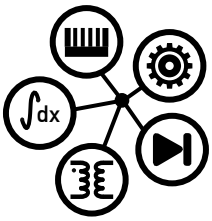

NI VeriStand Target

*DEMO MODEL*

# RL Load Control

**Loopback control of a simple first-order system**

Last updated in NI VeriStand TSP 1.0.1

**www.plexim.com**

- Request a PLECS and PLECS Coder trial license
- Get the latest NI VeriStand Target Support Package
- Check the PLECS and NI VeriStand documentation

# 1 Overview

This demo model shows an NI PXIe system used to simulate the response of a resistive-inductive (RL) circuit. The model also includes a PI controller which regulates the flow of current in the RL circuit following a disturbance.

The model is configured as a “virtual prototyping” platform where both the plant model and the controller are run on the same target hardware in real-time, highlighting the hardware’s capability as a hardware-in-the-loop (HIL) platform and real-time controller. Virtual prototyping allows testing the plant and controller models without any additional hardware connected to the real-time target.

To run this model the following items are needed:

- A PLECS Standalone or Blockset license supporting version 4.5.4 or newer and PLECS Coder license.
- The NI VeriStand Target Support Library must be downloaded and installed on the user PC. Follow the step-by-step instructions on configuring PLECS and the NI VeriStand target in the Quick Start guide of the NI VeriStand Target Support User Manual[1].
- An NI PXI or PXIe system with NI Linux RT and compatible NI-DAQ modules. A complete list of supported modules is included in the NI VeriStand Target Support User Manual[1].
- IO cables, breakout board, and accessories to connect NI-DAQ analog input and output signals.

---

**Note** This model contains model initialization commands that are accessible from:

*PLECS Standalone:* The menu **Simulation + Simulation Parameters... + Initializations**

*PLECS Blockset:* Right click in the **Simulink model window + Model Properties + Callbacks + InitFcn***

---

# 2 Model

An overview of the model is shown in Fig. 1. The model contains one subsystem which includes both the plant model and controller. The subsystem is enabled for code generation from the **Edit + Subsystem + Execution settings...** menu. This step is necessary to generate the model code for NI VeriStand target.

Input and output signals from the plant connect to the controller, and vice-versa, by using a loopback approach. The circuit current measurement, “iL_plant”, represents an analog output signal that is connected to the controller input, “iL_control”. Similarly, the output voltage from the controller is connected to a controlled voltage source within the plant.

When the analog output signal changes, the change in value will not be detected until the next execution of the model code. This latency is explicitly modeled by the Delay components shown in Fig. 1.

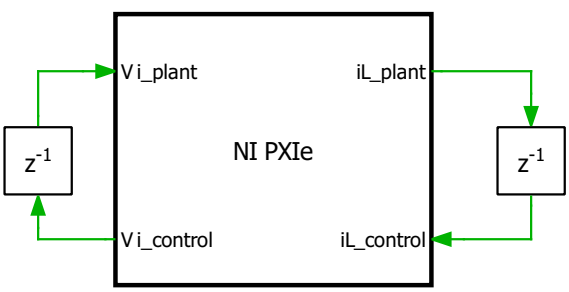


**Figure 1: Top-level view of the schematic showing signal loopback and delays**

Both the electrical plant and control circuit are included in the “NI PXIe” subsystem, as shown in Fig. 2. Components from the NI VeriStand Target Support Library are used to automatically configure the analog outputs and inputs of the NI-DAQ hardware.

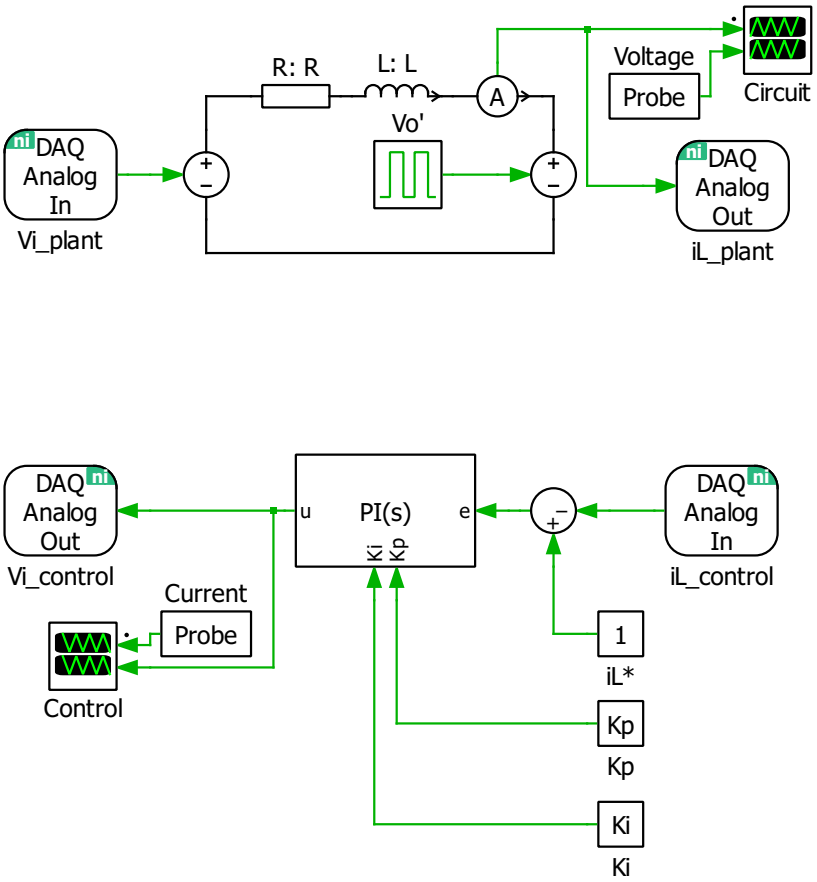


**Figure 2: Overview of the plant model (top) and controller (bottom)**

## 2.1 Plant

The plant consists of an RL circuit connecting an input voltage source and an output voltage. Both voltage sources are controllable. A pulse generator periodically toggles the output voltage every 200 msec to simulate a disturbance in the plant. The input voltage to the circuit is then adjusted upwards to regulate the flow of current in the RL circuit.

The plant transfer function can be defined as follows, where $V_{\mathrm{RL}}$ is the voltage across the RL circuit (i.e. $V_{\mathrm{RL}} = V_{\mathrm{i}} - V_{\mathrm{o}}$). The plant therefore has one real pole with a time constant of $\tau = \frac{L}{R}$.

$$P(s) = \frac{I_{\mathrm{L}}}{V_{\mathrm{RL}}} = \frac{1}{R + sL} = \frac{1/L}{R/L + s}$$

## 2.2 Controller

The controller regulates the inductor current to the “iL” set point of 1.0 A. The PI controller gains are adjustable by changing the “Kp” and “Ki” constants on the main schematic.

The PI controller has the following transfer function:

$$C(s) = \frac{V_{\mathrm{o}}}{I_{\mathrm{L}}} = K_{\mathrm{p}} + \frac{K_{\mathrm{i}}}{s} = \frac{sK_{\mathrm{p}} + K_{\mathrm{i}}}{s} = \frac{\alpha(s + \omega_{\mathrm{c}})}{s}$$

When the small latencies associated with the loopback delay are ignored, the resulting open loop transfer function is:

$$OL(s) = P(s)C(s) = \frac{1/L}{R/L + s} \frac{\alpha(s + \omega_c)}{s}$$

Selecting $\omega_c = R/L$ results in a cancellation of the pole and the open loop transfer function becomes $OL(s) = \alpha/(Ls)$. Therefore by adjusting $\alpha$ the crossover frequency can be directly selected. In this model the crossover frequency is initially selected as $\omega_c = 2\pi 50$ rad/s.

## 2.3 Model discretization

As part of the code generation process the PLECS model must be converted into a discretized representation of the system. A discretization step size, $T_d$, must be chosen. For the discretized model to approximate the continuous system the $T_d$ chosen must be sufficiently less than the time constants associated with the plant. In this model the discretization step size is selected as $T_d = \tau/20$, or $1/20^{th}$ the RL circuit's time constant. As $T_d$ increases, the impact of the latencies associated with the loopback delays will impact the closed loop response of the system. If the chosen $T_d$ is too large, then the closed loop system will become unstable. However, if the chosen $T_d$ is too small, then the real-time processor will not have sufficient time to execute the model prior to the next model step, resulting in missed model steps and poor model performance.

# 3 Simulation

In addition to running the model offline, the model can be deployed to NI target hardware either through the NI VeriStand software or directly via PLECS.

With the NI VeriStand Target Support package, PLECS generates a compiled model that can be imported into NI VeriStand. The generated model will have inport and outport signals corresponding to the blocks shown in Fig. 2. When the **Build type** parameter in the **Coder + Coder options + Target** tab is set to `VeriStand engine` or `Custom engine`, then the inports and outports are automatically mapped to the appropriate NI-DAQ hardware ports. If the **Build type** is set to `Model only`, then the inports and outports can manually be assigned within the VeriStand software.

The "iL" current reference and PI controller gains are configured to be adjustable in real-time while connected to the target. To configure additional parameters as adjustable, open the **Coder + Coder options...** menu and navigate to the **Parameter Inlining** tab. When a component from the schematic is dragged and dropped into the **Exceptions** list, tunable parameters associated with that component will be tunable during runtime. Note this behavior depends on the **Default behavior** setting, as the **Exceptions** list specifies components which have opposite behavior of the default setting.

The model is configured for the `Model only` build type, but instructions on how to configure the model for all build types are provided below. Detailed instructions on configuring the hardware target and other build options are available in the NI VeriStand Target Support User Manual [2].

---

**Note** The available hardware on the target device may not correspond to the default hardware IO specified in the demo model. Before proceeding generate a new hardware configuration file (`*.nce`) from NI Max for the target machine and update the slot numbers and IO channels used in the model. Refer to the *Quickstart* section of the NI VeriStand Target Support User Manual [2] for instructions on how generate a new hardware configuration file.

---

### Model only

The instructions below show the steps to manually import a model into NI VeriStand using the `Model only` build type:

- Open the **Coder + Coder options...** drop-down menu and select the "NI PXIe" subsystem from the **System** tab visible in the left-hand side of the window.
- Navigate to the **Target** tab and set the build type to `Model only` and click **Build**.
- The default path to the generated model is a directory titled `engine_demo_codegen` next to the model file. The name of the generated model will correspond that of the "Engine" subsystem and will have a `*.so` extension.
- Open NI VeriStand and import the model. Refer to the *Adding and Configuring a Model* section of the NI VeriStand manual [1] for step-by-step instructions on how to import the model generated from PLECS.
- Within NI VeriStand, manually map the available analog input and outputs to the model inports and outports.

### VeriStand Engine

The instructions below outline the steps to automatically create a complete VeriStand project using the `Veristand engine` build type:

- Open the **Coder + Coder options...** drop-down menu and select the "NI PXIe" subsystem from the **System** tab visible in the left-hand side of the window.
- Navigate to the **Target** tab and set the build type to `Veristand engine`. Enter in the settings of the NI real-time target and generate a new hardware configuration file if necessary.
- Select if the model will automatically be deployed to the target. If so, open the NI VeriStand application. Once the application is open, click **Build**.
- Open the generated VeriStand project. The default directory is titled `RL_load_control_codegen` next to the model file. The generated project will be titled "NI_PXIe" and have a `*.nivsprj` extension.
- The VeriStand application or PLECS can be used to interact with the model when it is executing on the NI real-time target. To connect via the PLECS External Mode open the **Coder + Coder options...** window and select the **External Mode** tab. Enter `localhost` for the **Target device** and click **Connect**.

### Custom Engine

The instructions below highlight the steps to deploy a model to NI real-time hardware using the `Custom engine` build type:

- Open the **Coder + Coder options...** drop-down menu and select the "NI PXIe" from the **System** tab visible in the left-hand side of the window.
- Navigate to the **Target** tab and set the build type to `Custom engine`. Enter in the settings of the NI real-time target and generate a new hardware configuration file if necessary.
- Select if the model will automatically be deployed to the target and click **Build**.
- To connect to a model executing on the real-time target via the PLECS External Mode, open the **Coder + Coder options...** window and select the **External Mode** tab. Enter the IP address of the remote target (e.g. `192.168.0.105`) for the **Target device** and click **Connect**.

Fig. 3 and Fig. 4 show a comparison of the offline results and the real-time results when the model is deployed to the target using the custom engine. The plots are aligned by setting the **Trigger channel** option to `5: [Vo:Source voltage]` and the trigger level to 2.0 and clicking the **Activate autotriggering** button. The offline and real-time results show a close alignment. Increasing the chosen discretization time, $T_\mathrm{d}$, will adversely impact the alignment between the offline and real-time simulation.

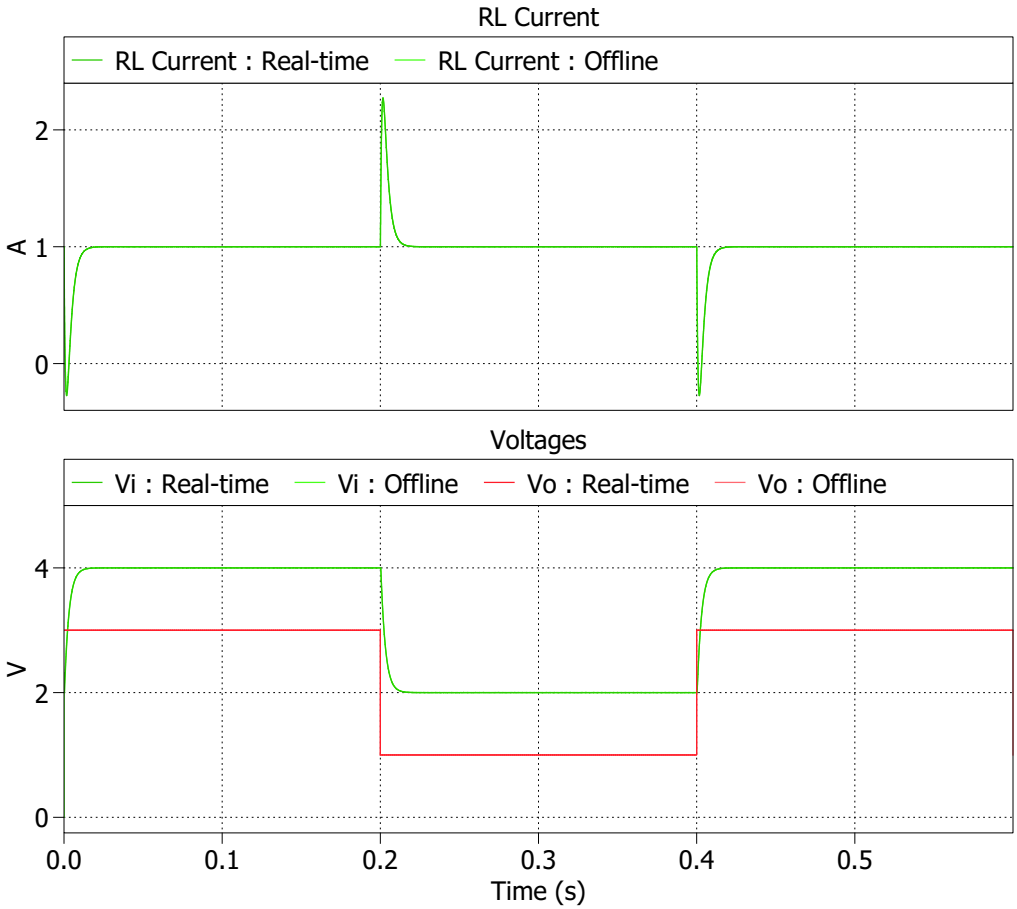


**Figure 3: Benchmarking of offline and real-time results**

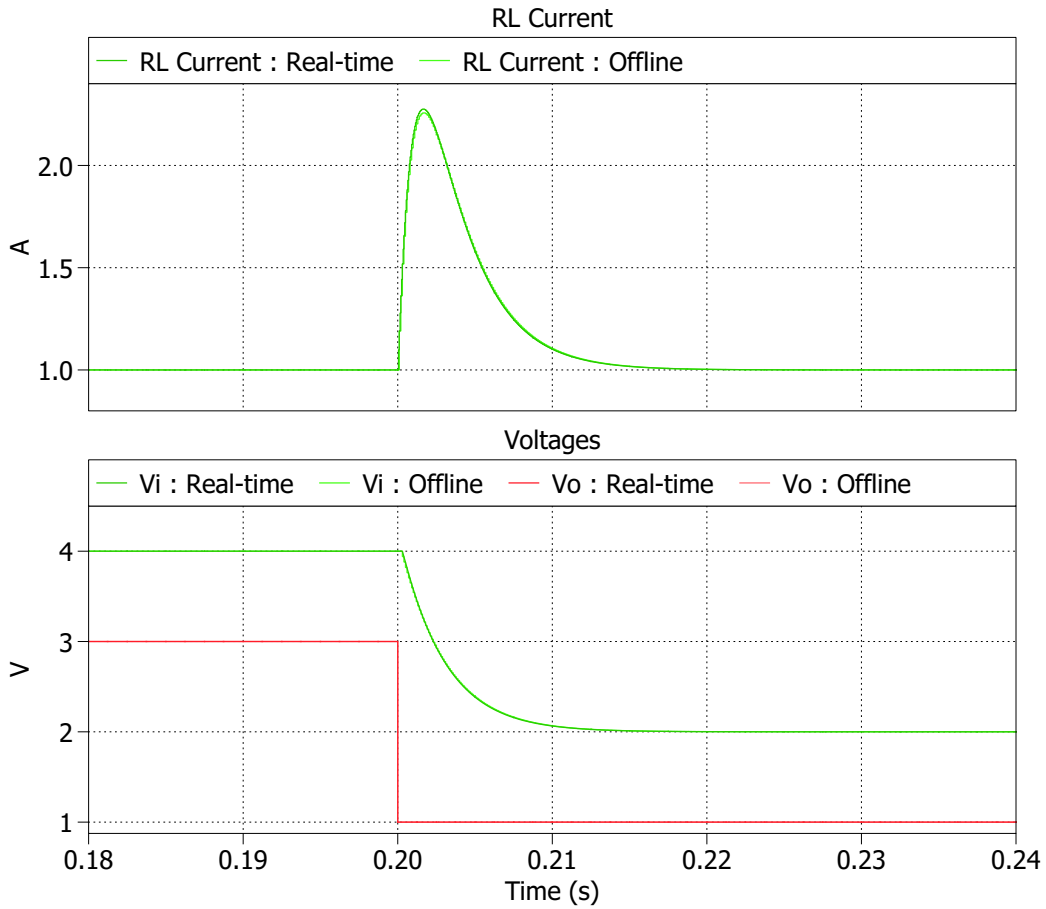


**Figure 4: Benchmarking of offline and real-time results - zoomed view of transient**

# 4 Conclusion

This model shows how to create a virtual prototyping system using the NI VeriStand Target Support Package using simple RL circuit and PI controller. In the virtual prototyping configuration, both the plant and controller are executed virtually on the target hardware. Hardware IO is looped back to enable closed loop operation of the system.

# References

[1] NI, *VeriStand Manual*, 2020,
URL: https://www.ni.com/documentation/en/veristand/latest.

[2] Plexim, *NI VeriStand Target Support User Manual*,
URL: https://www.plexim.com/download/documentation.

**Revision History:**

| | |
|---|---|
| NI VeriStand TSP 1.0.1 | First release |

**How to Contact Plexim:**


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


*NI VeriStand TSP Demo Model*


© 2002–2022 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.