



PLECS

Tutorial

Introduction to PLECS Standalone

Start learning PLECS Standalone by building two simple electrical circuits

Last updated in tutorials release 1.0

www.plexim.com

- ▶ Request a PLECS trial license
- ▶ Check the PLECS documentation

1 Introduction

The only way to become familiar with a new program is by using it. This tutorial has you model two example circuits step-by-step and in doing so demonstrates the basic features of PLECS.

In this exercise you will learn the following:

- How to build a simple model using components from the PLECS Library browser
- How to interconnect components on the PLECS schematic
- How to parametrize the components
- How to create a subsystem
- How to build a multi-domain model with controls
- How to use the Probe block

Before you begin Before you begin, ensure that you have installed PLECS Standalone and have the reference files that you can compare with your own models at each stage of the exercise.

2 Building a Simple Passive Network

The first electrical system we are going to model is a simple RLC network, as shown in Fig. 1. A capacitor is charged by a DC voltage source via an RL-branch and its voltage is monitored with a voltmeter.

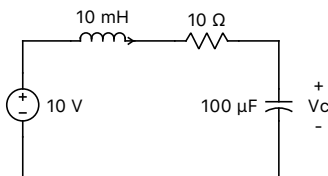


Figure 1: Simple RLC network



Your Task: Start PLECS. After the program loads the PLECS Library Browser is displayed. In order to enter the example circuits in PLECS we need to open a **New Model** from the **File** menu of the Library Browser.

The yellow schematic window that appears is the modeling interface. In the libraries you find various components from which you can create your circuits. You can browse through the available libraries and see which components are available. You may always pull up the Library Browser from the **Window** menu of the schematic window or pressing **Ctrl-L** (for Mac users, use **Cmd** instead of **Ctrl** for all shortcuts from here on).

Components

The components required for building the circuit must be dragged and dropped into the model window from the Library Browser. If you want to duplicate an existing component already present in the model window, hold down the **Ctrl** key, drag and drop the component to a new location or use the right mouse button (of course you can also select the component and **Copy** and **Paste** as usual).

The electrical components that you need in order to build the RLC network can be found in the Electrical domain library, under the sub-libraries Sources, Meters and Passive Components. The Scope component is located in the System library. Instead of browsing for the components by clicking on the arrows next to the domain names or using the arrow keys on your keyboard once something is selected, you can also search for them by typing the first letters of the component you need in the search bar. For example, typing `sc` displays the scope or typing `res` displays all the available resistors etc.

Signals

In addition to the electrical connections (black wires) that are used to connect electrical components, PLECS also makes use of unidirectional signals. The signals are in green and have an arrowhead to indicate their direction. These signals carry non-electrical information like measurement values or triggering pulses for switches. Signals can be used in calculations or displayed in a scope. Electrical connections cannot be fed into a scope directly, you always have to use a voltmeter or an ammeter to convert the electrical quantities into a signal first.



Your Task:

- 1 Drag and drop the Voltage Source DC, Inductor, Resistor, Capacitor, Voltmeter and Scope components from the library onto the schematic window, as shown in Fig. 2.
- 2 You can move the required components with the left mouse button. To rotate selected components press **Ctrl-R**, to flip them horizontally press **Ctrl-F**, or to flip them vertically press **Ctrl-I**. All these functions can also be accessed from the **Format** menu.



After you have copied all components to the schematic window, your model should be the same as the reference model `rlc_network_1.plecs`.

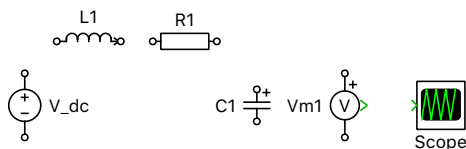


Figure 2: Components of the RLC network

Connections

Each component in the current schematic has black circles or green arrowheads to represent terminals. If we bring the mouse pointer close to such a terminal the pointer shape changes from an arrow to a cross. We now can drag a connection to another component by holding the left mouse button down. When we approach another terminal or an existing connection of the same connection type, the pointer shape changes into a double cross. As soon as we release the mouse button a new connection will be created.

For creating a branch connection from an existing connection, place the mouse cursor over it, hold **Ctrl**, then click and drag your cursor in the desired direction where you want the branch to start.



Your Task: Interconnect the components to complete the RLC circuit. Then connect the output terminal of the voltmeter to the input terminal of the scope.



At this stage, your model should be the same as the reference model `rlc_network_2.plecs`, as shown in Fig. 3.

Component Parameters

Each component is identified by a unique name, which is chosen automatically. If desired, this name can be changed by double-clicking on the component in the schematic. This is intended for documentation purposes only and does not affect the simulation. The parameters, on the other hand, are of

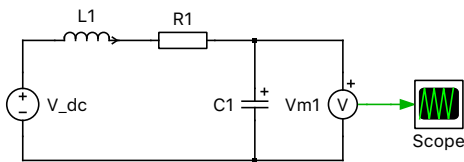


Figure 3: RLC network with scope

greater importance. The parameters determine, for example, the inductance of an inductor, the capacitance of a capacitor, or the voltage of a DC voltage source. Double-clicking on the component icon opens a dialog window in which you can set these parameters, as shown in Fig. 4.

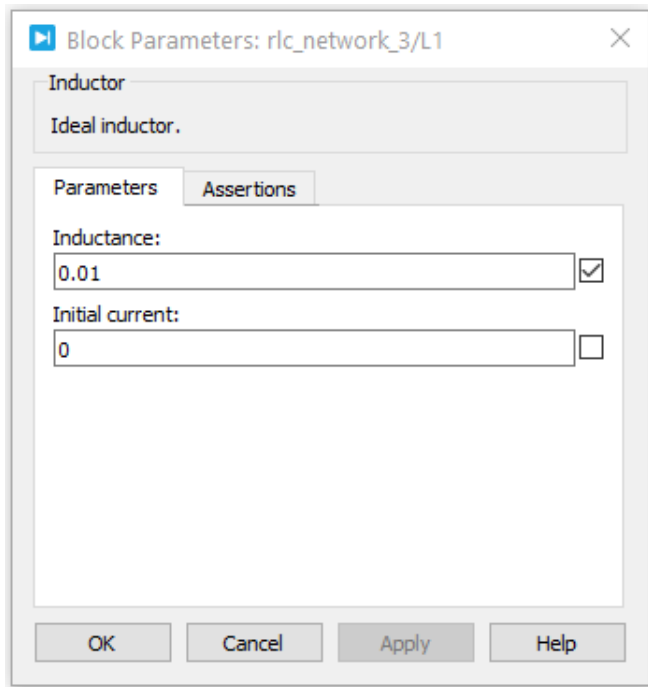


Figure 4: Parameter dialog window of an inductor

Notice that there is a checkbox to the right of each parameter field. If you check this checkbox, that specific parameter value will be displayed at the schematic level. This allows you to view and edit component parameters directly from the schematic. To avoid unnecessary text in the schematic, it is preferable to only display the most important parameters of a component.

Units


PLECS does not understand units since letters can be used as custom variables. Therefore, the user must make sure that the variables are scaled correctly. For power electronics, it is recommended to use the SI quantities. For example, use $10e-3$ or 0.001 , instead of 10 mH for an inductance value. For power systems, it may be more appropriate to work with “per unit” quantities.



Your Task:

- 1 Double-click on each component of the RLC network to display its parameter dialog window. Then set the source **Voltage** to 10, **Inductance** to 0.01, **Resistance** to 10 and the **Capacitance** to $100e-6$.

- 2 Check the checkboxes next to these parameters to display them on the schematic.
- 3 The inductor and capacitor have additional parameters such as **Initial current** and **Initial voltage**, respectively. Leave both these values as their defaults of zero.

 At this stage, your model should be the same as the reference model `r1c_network_3.plecs`, as shown in Fig. 5.

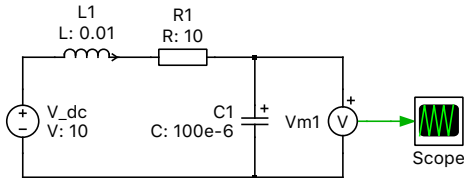


Figure 5: RLC network with parameters


Simulation

The simulation parameters such as the simulation time, solver type, solver options, model initialization commands etc., are located in the **Simulation Parameters** dialog window from the **Simulation** menu. This dialog can also be opened using the keyboard shortcut **Ctrl-E**.



Your Task:

- 1 Open the Simulation Parameters window and set the simulation **Stop time** to 0.1. Leave the other parameters as their default values.
- 2 Next, start the simulation by selecting **Start** from the **Simulation** menu or by pressing **Ctrl-T**.

 Your simulation results obtained should be the same as those shown in Fig. 6.

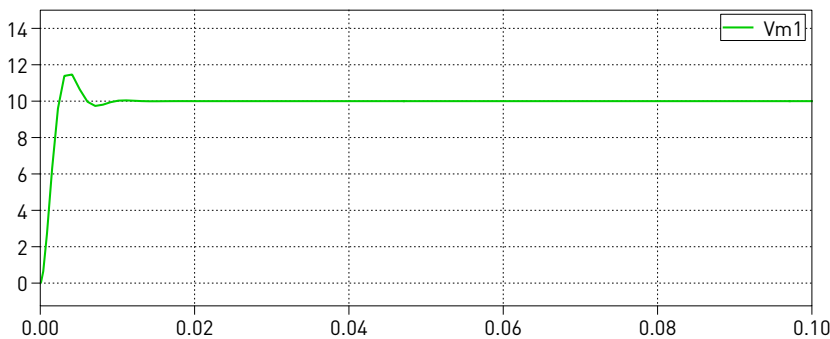


Figure 6: Simulation results of the RLC network capacitor voltage

Adding Control Blocks

To enhance the static electrical model, we can add some dynamic behavior. Let's observe how the capacitor in our example charges and discharges if we apply a pulsed voltage.



Your Task:

- 1 In the schematic, delete the Voltage Source DC component and replace it with a Voltage Source (Controlled) block. This block accepts any signal from the Control domain as an input.
- 2 Drag and drop a Pulse Generator component to the left of the voltage source. Set the **High-state output** to 10 and **Frequency** to 25. Leave all the other parameters as their default values.
- 3 Connect the output terminal of the Pulse Generator block to the input terminal of the controlled voltage source.
- 4 Now drag and drop a Signal Multiplexer component in between the Voltmeter and Scope blocks and set the number of inputs to 2.
- 5 Using the Signal Multiplexer, connect the outputs of the Pulse Generator and the Voltmeter to the Scope.
- 6 Start the simulation.



At this stage, your model should be the same as the reference model `rlc_network_final.plecs`, as shown in Fig. 7. Your simulation results obtained should be the same as those shown in Fig. 8.

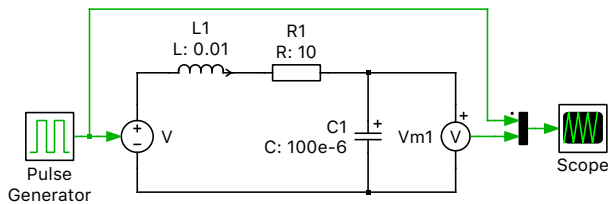


Figure 7: RLC network with a pulsed voltage source

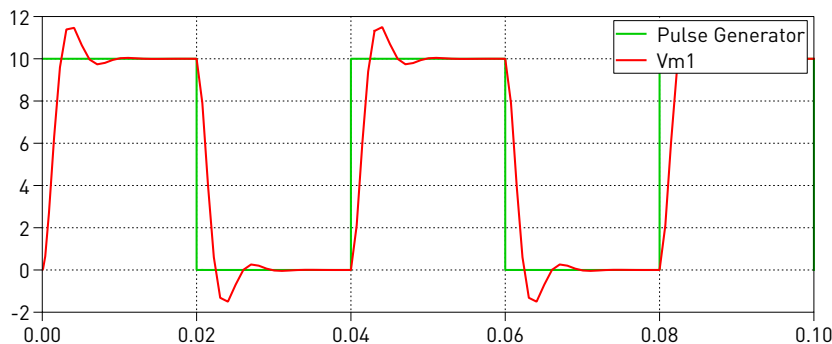


Figure 8: Simulation results of the RLC network with dynamic stimulation

3 Buck Converter

Next, in this section we will introduce the concept of ideal switches. We will build the buck converter shown in Fig. 9 along with a closed-loop controller.

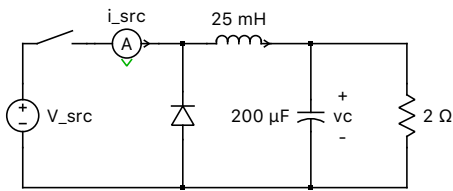


Figure 9: Schematic of the buck converter

Switches

For the buck converter shown in Fig. 9, the transistor can be modeled using a controllable switch, or an ideal IGBT or MOSFET component, bearing in mind that it may conduct current in only one direction. We also need a free-wheeling diode. The diode is a switch that closes as the voltage across it becomes positive, and opens as the current through it becomes negative.

The diode can be found in the Electrical/Power Semiconductors library and the switch in the Electrical/Switches library. All components in these two libraries are based on ideal switches that have zero on-resistance (you can optionally add some) and infinite off-resistance. They open and close instantaneously. For more information, right-click on any component in the library and select **Help**.

The switch is controlled by an external signal. It will close upon a non-zero input and open when the signal goes back to zero.



Your Task:

- 1** Open a new schematic window and drag and drop the Switch and Diode components from the library onto this window.
- 2** Build the rest of the electrical part of the buck converter, as shown in Fig. 10.



Note: To hide a component's name on the schematic, right-click on the component, then from the **Format** menu, uncheck **Show name** or use the keyboard shortcut **Ctrl-Shift-N**.



At this stage, your model should be the same as the reference model buck_converter_1.plecs.

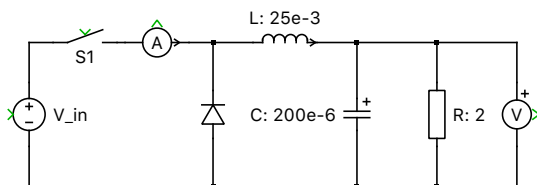


Figure 10: Electrical part of the buck converter

Subsystems

We can separate certain parts of the complete model from one another, such as the electrical part from the control part in this case. This has no effect on the simulation result but makes the whole system more structured. The use of subsystems can provide a clear presentation of the model and make debugging easier.



Your Task:

- 1 After completing the circuit shown in Fig. 10, select all the components (either by clicking an empty space in the upper left corner of the schematic and dragging a frame to the lower right corner, or by using the keyboard shortcut **Ctrl-A**).
- 2 Create a new subsystem by clicking on **Create Subsystem**, either from the **Edit** menu, or by right-clicking on one of the selected components, or using the keyboard shortcut **Ctrl-G**. The electrical components should now be in a new subsystem called “Sub”.
- 3 Optionally, you can rename the subsystem to, e.g. “Circuit”.



Note: The size of the subsystem icon can be changed by dragging one of the selected corners. You can also move the name label to another position by clicking and dragging it to the borders or the corners of the icon.

Signal Ports

To connect what is contained within the subsystem to the outer schematic we need to place ports into it.



Your Task:

- 1 Drag two Signal Inports and two Signal Outports into the subsystem schematic.
- 2 Connect the Signal Inports to the voltage source and the switch, respectively.
- 3 Connect the Signal Outports to the voltmeter and the ammeter, respectively.
- 4 Optionally, rename the signal ports.



Note: You will notice that a new terminal appears in the subsystem icon for each port that you drag into the subsystem schematic. To move these terminals along the icon’s frame, drag them with the mouse while holding down the **Shift** key.



At this stage, your model should be the same as the reference model `buck_converter_2.plecs`.

Control

For the buck converter, we will implement a hysteresis type control that keeps the capacitor voltage roughly in a $\pm 0.2\text{V}$ band around 6V . This involves using a Constant block as a reference set point and feeding back the measured load voltage. An error signal is then generated as the difference of the set point and measured voltage, and this is then fed to a Relay block whose output is now the gating signal to the switch. The control scheme here sets the gate signal to the switch high if the load voltage falls below 5.8V , and the gate signal to low if the voltage increases above 6.2V . We are simply regulating the current flow through the inductor and therefore the charge state of the capacitor.



Your Task:

- 1 Drag and drop a Relay, Sum, and Constant block from the Control domain library onto the schematic window.
- 2 Build the controller shown in Fig. 11.

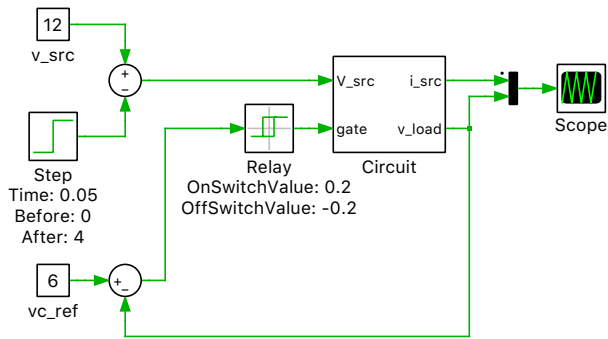



Figure 11: Buck converter with hysteresis control

3 To make things a bit more interesting, apply a step change, from 12 V down to 8 V, to the input voltage during the simulation.

4 Set the simulation **Stop time** as 0.1 s and start the simulation.

 At this stage, your model should be the same as the reference model `buck_converter_3.plecs`. Your simulation results obtained should be the same as those shown in Fig. 12.

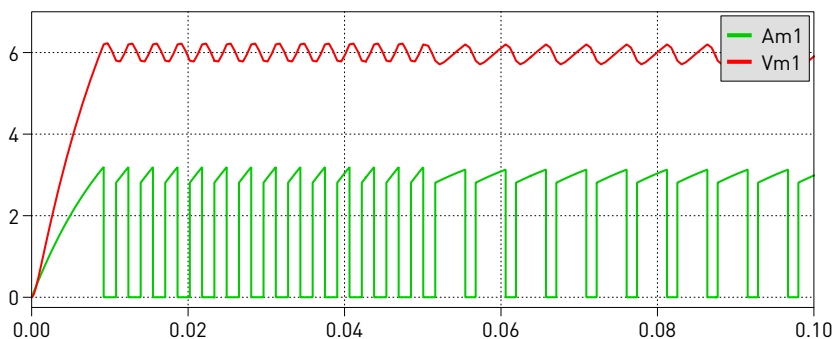


Figure 12: Simulation results of buck converter with hysteresis control

Probe

Instead of using an Ammeter or Voltmeter component to measure the signals, the signals can also be measured using a Probe block. This component can help to keep a model organized by reducing unnecessary connections, and gives you access to quantities that are difficult to measure otherwise. To associate a component with the Probe block, drag the component of interest onto it in the schematic and release your mouse once an arrow appears. Don't worry - if you are doing this correctly, as soon as you release the mouse, the component that you were dragging will return back to its original place. You can also double-click on the Probe block and drag the component of interest into the Probe Editor window that opens.

Your Task:

- 1** Drag and drop a Probe block from the System library onto the schematic window.
- 2** Associate the Voltage Source and Resistor components from the "Circuit" subsystem and select the **Source current** and **Resistor voltage** component signals, respectively.

- 3** Add a second plot to the scope window by right-clicking on the existing plot and selecting **Insert plot above**.
- 4** Connect the output terminal of the Probe to the newly created input terminal of the Scope block.
- 5** Start the simulation. You will notice that both the upper and lower plots are identical.



At this stage, your model should be the same as the reference model `buck_converter_final.plecs`.

4 Demo Models

Now that you've built your first models in PLECS it may be worthwhile to take a look at the demo models that come with PLECS. Open the demo model browser by selecting **Demo Models** from the **View** menu.

Revision History:

Tutorials 1.0 First release

How to Contact Plexim:

☎	+41 44 533 51 00	Phone
	+41 44 533 51 01	Fax
✉	Plexim GmbH Technoparkstrasse 1 8005 Zurich Switzerland	Mail
@	info@plexim.com	Email
	http://www.plexim.com	Web

PLECS Tutorial

© 2002–2020 by Plexim GmbH

The software PLECS described in this document is furnished under a license agreement. The software may be used or copied only under the terms of the license agreement. No part of this manual may be photocopied or reproduced in any form without prior written consent from Plexim GmbH.

PLECS is a registered trademark of Plexim GmbH. MATLAB, Simulink and Simulink Coder are registered trademarks of The MathWorks, Inc. Other product or brand names are trademarks or registered trademarks of their respective holders.