connect the IN input to the stimulus, here the (+) pin of the comparator and link the OUT pin to the response node, the converter output voltage. Then, double-click on the box.

Enter a name and tick the two boxes



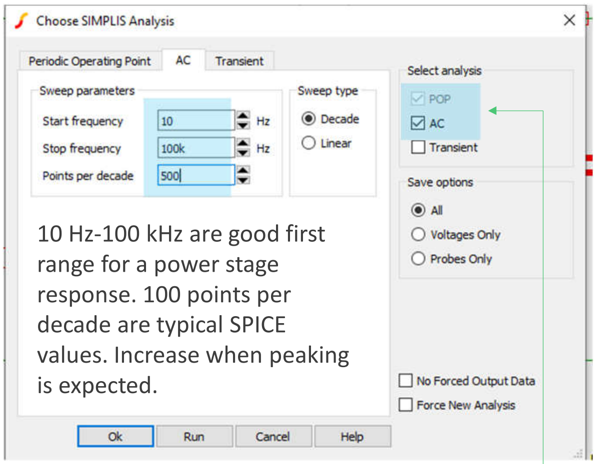


I like a black magnitude and a red phase curve :)

It is more comfortable to read when both magnitude and phase plots are symetrical with respect to the 0 point: 0 dB and 0°

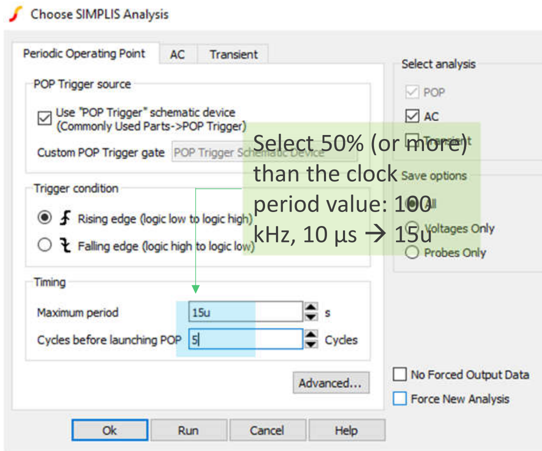
Now you can tailor the way the frequency response will be displayed after the simulation. Add a label which will appear as a tab name in the waveform viewer. Then affect black and red colors respectfully to magnitude and phase responses if you wish and have magnitude graph above phase graph.

Press F8 to access the simulation parameters



10 Hz-100 kHz are good first range for a power stage response. 100 points per decade are typical SPICE values. Increase when peaking is expected.

Tick ac analysis, POP will be checked automatically

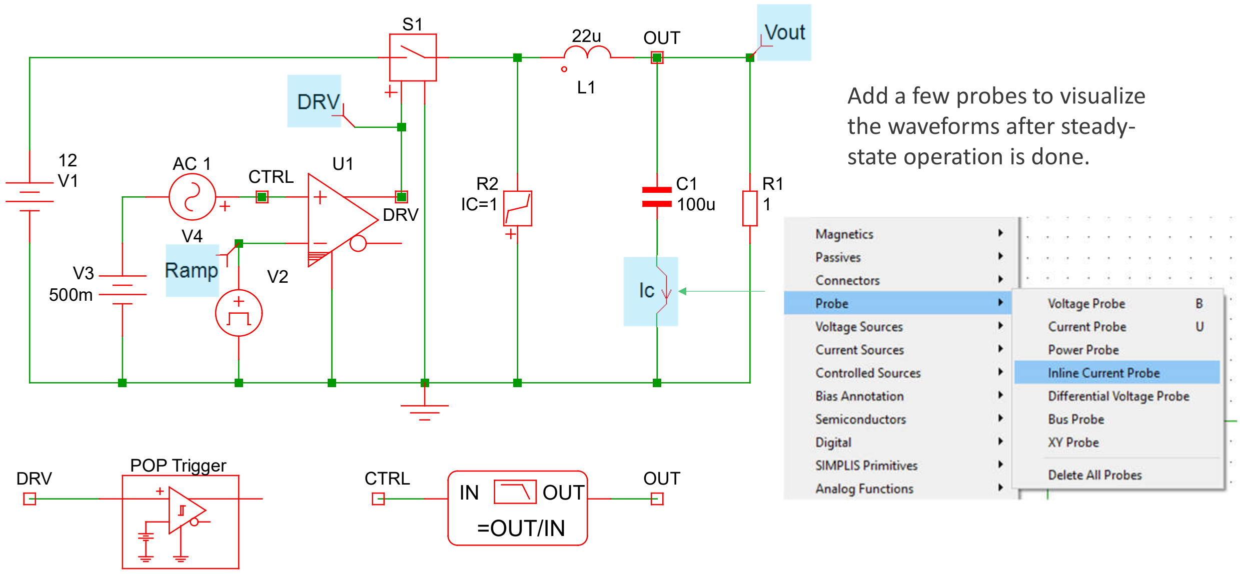


Select 50% (or more) than the clock period value: 100 kHz, 10 µs → 15u

The periodic operating point or POP is a SIMPLIS proprietary algorithm which determines the steady-state operating point (average currents in capacitor and average voltages across inductors are all zero) in a record time. When POP is finished, the program launches the ac analysis.

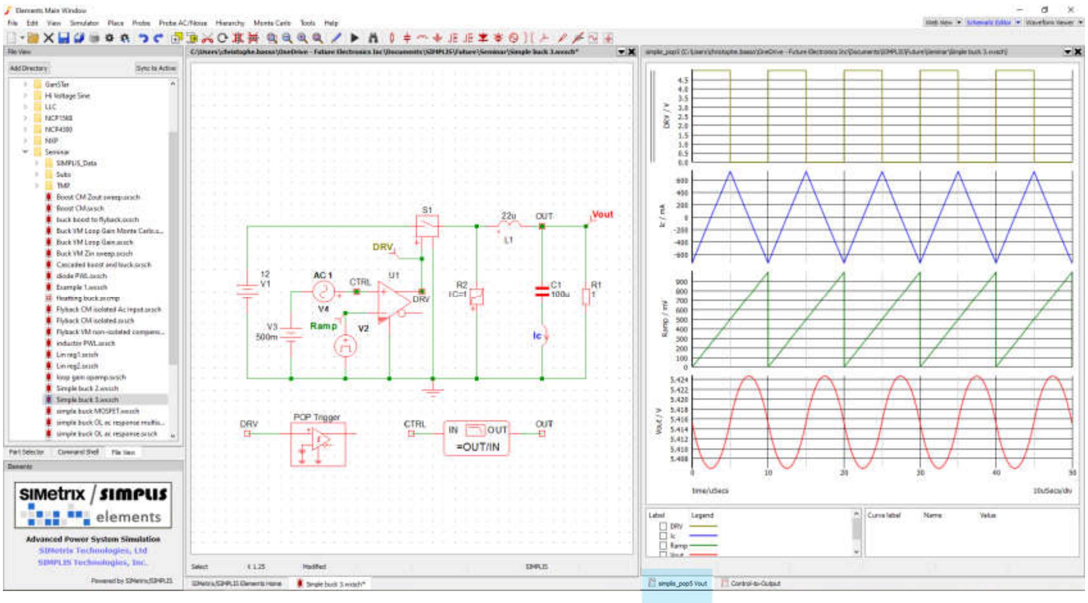
Now that the circuit is ready, we can press F8 and select the AC tab. We will sweep from 10 to 100 kHz and choose 500 points. Increasing the number of points is typical for revealing a peaky response. Then tick ac analysis which will automatically check the POP algorithm. The POP or Periodic Operating Point is a way to quickly bring the converter to steady state and skip the transient period. The maximum period is set to typically 1.5 x the switching period. In this 100-kHz converter (T_s = 10 µs), the maximum period is set to 15 µs. The number of cycles before launching POP will depend on the circuit and can the value can be adjusted to higher numbers if an error message pops up.

Add a few probes to visualize the waveforms after steady-state operation is done.



Ready to run the simulation! Press F9

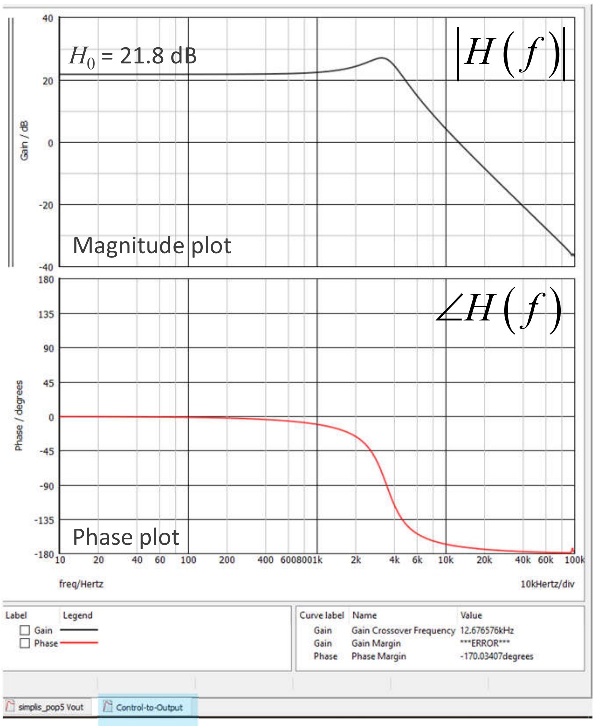
I have added fixed probes to visualize the periodic point once simulation is done. The current in the capacitor is monitored via an in-line current probe. Displaying the transient operating point is similar to checking the dc operating point when using an average model and verify the converter is stabilized to the right output voltage or current. We can now press F9 to start the simulation engine.



FUTURE ELECTRONICS

This is the steady-state operating point: the converter is stabilized and the start-up sequence is long gone. The waveforms are the probes you placed

The simulation time is flashing and confirms a dc output voltage of 5.4 V roughly. It is sinusoidal because there is no resistance in series with the output capacitor. By changing the dc bias from V3, the duty ratio (currently 50%) will move increasing or decreasing the dc output voltage.



Quasi-static or dc gain value:

$$H(s) = H_0 \frac{N(s)}{1 + \frac{s}{\omega_0 Q} + \left(\frac{s}{\omega_0}\right)^2}$$

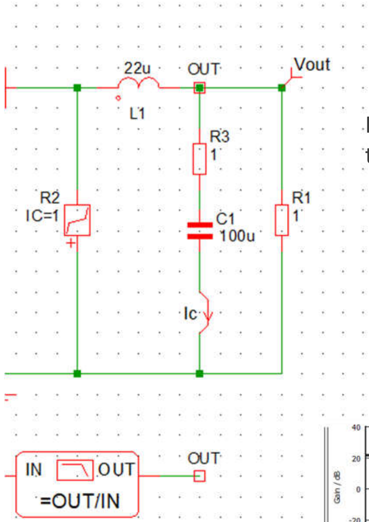
Input voltage, 12 V

$$H_0 = \frac{V_{in}}{V_p} \frac{R_L}{R_L + r_L} \approx \frac{V_{in}}{V_p} \quad \Rightarrow \quad H_0 = 20\log\frac{12\text{ V}}{1\text{ V}} \approx 21.6\text{ dB}$$

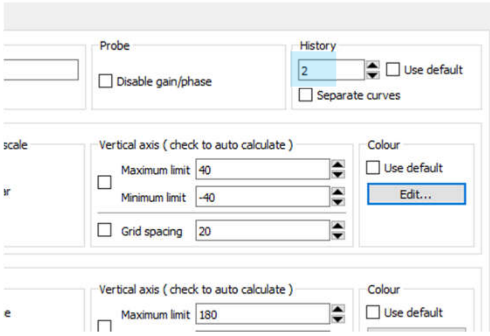
Peak amplitude of the sawtooth

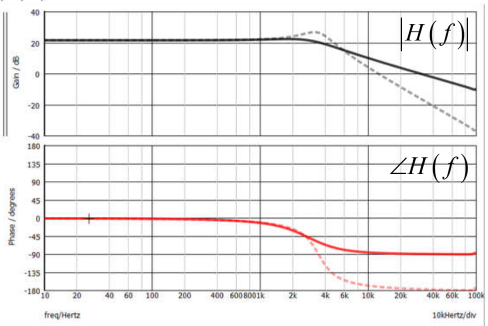
The Bode plot is quickly displayed and shows the expected magnitude and phase graphs. The dc gain H_0 depends on the input voltage and the sawtooth peak amplitude. The various ohmic losses (omitting the MOSFET $r_{DS(ON)}$) also affects the static gain but theoretical and measured gain values are very close in this example.

Select a remanence of 2



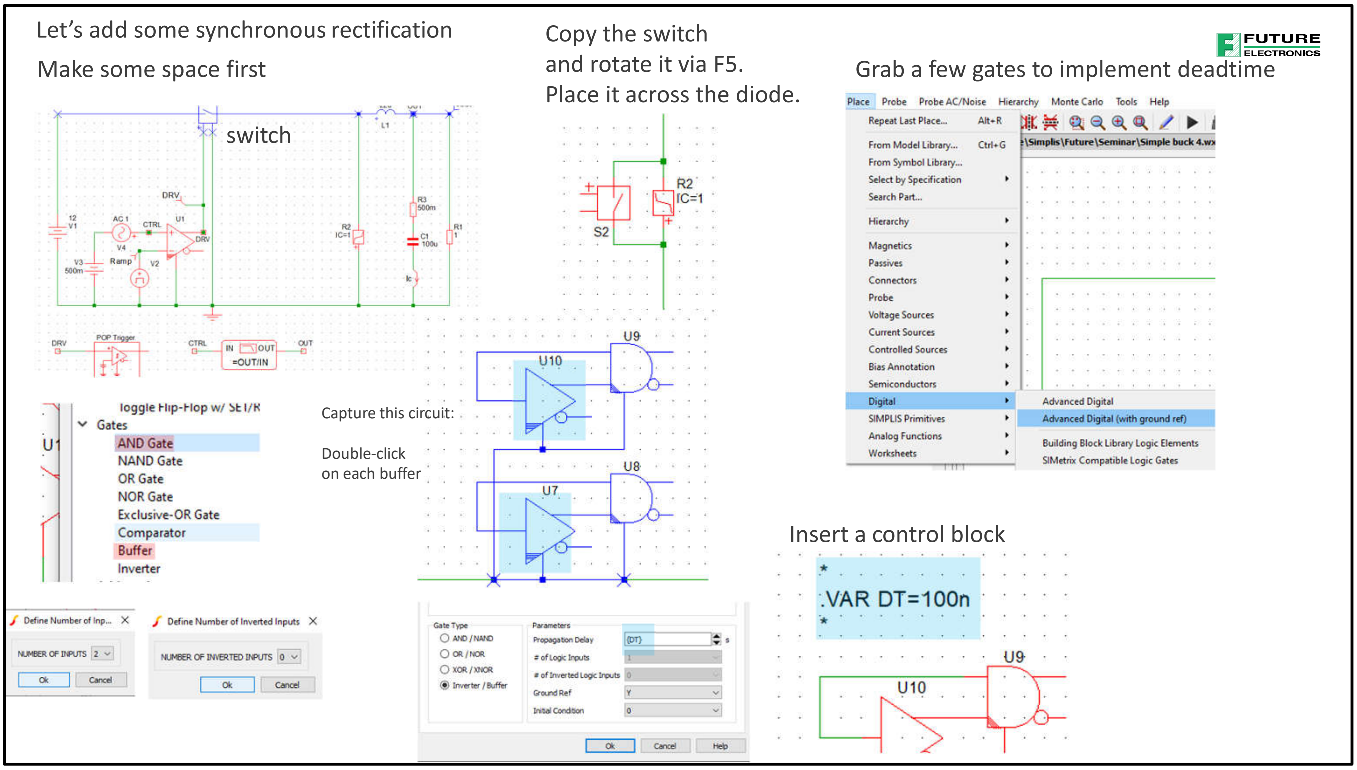
Insert a resistor in the schematic



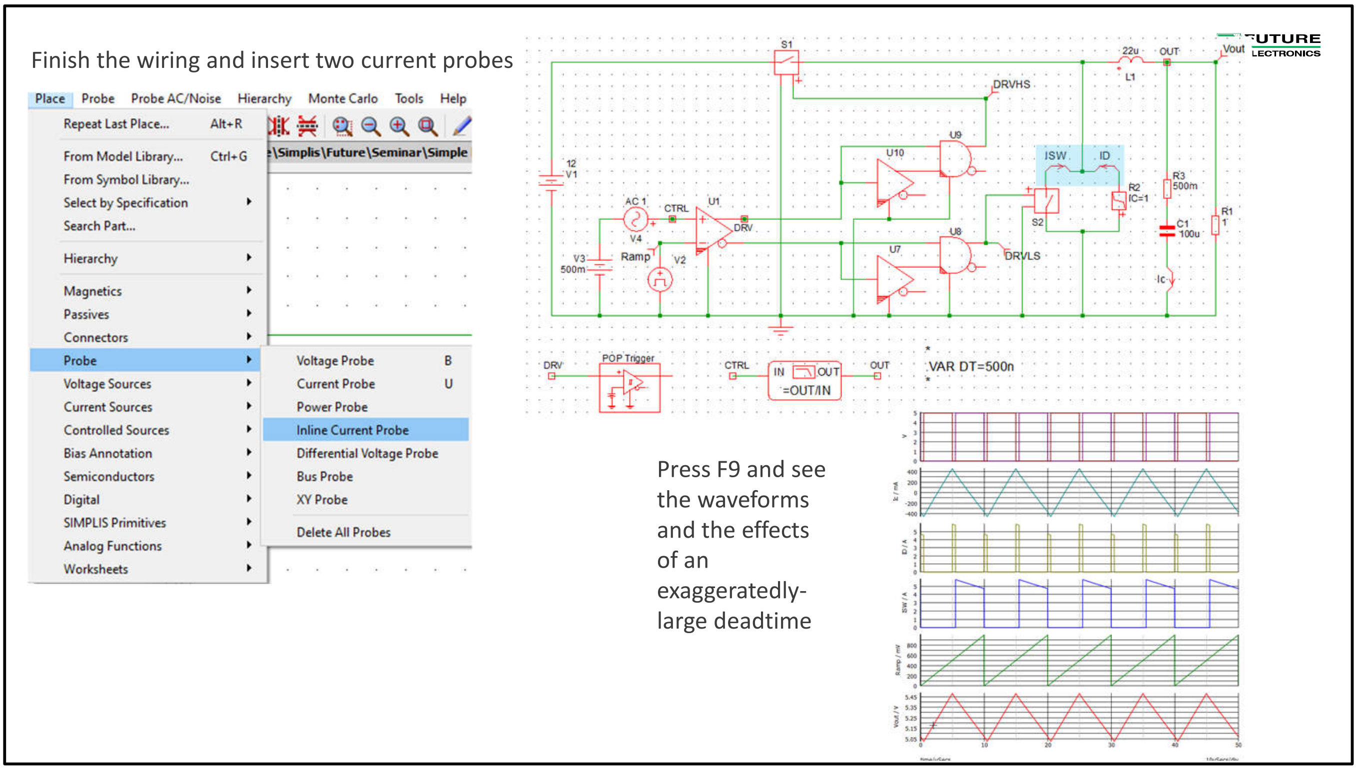


Change the series resistance value and see the different responses. The dashed lines are the first results.

We can now add an equivalent series resistance (ESR) to the capacitor and see the effects on the Bode plot when varying its value. Make sure the history depth is increased by double-clicking on the Bode box. As expected, increasing the ESR will damp the system and lower the quality factor.



Now that the buck is running, we can add synchronous rectification to it. The upper-side switch is copied and pasted across the existing diode with similar characteristics. We now create a simple deadtime generator featuring an inverter and an AND gate. The inverter is programmed to delay the signal by the amount DT passed as a parameter.



When the extra switch and the dead-time generator are assembled, add two inline current probes to monitor the currents in the synchronous switch and the freewheel diode. Press F9 to check steady-state waveforms and the effect of the deadtime.

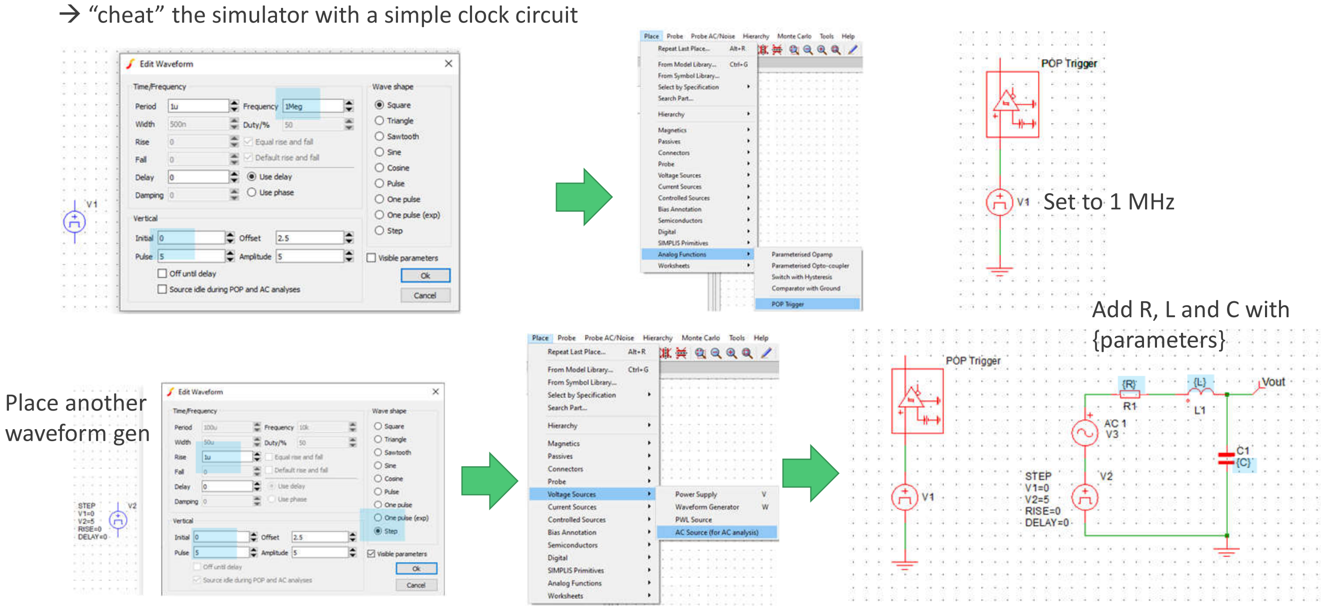
Agenda

- What is Elements?
- Running a Basic Simulation
- A Buck Converter
- Ac Linear Analysis with SIMPLIS
- Importing a SPICE Model
- The Ready-to-Use Template

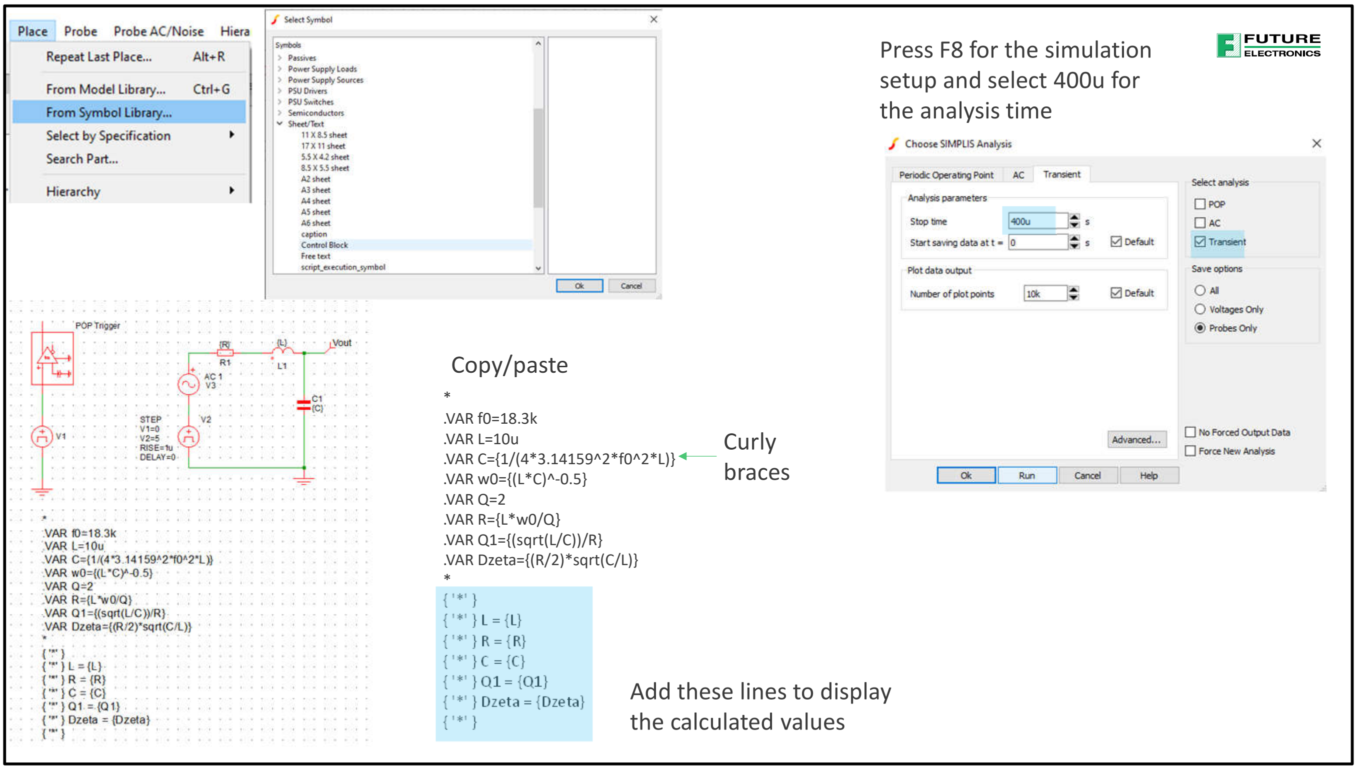
Let’s now see how to smoothly start using Elements

Ac analysis with SIMPLIS

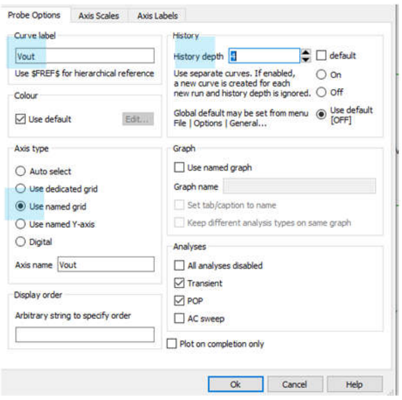
SIMPLIS is a time-domain simulator therefore switching is needed to perform an ac analysis
→ "cheat" the simulator with a simple clock circuit



SIMPLIS is a time-domain simulator which needs a switching node to sync its algorithm. In a pure linear circuit such as a RLC filter, there is no switching node. We can create one with a waveform generator delivering a clock solely for synchronization purposes. In this example, it is set to 1 MHz. Do not forget to add parameters to the passive RLC elements as suggested in the illustration.



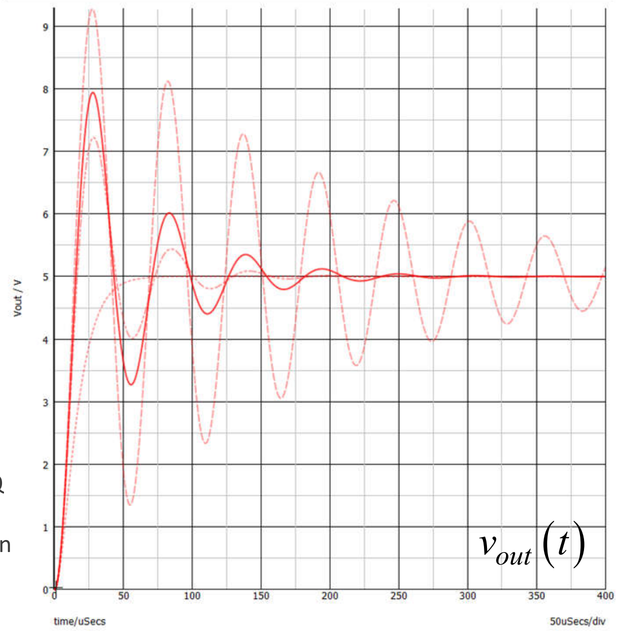
You can place a control block on the schematic and determine the values for R and C considering a quality factor *Q* you may choose. Because all the variables are local, a .VAR statement will be used to define the parameters values. It is recommended to add curly braces when using mathematical formulas in the control block.



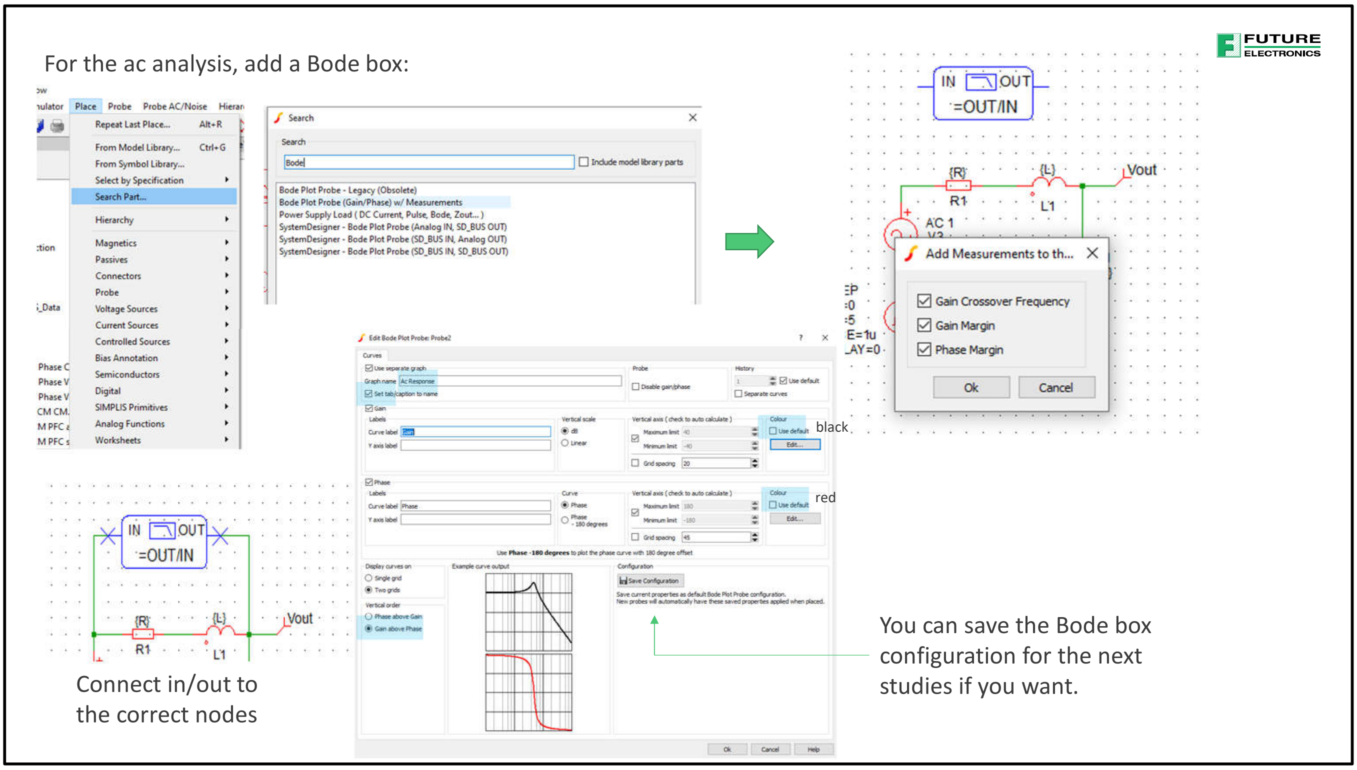
Add the V_{out} probe and select a persistence of 4

```
*
.VAR f0=18.3k
.VAR L=10u
.VAR C={1/(4*3.14159^2*f0^2*L)}
.VAR w0={(L*C)^-0.5}
.VAR Q=2
.VAR R={L*w0/Q}
.VAR Q1={(sqrt(L/C))/R}
.VAR Dzeta={(R/2)*sqrt(C/L)}
```

Change the value of Q in the control block and press F9. The plain curve is the last run.

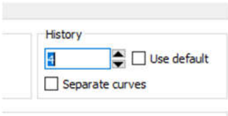


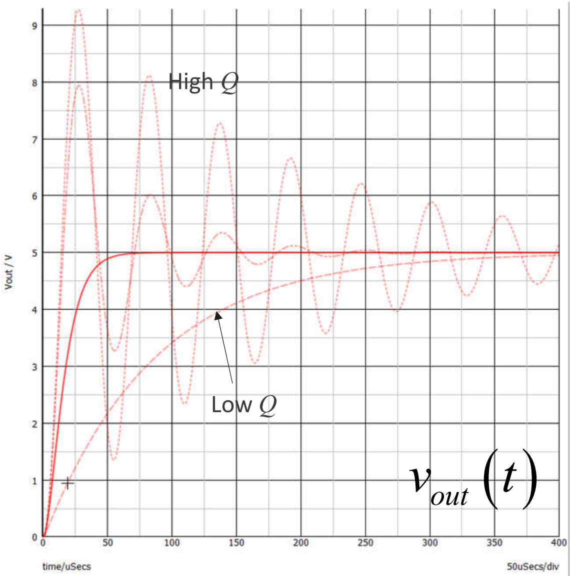
A probe is added to sense the output voltage and the persistence is set to 4. Now change the value of *Q* and press F9 after each change. The plain curve is the most recent result.

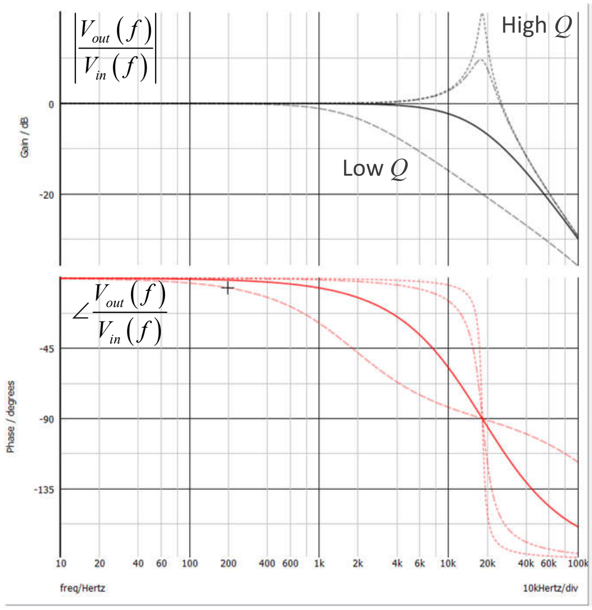


As we are interested by the ac analysis, we need to add a Bode box to plot the transfer function linking the stimulus input to the response. We will also change the quality factor on the fly and check the impact on the response.

If you set history or persistence to 4 and change Q, you should see







By changing the value of Q, you see different ringing responses: a low Q (or a high damping ratio) means losses in the circuit and heavy damping. The second-order system can be approximated by two cascaded poles with one dominating at low frequency and the second one at higher frequency. When you increase the Q, the series resistance gets smaller and the circuit has less losses hence more oscillations as the poles approach the imaginary axis.

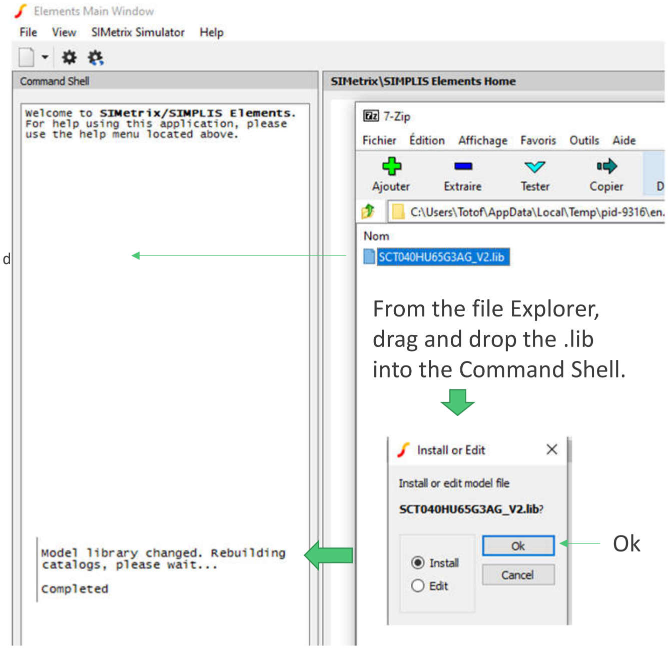
Agenda

- What is Elements?
- Running a Basic Simulation
- A Buck Converter
- Ac Linear Analysis with SIMPLIS
- Importing a SPICE Model
- The Ready-to-Use Template

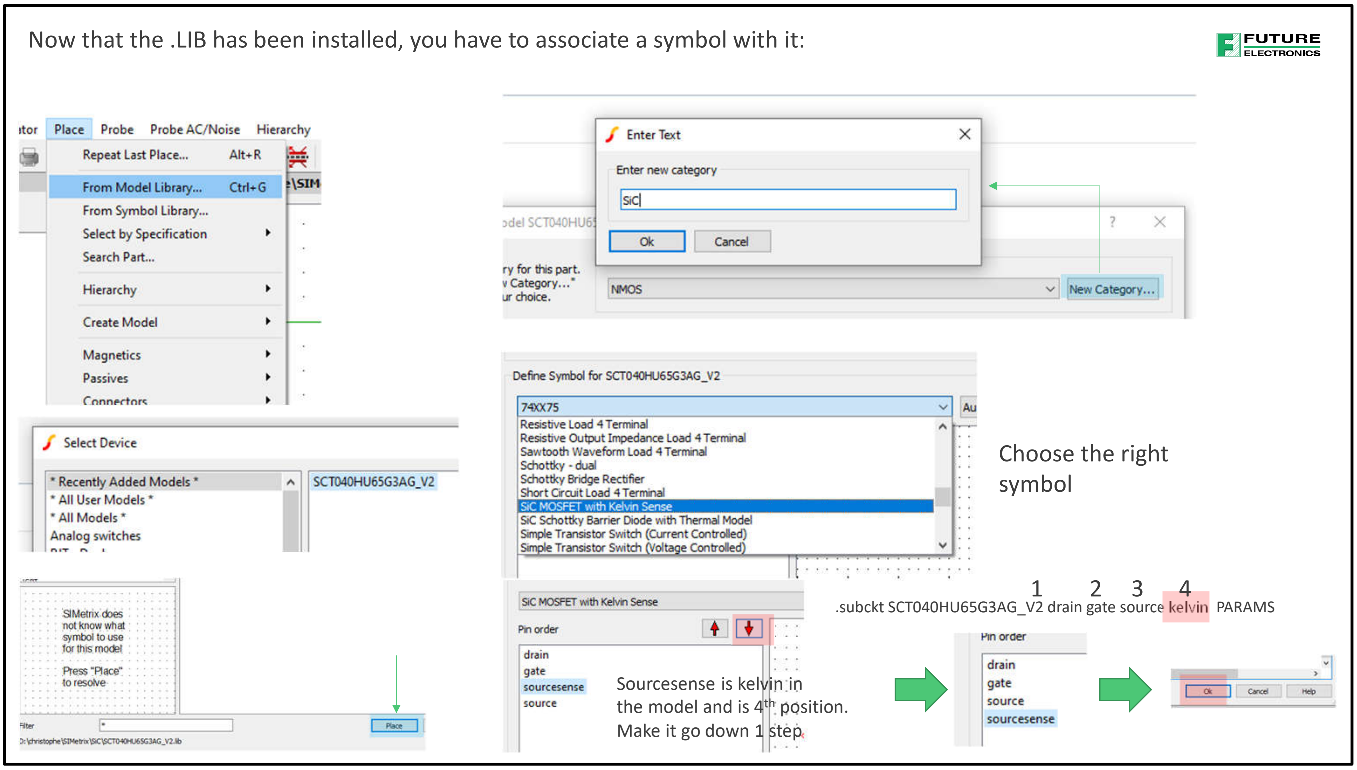
We are now going to look at how we can incorporate a SPICE model from the web.

Assume you have downloaded a SPICE model coming from a manufacturer, ST in this example:

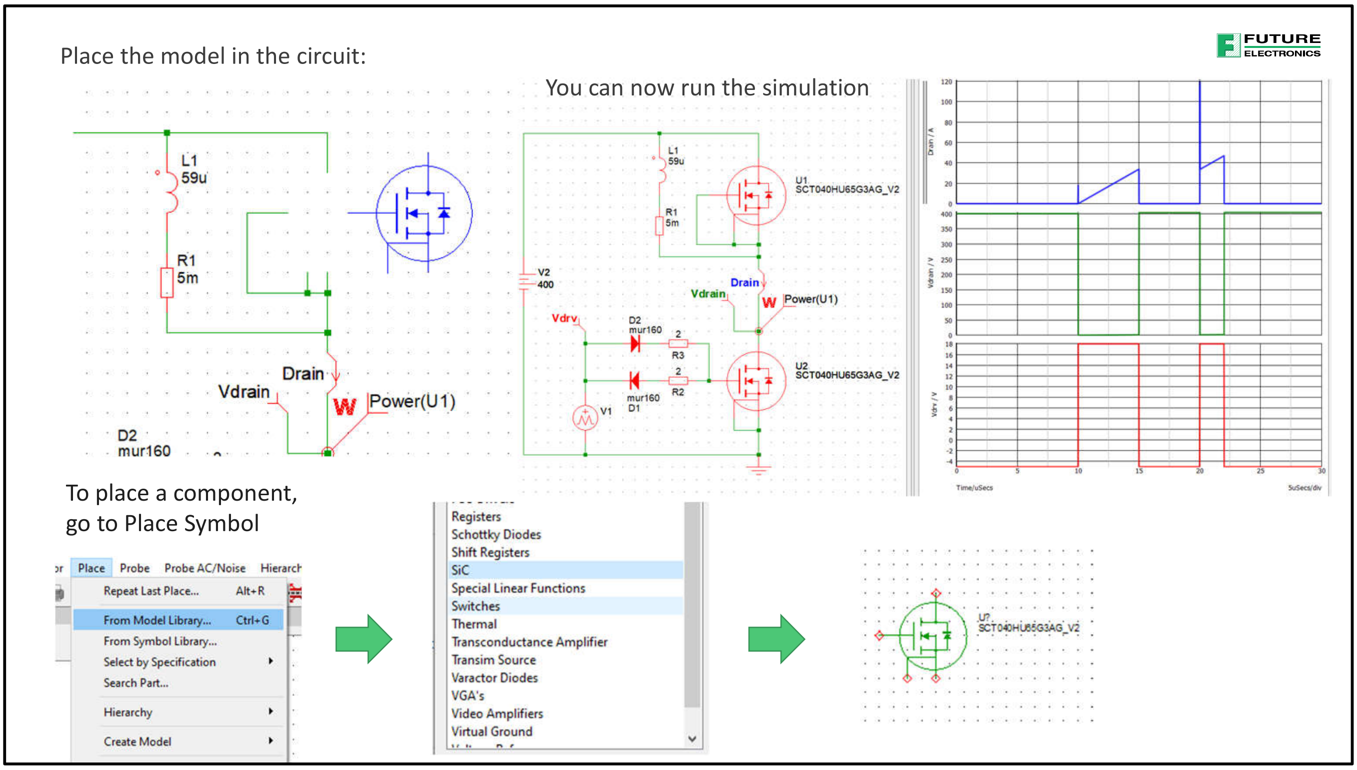
```
*******************************************************************
****** STMicroelectronics MOSFET, IGBT and Bipolar Library ******
*******************************************************************
*                                                                 *
* STMicroelectronics therefore does not assume any               *
* responsibility arising from their use.                          *
* STMicroelectronics reserves the right to change models          *
* without prior notice.                                            *
*                                                                 *
* v.19.3.0 - FEB 2021                                             *
*******************************************************************
.subckt SCT040HU65G3AG_V2 drain gate source kelvin  PARAMS:  dR=0 dVth=0 dVsd=0 dCi=0 d

E1   Tj    val_T    VALUE={TEMP}
R1  val_T  0    1m
Rkelvin kelvin s2 5m
Ckelvin kelvin s2 1p
VLd   drain  d3x  0
*RLd  drain  d3x   1
VR_dr  d3  d2   0
Rdrain-fissa d3x d3 3m
VLg gate    g2    0
*RLg gate    g2     1
VLs source  s2     0
*RLs source  s2     1

Rg   g2 g1  1.0
......
```



Assume you want to simulate with a model whose lib file is supplied by the selected manufacturer. Select the .LIB file from the file Explorer and drop it into the SIMetrix/SIMPLIS command shell. The program recognizes the syntax and installs the model. Now you need to associate a symbol with it.

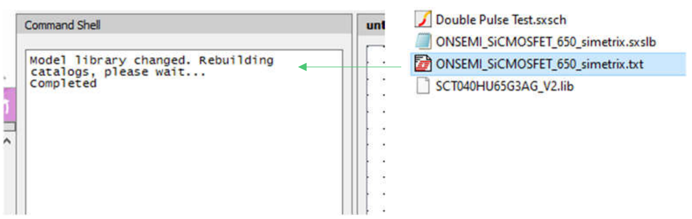


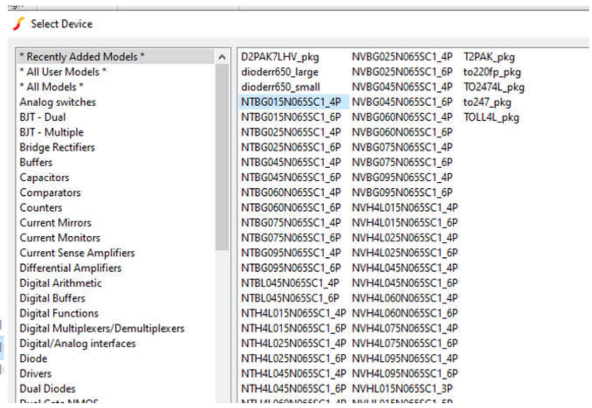
Once the LIB file has been installed in SIMetrix, you have to associate a symbol with it so that you can easily place it in the schematic. Check if the category exists or create one like here. Then associate a symbol to the model and check the pin order. It is important to carefully check this point otherwise simulation may not work of course if pins are swapped. Here, the kelvin connection is the last one and this is sourcesense which needs to be brought down by one step.



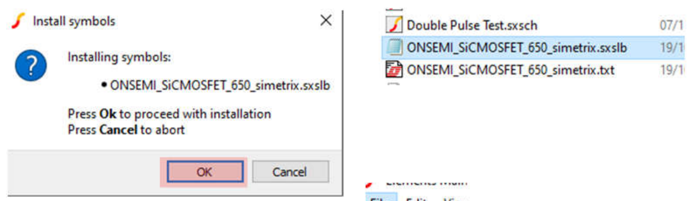
The symbol is now appearing in the schematic capture and can appropriately be placed in the electrical schematic. You can also call it by pressing control G and selecting the part in the SiC category you have created.

Very often, manufacturers provide models and their associated symbols in case of a complete library. The operation is very similar (you can also see [AND9783D](AND9783D)). First drop the .LIB containing all the subcircuits in the Command Shell Window:



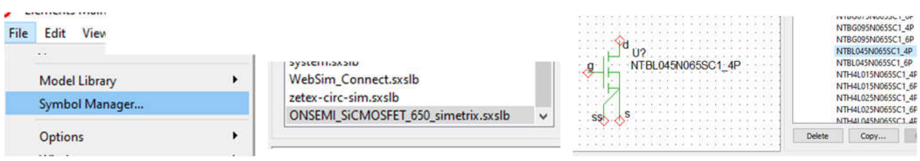


Now, you have to load the associated symbol file:



Then CTRL+ G to access the models

You can see the models are properly imported:

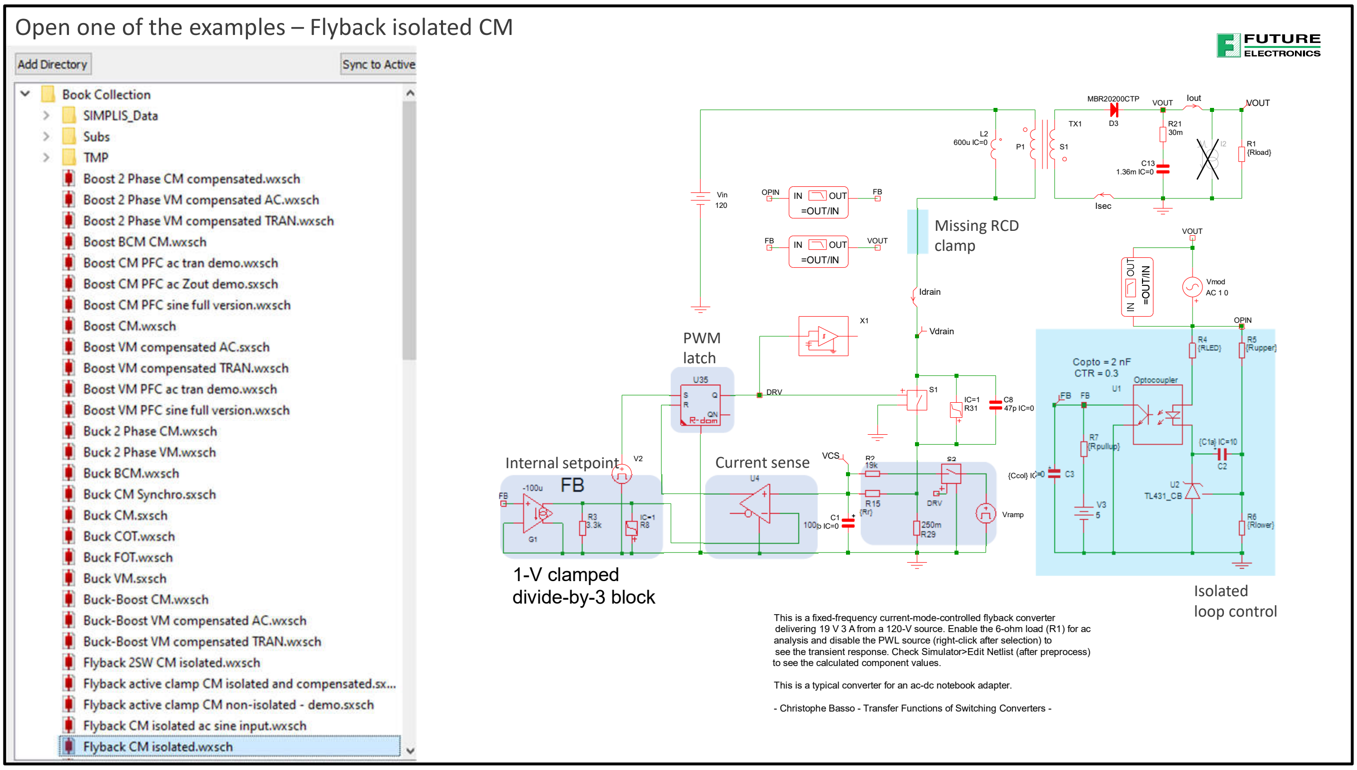


Some manufacturers also provide set of models in the form of a file containing multiple subcircuits. This is the case with the onsemi set of SiC models for instance. In this is case, you can import the .LIB models and the associated files by dropping each file separately in the command shell window. The AND9783D application note gives more details on how to proceed with these models.

Agenda

- What is Elements?
- Running a Basic Simulation
- A Buck Converter
- Ac Linear Analysis with SIMPLIS
- Importing a SPICE Model
- The Ready-to-Use Template

We are now going to look at the ready-made templates.



You can now open one of the ready-made templates I supplied with the last book on transfer functions. Here you see an isolated flyback converter whose loop is closed via the classic couple TL431+optocoupler. A simple current-mode converter is assembled and simulated. It delivers 19 V for a classical ac-dc notebook adapter. The feedback dynamic range at the optocoupler collector is adjusted between ground and 3 V for the upper range. A clamped 1-V limit sets the maximum peak current. Please note the absence of the RCD clamp in this example.

```
*
.VAR Vin=120
.VAR Vout=19
.VAR Lp=600u
.VAR Ri=250m
.VAR N=250m
.VAR Rload=6
.VAR Ts=15u * please update clock and ramp generators *
*
.VAR D=Vout/(Vout+N*Vin) * duty ratio calculation *
.VAR mc=0.818/(1-D) * recommended compensation value for a Q of 1 *
.VAR Sn={(Vin/Lp)*Ri}
.VAR Sramp={2.5/Ts} * 2.5 V over Ts - check your IC specs *
.VAR mc=1.5 * set this value for ramp comp *
.VAR Se={(mc-1)*Sn}
.VAR Rr={(Se/Sramp)*19k+1m}
```

Slope compensation

```
.VAR fRHPZ={((1-D)^2*Rload/(D*Lp*N^2))/(2*pi)}
.VAR fcMAX=0.3*fRHPZ
*
```

RHPZ location for max crossover

```
* Enter values extracted from the plant Bode plot
*
.VAR Gfc=-13 * magnitude at crossover *
.VAR PS=-80 * phase lag at crossover *
*
```

Read the power stage response

```
* Enter Design Goals Information Here *
*
.VAR fc=2k * targeted crossover *
.VAR PM=60 * choose phase margin at crossover *
*
```

Your targeted values (2 kHz, 60°)

```
* Enter the Values for Vout and Bridge Bias Current *
*
.VAR Vout=19
.VAR Ibias=250u
.VAR Vref1=2.5
.VAR Rlower=Vref1/Ibias
.VAR Rupper=(Vout-Vref1)/Ibias
```

```
* Optocoupler specifications *
*
.GLOBALVAR Rpullup=20k * check with the selected control chip *
.GLOBALVAR Fopto=6k
.GLOBALVAR Copto=1/(2*pi*Fopto*Rpullup)
.GLOBALVAR CTR=0.33
*
```

Optocoupler characterization

```
.VAR VL=0.2
.VAR VCEsat=0.3
.VAR Vdd=5
.VAR Vf=1
.VAR A=Vout-Vf-VL
.VAR B=Vdd-VCEsat
.VAR Rmax=(A/B)*Rpullup*CTR
*
* Do not edit the below lines *
*
.VAR boost=PM-PS-90
.VAR fp=(tan(boost*pi/180)+sqrt((tan(boost*pi/180))^2+1))*fc
.VAR fz=fc^2/fp
.VAR G=10^(-Gfc/20)
.VAR RLED=CTR*Rpullup/G
.VAR C1a=1/(2*pi*fz*Rupper)
.VAR C2a=1/(2*pi*fp*Rpullup)
.VAR Ccol=C2a-Copto
*
*
```

Pole-zero calculation

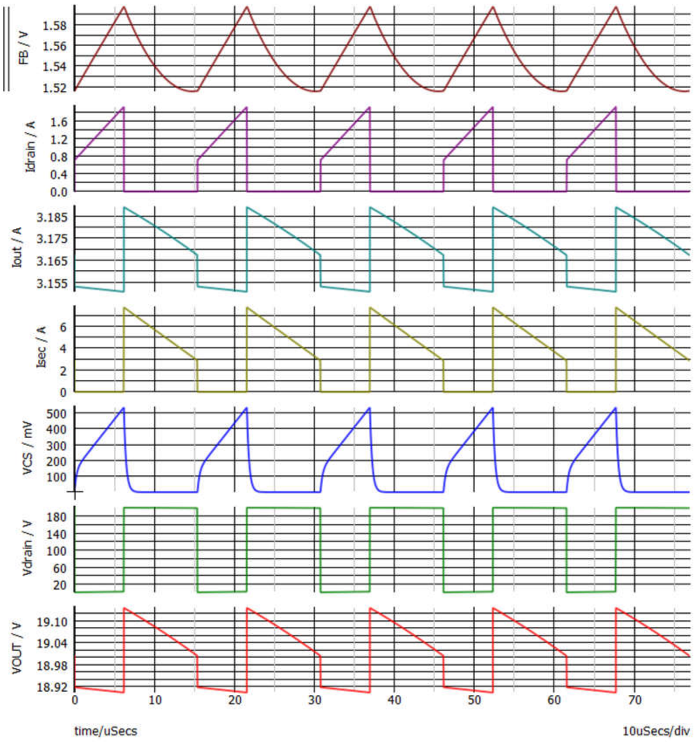
```
{'*'}
{'*'} Rupper = {Rupper}
{'*'} Rlower = {Rlower}
{'*'} C2 = {C2a}
{'*'} C1 = {C1a}
{'*'} Boost = {boost}
{'*'} Fz = {Fz}
{'*'} Fp = {Fp}
{'*'} Sn = {Sn}
{'*'} Se = {Se}
{'*'} D = {D}
{'*'} Mc = {mc}
{'*'} Rramp = {Rr}
{'*'} Rmax = {Rmax}
{'*'} FRHPZ = {fRHPZ}
{'*'} FcMAX = {fcMAX}
{'*'}
```

Print components values

Most of the ready-made templates come with a script compensating the loop

Using control blocks, it is possible to automate the calculation of the compensation path. Once the power stage Bode plot is displayed, you read the magnitude and phase graphs at the selected crossover frequency and feed the macro with these values. The compensation elements are automatically calculated and passed to the schematic.

Press F9 to run the simulation and make sure the operating point is correct: V_{out} = 19 V/3 A √



Feedback voltage within the 3-V dynamic

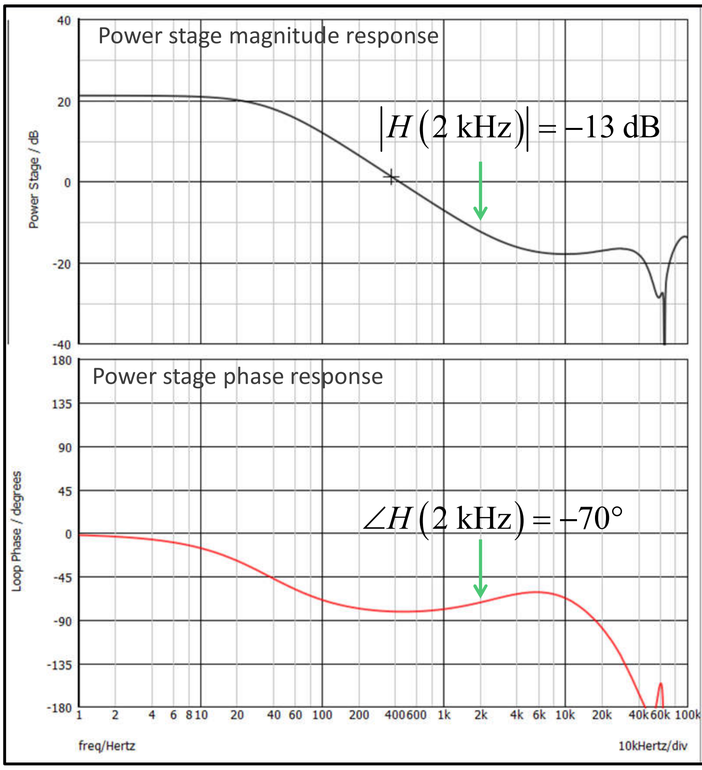
Current in the main switch – check rms current

Dc output current

Secondary-side current – check peaks and rms content

Add leakage inductance and RCD clamp (full version)

The simulation reveals the operating point and confirms an output value at 19 V with a feedback level within the accepted dynamic range (maximum FB is 3 V on the collector and 1 V for the maximum peak current voltage setpoint). From this steady-state waveforms, you can measure the rms, average or any values by pressing F3 and selecting the curve.



Now check the control-to-output transfer function of the power stage. Check the RHPZ value and limit the crossover frequency to 30-20% of this zero position:

```
*
* Rupper = 66000
* Rlower = 10000
* C2 = 1.4481915480445З3e-09
* C1 = 3.31268645711794e-09
* Boost = 50
* Fz = 727.940468532405
* Fp = 5494.95483890924
* Sn = 50000
* Se = 25000
* D = 0.387755102040816
* Mc = 1.5
* Rramp = 2850.001
* Rmax = 24995.7446808511
* FRHPZ = 24616.8762677045
* FcMAX = 7385.06288031136
*
```

$f_{c,\max} \approx 7\ \mathrm{kHz}$

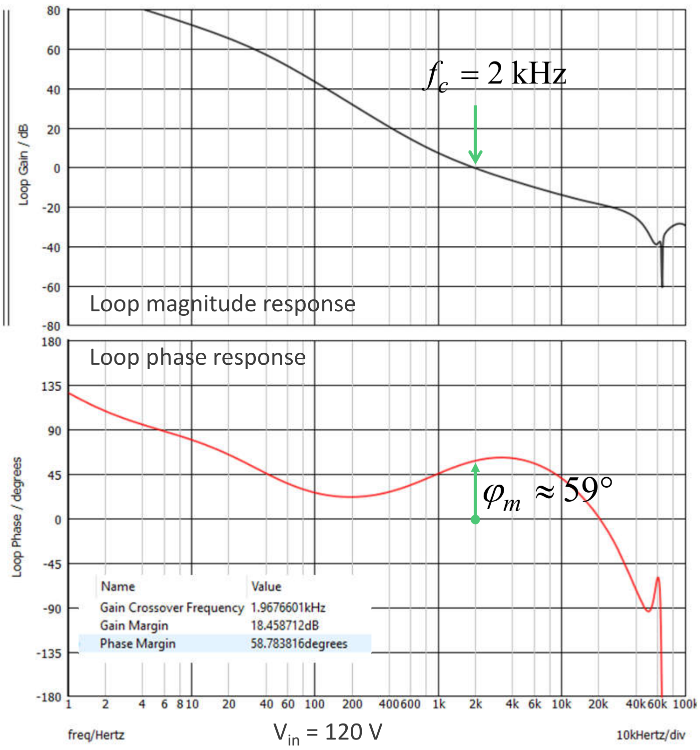
Consider the 6-kHz opto pole

Choose a 2-kHz crossover
Read the magnitude plot a 2 kHz ≈ -13 dB
Read the phase plot at 5 kHz ≈ -70°

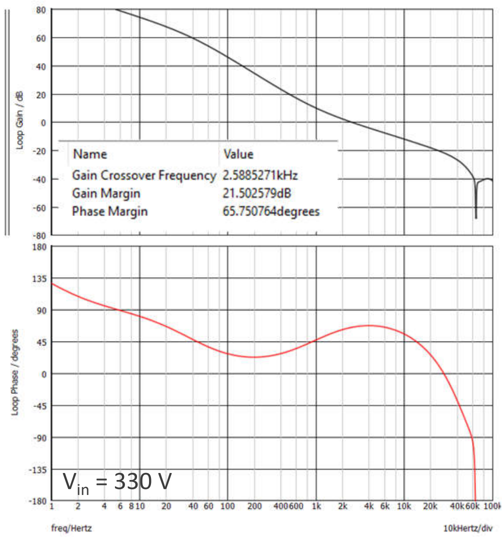
```
* Enter values extracted from the plant Bode plot
*
.VAR Gfc=-13 * magnitude at crossover *
.VAR PS=-70 * phase lag at crossover *
*
* Enter Design Goals Information Here *
*
.VAR fc=2k * targetted crossover *
.VAR PM=60 * choose phase margin at crossover *
*
```

Fill out the requirements

When the power stage control-to-output transfer function is obtained, read the magnitude and phase graphs. In this example, we want a 2-kHz crossover frequency, so we read the graph at 2 kHz and see a gain attenuation of 13 dB with a phase lag of 70°. Enter these values in the control block and see the resulting calculated values in the netlist.

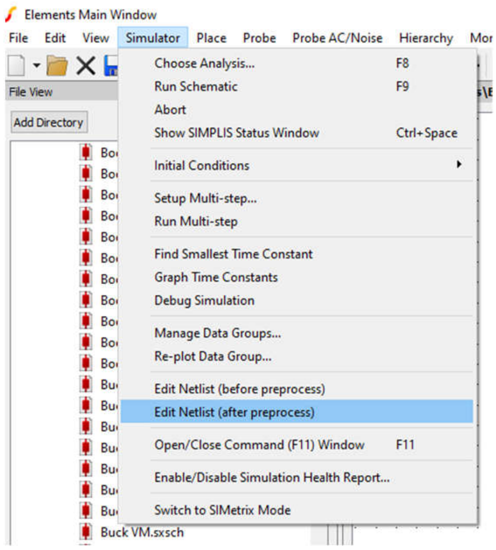


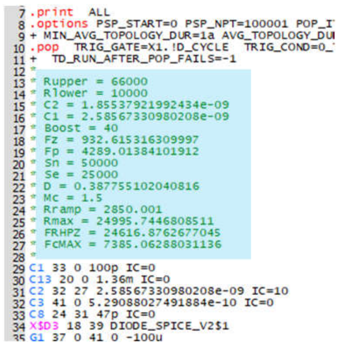
The crossover frequency and phase margin are on target for this low-line operation. You can now increase the input voltage and check how stability evolves at this voltage. Same with the load, make it change and assess the stability at different points.



Once all values are processed by SIMPLIS, the loop gain is displayed with the important information like all the margins at low and high line.

Once the simulation is run, you can unveil the values computed by SIMPLIS:

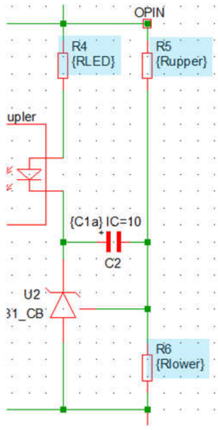




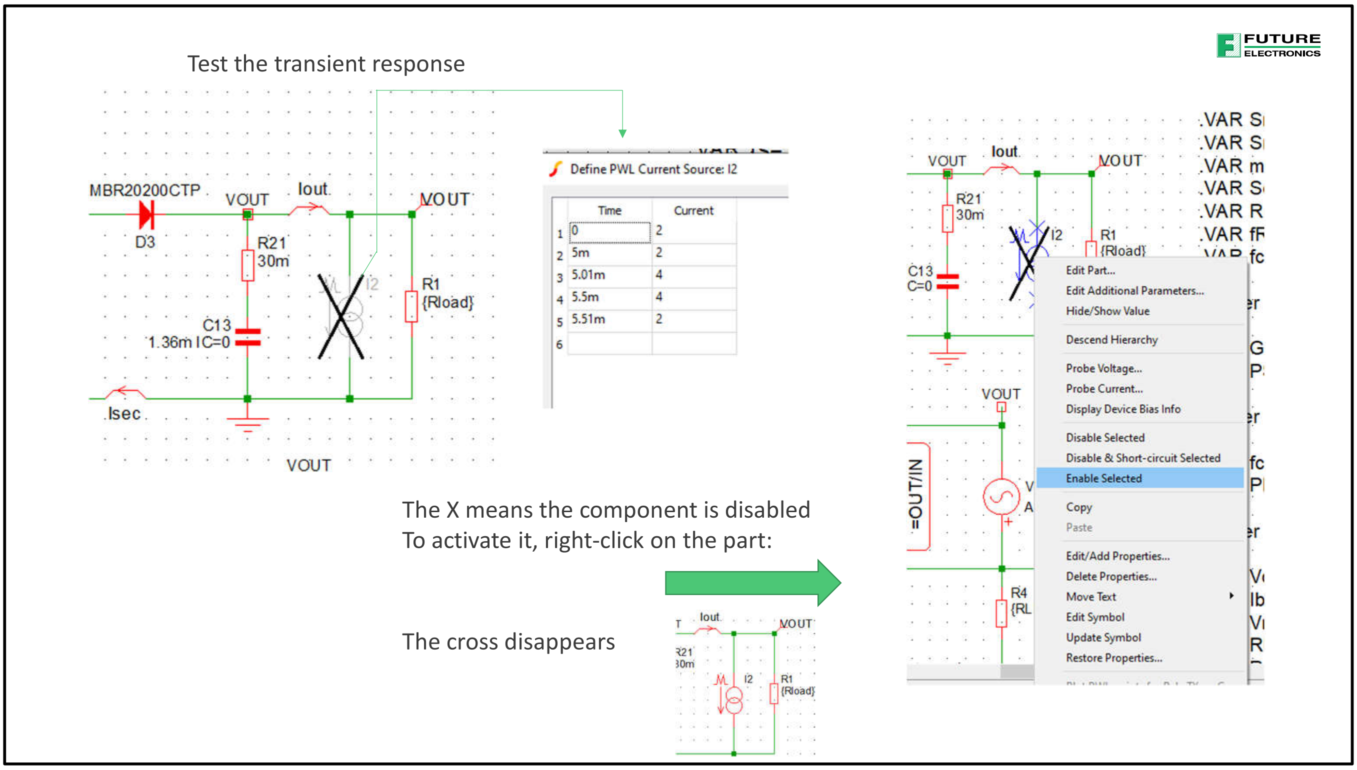
You can add more computed values

{ '*' } Rupper = {Rupper}
{ '*' } Rlower = {Rlower}
{ '*' } C2 = {C2a}
{ '*' } C1 = {C1a}
{ '*' } Boost = {boost}

Value passed to the component

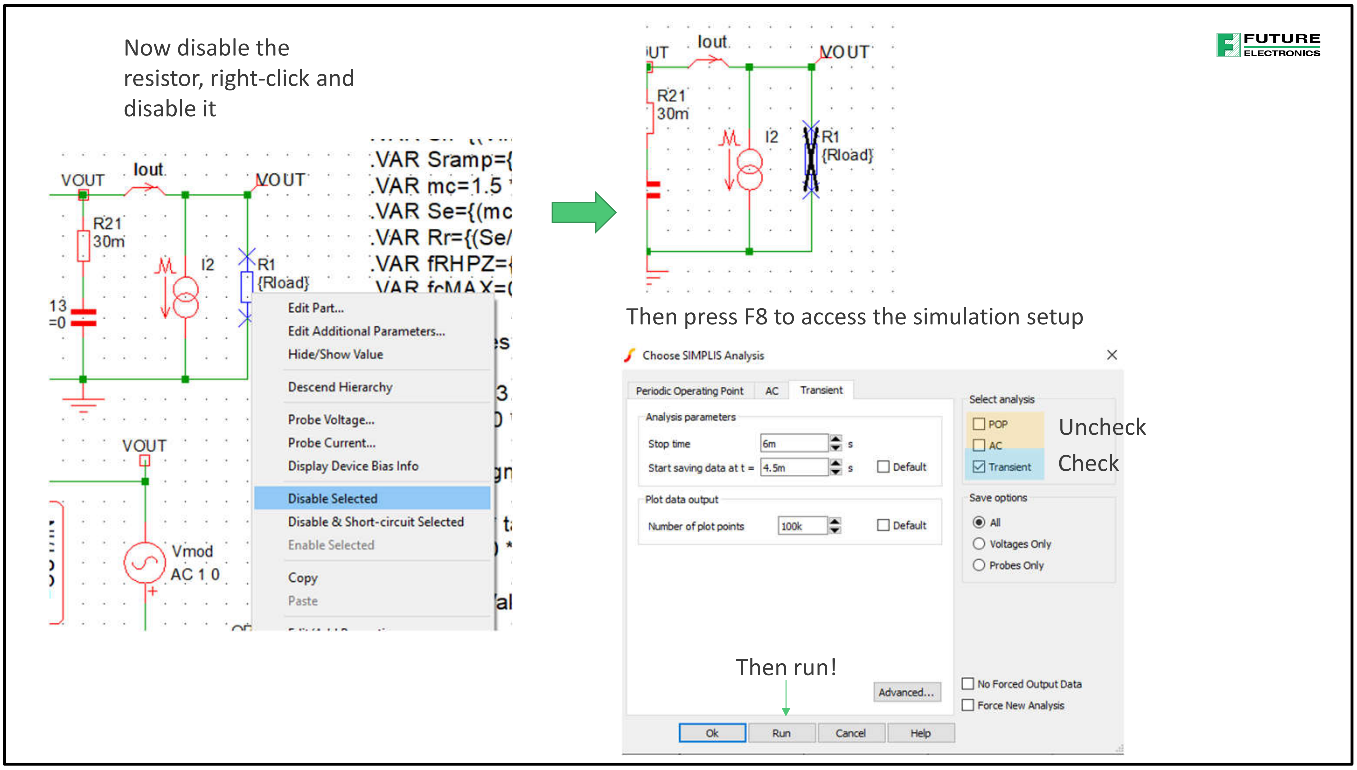


The automatically-calculated components values can be retrieved by invoking the Simulator pull-down menu and selecting Edit Netlist (after preprocess). You can also add more values to display in the control block if needed.

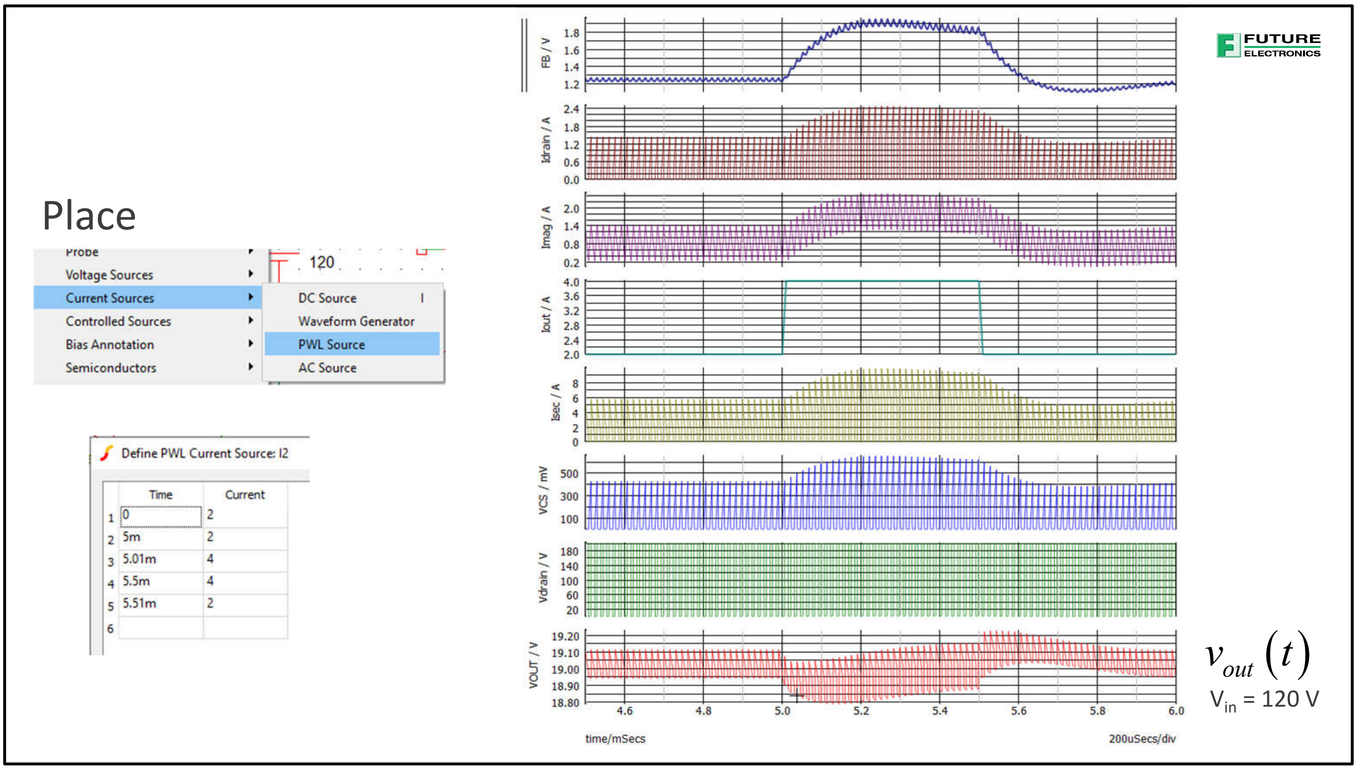


Now that the converter is properly compensated, it is interesting to test its transient response to a load step. As you can see, there is a big black shape – a cross - over the current source. This symbol means the current source is disabled. We are going to enable it for the next run: right-click on the part and select Enable Selected: the cross disappears.

“There’s a big black shape looking up at me” – Live Evil, Black Sabbath

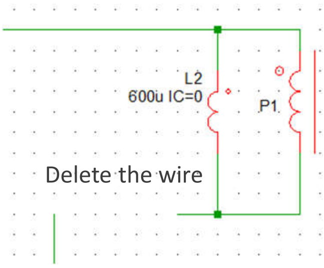


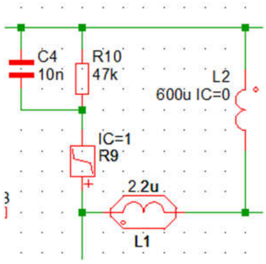
Now select the load resistance and right-click over the symbol: choose Disable Selected and the resistance disappears from the netlist, confirmed by the black cross over the component. Press F8 to invoke the simulation setup and choose Transient analysis. Press Run to launch the simulation engine.

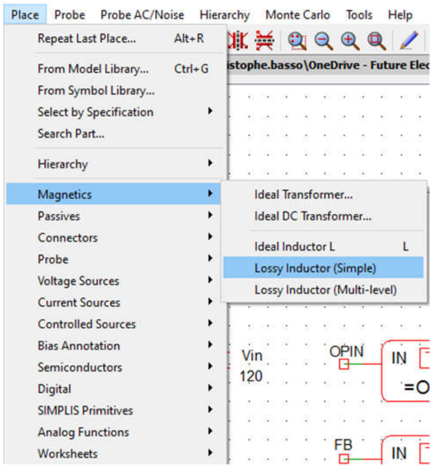


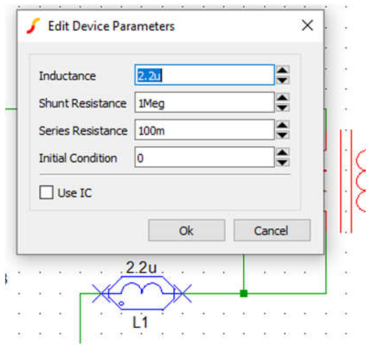
The transient current source is a piece-wise linear (PWL) programable source which absorbs 2 A from the start and peaks from 2 to 4 A at 5.01 ms then comes back to 2 A a few hundred of micro-seconds later.

We can now try to add a RCD clamping network



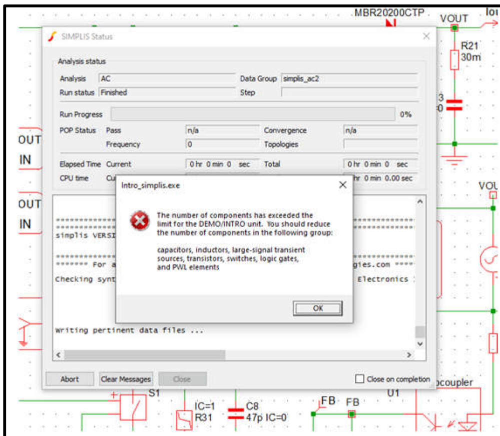




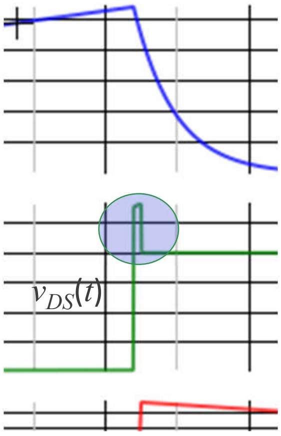


The lossy inductor offers more robust convergence by including parasitics

The RCD clamping network is there to limit the voltage excursion on the drain. Calculating components values goes beyond this seminar and this is a critical point that needs to be carefully checked depending on the leakage inductance value.

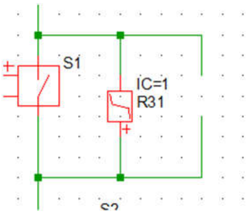


If you run the simulation, you now exceed the demo size limit.

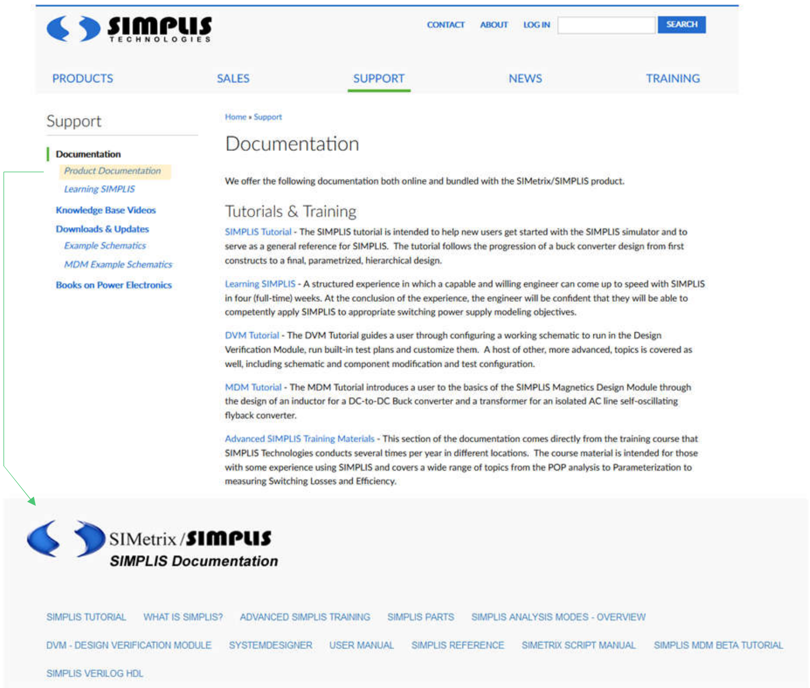


The drain-source voltage now shows the spike linked to the leakage inductance.

Delete the DS capacitance



You can delete the drain-source capacitance and Elements now lets you proceed with the simulation. As expected, the drain-source voltage shows the high-voltage spike incurred to the added leakage inductance. A more complex switch model would be needed to show oscillations when the RCD diode blocks.



https://www.simplistechnologies.com/support/documentation

The documentation database of SIMPLIS is a truly comprehensive source of information on the program with many examples, tutorials and videos.

For more information or additional details on specific subjects, the on-line documentation is extremely comprehensive and easily accessible.