Getting Started with the MapleSim Connector for LabVIEW and NI VeriStand Software

Copyright ${\ensuremath{\mathbb C}}$ Maplesoft, a division of Waterloo Maple Inc. 2020

Getting Started with the MapleSim Connector for LabVIEW and NI VeriStand Software

Copyright

Maplesoft, Maple, and MapleSim are all trademarks of Waterloo Maple Inc.

© Maplesoft, a division of Waterloo Maple Inc. 2010-2020. All rights reserved.

No part of this book may be reproduced, stored in a retrieval system, or transcribed, in any form or by any means — electronic, mechanical, photocopying, recording, or otherwise. Information in this document is subject to change without notice and does not represent a commitment on the part of the vendor. The software described in this document is furnished under a license agreement and may be used or copied only in accordance with the agreement. It is against the law to copy the software on any medium except as specifically allowed in the agreement.

National Instruments, LabVIEW, and NI VeriStand are registered trademarks or registered trademarks of National Instruments.

Microsoft, Windows, and Visual Studio are registered trademarks of Microsoft Corporation.

Linux is a registered trademark of Linus Torvalds.

Macintosh and Mac OS are registered trademarks of Apple Computer, Inc.

All other trademarks are the property of their respective owners.

This document was produced using Maple and DocBook.

Contents

Introduction	iv
1 Getting Started	. 1
1.1 Getting Help	
1.2 Using the LabVIEW Component Block Generation Template	. 1
Subsystem Preparation	. 1
Subsystem Selection	
Port and Parameter Management	
EMI Component Options	
SIT Component Options	
Generate SIT Component Code	
Generate EMI Component Code	
View EMI or SIT Component Code	
1.3 Using the LabVIEW Block Generation Templates	
Viewing Examples	
1.4 Example: RLC Circuit Model	
Generating a LabVIEW EMI Block	
Generating a LabVIEW Block for NI VeriStand or the LabVIEW SIT	
2 Example: Exporting a Model as a LabVIEW EMI Block	
2.1 Preparing a Model for Export	
Converting the Model to a Subsystem	
Defining Subsystem Inputs and Outputs	
2.2 Defining and Assigning Subsystem Parameters	
2.3 Exporting Your Model Using the LabVIEW EMI Block Generation Template	
3 Working with Your Block in NI VeriStand or LabVIEW SIT	
3.1 Preparing Your MapleSim Model to Run in NI VeriStand	
Creating a New Project File	
Adding the MapleSim Model to the System Definition File	
Running the Project	
Adding a Dial to the Workspace	
Adding a Graph to the Workspace	
3.2 Importing a MapleSim Model to the LabVIEW SIT Environment	
Creating a LabVIEW SIT Interface	
Connecting the MapleSim Model and the LabVIEW SIT User Interface	
4 Running a Simulation on a LabVIEW Real-Time Target Machine	
4.1 Preparing the LabVIEW Real-Time Project	
4.2 Moving the .dll File to the Target Real-Time Machine	
Index	32

Introduction

The MapleSimTM Connector for LabVIEW® and NI VeriStandTM Software provides all of the tools you need to prepare and export your dynamic systems models to National InstrumentsTM (NI) LabVIEW as External Model Interface (EMI) or Simulation Interface Toolkit (SIT) blocks, or as models for NI VeriStandTM. You can create a model in MapleSim, simplify it in MapleTM by using an extensive range of analytical tools, and then generate virtual instruments (VIs) that you can incorporate into your LabVIEW or NI VeriStand toolchain.

You can also use these tools for exporting mathematical models that you have created from first principles in Maple as VIs.

Furthermore, various options allow you to use the C code generation feature in Maple to create code libraries of your MapleSim models for implementation in other applications.

Features include:

- Maple templates, which provide an intuitive user interface for optimizing your MapleSim model, and then generate a dynamic-link library (.dll) file for LabVIEW or NI VeriStand.
- A range of examples illustrating how to prepare and export your models.
- Commands for developing VIs of mathematical models from first principles in the Maple environment and examples to illustrate how to do it.
- Access to commands in the LabVIEWConnector package in Maple for developing dynamic-link library (.dll) files for LabVIEW or NI VeriStand.

Scope of Model Support

MapleSim is a comprehensive modeling tool where it is possible to create models that could go beyond the scope of this MapleSim Connector for LabVIEW and NI VeriStand Software release. In general, the MapleSim Connector for LabVIEW and NI VeriStand Software supports systems of any complexity, including systems of DAEs of any index, in any mix of domains.

Requirements

NI LabVIEW 2011 or 2012 with at least one of the following:

- LabVIEW Control Design and Simulation Module (for generating an EMI block)
- LabVIEW Simulation Interface Toolkit (to generate a block for the LabVIEW Simulation Interface Toolkit)
- NI VeriStand 2011, 2012, or 2013 (to generate a block for NI VeriStand)

Also requires Microsoft Visual Studio 2013, 2015 or 2017.

For installation instructions and system requirements, see the **Install.html** file on the product disc or visit the Maplesoft System Requirements website at <u>http://www.maplesoft.com/products/system_requirements.aspx</u>.

Adding External Libraries to Your Search Path

You can export a model that uses an external library as part of the model to LabVIEW. In order to do this, you **first** need to add the directory that contains the external library file (that is, the .dll or .so file) to your search path. This involves appending the external library directory to either your PATH environment variable (for Windows®) or your LD LIBRARY PATH environment variable (for Linux® and Macintosh®).

To add an external library directory to your search path

1. Determine the location of the external library directory.

Note: This is the directory that contains the .dll file (Windows) or the .so file (Linux or Macintosh) that is used in your model.

- 2. Add the library directory found in step 1 to the appropriate environment variable for your operating system.
 - For Windows, add the library directory to your PATH environment variable.
 - For Linux and Macintosh, add the library directory to your LD_LIBRARY_PATH environment variable.

Consult the help for your operating system for instructions on how to edit these environment variables.

3. Restart your computer.

1 Getting Started

1.1 Getting Help

Refer to the LabVIEWConnector help page.

1.2 Using the LabVIEW Component Block Generation Template

The MapleSim Connector provides LabVIEW Component Block Generation templates in the form of Maple worksheets for manipulating and exporting MapleSim subsystems. These templates contain pre-built embedded components that allow you to generate EMI Components, SIT Components, or C code from a MapleSim subsystem, export the subsystem as a LabVIEW block, and save the source code.

Using these templates, you can define inputs and outputs for the system, set the level of code optimization, choose the format of the resulting EMI Component, and generate the source code, library code, block script, or LabVIEW block. You can use any Maple commands to perform task analysis, assign model equations to a variable, group inputs and outputs to a single vector and define additional input and output ports for variables.

Note: Code generation now handles all systems modeled in MapleSim, including hybrid systems with defined signal input (RealInput) and signal output (RealOutput) ports.

Block generation for EMI or SIT consists of the following steps:

- Subsystem preparation
- · Subsystem selection
- · Port and parameter management
- EMI or SIT component options
- Generate EMI or SIT component code
- View generated EMI or SIT component code

Subsystem Preparation

Convert your model or part of your model into a subsystem. This identifies the set of modeling components that you want to export as a block component. Since LabVIEW only supports data signals, properties on acausal connectors such as mechanical flanges and electrical pins, must be converted to signals using the appropriate ports.

To connect a subsystem to modeling components outside of its boundary, you add subsystem ports. A subsystem port is an extension of a component port in your subsystem. The resulting signals can then be directed as inputs and outputs for the LabVIEW Component Block Generation templates.

Note: For connectors you must use signal components since acausal connectors can not be converted to a signal.

By creating a subsystem you not only improve the visual layout of a system in **Model Workspace** and but also prepare the model for export. The example in Chapter 2 shows you how to group all of the components into a subsystem.

Subsystem Selection

You can select which subsystems from your model you want to export to a LabVIEW block. After a subsystem is selected, click **Load Selected Subsystem**. All defined input and output ports are loaded.

Port and Parameter Management

Port and Parameter Management lets you customize, define and assign parameter values to specific ports. Subsystem components to which you assign the parameter inherit a parameter value defined at the subsystem level. After the subsystem is loaded you can group individual input and output variable elements into a vector array, and add additional

input and output ports for customized parameter values. Input ports can include variable derivatives, and output ports can include subsystem state variables.

Note: If the parameters are not marked for export they will be numerically substituted.

Input Ports:

	Input Variables	Port Grouping Name	Change Row
1			

🔄 Group all inputs into a single vector 🛛 🔄 Add additional inputs for required input variable derivatives

Select **Group all inputs into a single vector** to create a single 'vector' input port for all of the input signals instead of individual ports. The order of the inputs are the same as given in the S-function mask window.

Select Add additional inputs for required input variable derivatives to specify calculated derivative values instead of numerical approximations.

Output Ports:

	Output Variables	Port Grouping Name	Change Row
1			

🔲 Group all outputs into a single vector 🛛 🔄 Add an additional output port for subsystem state variables

Select Group all outputs into a single vector to define outputs as an S-Function 'mask'.

Select Add an additional output port for subsystem state variables to add extra output ports for the state variables.

Parameters:

	Parameters	Value	Export	Updated Row
1				

Group all parameters into a single vector

Select **Group all parameters into a single vector** to create a single parameter 'vector' for all of the parameters in the S-function. If not selected, the S-function mask will contain one parameter input box for each of the S-function parameters.

EMI or SIT Input and Output Ports

The following selections specify the input ports, output ports, and states for generating LabVIEW blocks.

EMI or SIT Input Ports:

Add additional inputs for required input variable derivatives

Select Add additional inputs for required input variable derivatives to specify calculated derivative values instead of numerical approximations.

EMI or SIT Output Ports:

Add an additional output port for subsystem state variables

Select Add an additional output port for subsystem state variables to add extra output ports for the state variables.

EMI Component Options

The EMI Component Options settings specify the advanced options for the code generation process.

Optimization Options

Set the level of code optimization to specify whether equations are left in their implicit form or converted to an ordinary differential equation (ODE) system during the code generation process. This option specifies the degree of simplification applied to the model equations during the code generation process and eliminates redundant variables and equations in the system.

Level of code optimization (0=None, 3=Full): 0 1 2 3

Select one of the following options:

None (0): no optimization is performed; the default equations will be used in the generated code.

Partial (1, 2): removes redundant equations from the system.

Full (3): performs index reduction to reduce the system to an ODE system or a differential algebraic equation (DAE) system of index 1, and removes redundant equations.

Constraint Handling Options

The **Constraint Handling Options** area specifies whether the constraints are satisfied in a DAE system by using constraint projection in the generated LabVIEW block. Use this option to improve the accuracy of a DAE system that has constraints. If the constraint is not satisfied, the system result may deviate from the actual solution and could lead to an increase in error at an exponential rate.

Maximum number	3	
Error tolerance:	0.1e-4	

Apply projection during event iterations

Set the **Maximum number of projection iterations** to specify the maximum number of times that a projection is permitted to iterate to obtain a more accurate solution.

Set the Error tolerance to specify the desirable error tolerance to achieve after the projection.

Select Apply projection during event iterations to interpolate iterations to obtain a more accurate solution.

Constraint projection is performed using the **constraint projection** routine in the External Model Interface as described on The MathWorksTM web site to control the drift in the result of the DAE system.

Event Handling Options

The **Event Handling Options** area specifies whether the events are satisfied in a DAE system by using event projection in the generated LabVIEW block. Use this option to improve the accuracy of a DAE system with events. If the constraint is not satisfied, the system result may deviate from the actual solution and could lead to an increase in error at an exponential rate.

Maximum number of event iteration	is: 10
Width of event hysteresis band:	0.1e-9

Optimize for use with fixed-step integrators

Set the **Maximum number of event iterations** to specify the maximum number of times that a projection is permitted to iterate to obtain a more accurate solution.

Set the Width of event hysteresis band to specify the desirable error tolerance to achieve after the projection.

Select **Optimize for use with fixed-step integrators** to optimize the event iterations as a function of hysteresis bandwidth.

Baumgarte Constraint Stabilization

The Baumgarte constraint stabilization method stabilizes the position constraint equations by combining the position, velocity, and acceleration constraints into a single expression. By integrating the linear equation in terms of the acceleration, the Baumgarte parameters, alpha and beta, act to stabilize the constraints at the position level.

Select **Apply Baumgarte constraint stabilization** to apply Baumgarte constraint stabilization to your model. When selected, you can enter values for the derivative gain (**Alpha**) and the proportional gain (**Beta**) that are appropriate for your model.

Select **Export Baumgarte parameters** to add **Alpha** and **Beta** as parameters in the generated plugin solver code for your model. This allows you to change the values of **Alpha** and **Beta** when using your plugin solver.

📝 Apply Baumgarte constraint stabilization 🛛 📝 Export Baumgarte parameters

Alpha:	10	
Beta:	2	

Discretization

Select **Export as a discrete model (no continuous states)** to apply discretization to your model. When selected, you can select a solver type from one of the following options:

- Euler: forward Euler method
- RK2: second-order Runge-Kutta method
- RK3: third-order Runge-Kutta method
- RK4: fourth-order Runge-Kutta method
- Implicit Euler: implicit Euler method

In this section, you can also set the Discrete Timestep (in seconds) for the discretization.

Export as a discrete model (no continuous states)						
Embedded solver:	🔘 Euler) RK2	🔘 RK3	🔘 RK4	Implicit Euler	
Discrete Timestep	0.1e-2					

SIT Component Options

These settings specify the advanced options for the code generation process.

Optimization Options

Set the level of code optimization to specify whether equations are left in their implicit form or converted to an ordinary differential equation (ODE) system during the code generation process. This option specifies the degree of simplification

applied to the model equations during the code generation process and eliminates redundant variables and equations in the system.

Level of code optimization (0=None, 3=Full): 0 1 2 3

Select one of the following options:

None (0): no optimization is performed; the default equations will be used in the generated code.

Partial (1, 2): removes redundant equations from the system.

Full (3): performs index reduction to reduce the system to an ODE system or a differential algebraic equation (DAE) system of index 1, and removes redundant equations.

Constraint Handling Options

The **Constraint Handling Options** area specifies whether the constraints are satisfied in a DAE system by using constraint projection in the generated LabVIEW block. Use this option to improve the accuracy of a DAE system that has constraints. If the constraint is not satisfied, the system result may deviate from the actual solution and could lead to an increase in error at an exponential rate.

Maximum number of projection iterations: 3
Error tolerance: 0.1e-4
Apply projection during event iterations

Set the **Maximum number of projection iterations** to specify the maximum number of times that a projection is permitted to iterate to obtain a more accurate solution.

Set the Error tolerance to specify the desirable error tolerance to achieve after the projection.

Select **Apply projection during event iterations** to interpolate iterations to obtain a more accurate solution. Constraint projection is performed using the **constraint projection** routine in the External Model Interface as described on The MathWorksTM web site to control the drift in the result of the DAE system.

Event Handling Options

The **Event Handling Options** area specifies whether the events are satisfied in a DAE system by using event projection in the generated LabVIEW block. Use this option to improve the accuracy of a DAE system with events. If the constraint is not satisfied, the system result may deviate from the actual solution and could lead to an increase in error at an exponential rate.

Event Handling Options:					
Maximum number of event iteration	ns:	10			
Width of event hysteresis band:	ο.:	1e-9]		

 \checkmark Optimize for use with fixed-step integrators (Must be checked for SIT)

Set the **Maximum number of event iterations** to specify the maximum number of times that a projection is permitted to iterate to obtain a more accurate solution.

Set the Width of event hysteresis band to specify the desirable error tolerance to achieve after the projection.

Event projection is performed using the **event projection** routine in the External Model Interface as described on The MathWorksTM web site to control the drift in the result of the DAE system.

Baumgarte Constraint Stabilization

The Baumgarte constraint stabilization method stabilizes the position constraint equations by combining the position, velocity, and acceleration constraints into a single expression. By integrating the linear equation in terms of the acceleration, the Baumgarte parameters, alpha and beta, act to stabilize the constraints at the position level.

Select **Apply Baumgarte constraint stabilization** to apply Baumgarte constraint stabilization to your model. When selected, you can enter values for the derivative gain (**Alpha**) and the proportional gain (**Beta**) that are appropriate for your model.

Select **Export Baumgarte parameters** to add **Alpha** and **Beta** as parameters in the generated plugin solver code for your model. This allows you to change the values of **Alpha** and **Beta** when using your plugin solver.

📝 Apply	/ Baumgarte con:	straint stabilization	Export Baumgarte parameters
Alpha:	10]	
Beta:	2		

Baserate

The **Baserate** area specifies the rate at which the model runs. Use this option to improve the accuracy of a DAE system with events. If the constraint is not satisfied, the system result may deviate from the actual solution and could lead to an increase in error at an exponential rate.

Baserate:

The rate at which the model runs: 0.1e-2

Inputs

Specify the input type; internal, external or both.

Specify input type: "internal", "external" or both (default="external")

📄 internal 🛛 📝 external

Generate SIT Component Code

Target directory:

	Browse
SIT directory:	
	Browse
Visual C++ directory:	
	Browse
Block Name:	
Generate SIT Component Generate and Compile SIT Component to Veris	Stand

Provide a block name, SIT and Visual C++ directories and specify the location for the generated SIT file.

To generate SIT Component code without a VeriStand connection, click Generate SIT Component.

To generate and compile SIT Component code into VeriStand, click Generate and Compile SIT Component to VeriStand.

Generate EMI Component Code

Target directory:	
	Browse
LabVIEW (32-bit) directory:	
	Browse
Visual C++ directory:	
	Browse
Block Name:	
Generate EMI Component Generate and Compile EMI Component to Lab	VIEW

Provide a block name, LabVIEW and Visual C++ directories and specify the location for the generated EMI file.

To generate EMI Component code without a LabVIEW connection, click Generate EMI Component.

To generate EMI Component code, click Generate and Compile EMI Component to LabVIEW.

Note: If your model contains an external library, then you must add the directory that contains the external library to your search path. See *Adding External Libraries to Your Search Path (page iv)* for instructions on how to do this.

View EMI or SIT Component Code

After you generate the EMI Component code and create the block, a LabView command window opens and the block with any of the following specified parameters is generated in LabVIEW:

- Header File
- C Code

1.3 Using the LabVIEW Block Generation Templates

The MapleSim Connector for LabVIEW and NI VeriStand Software provides an NI LabVIEW EMI Block Generation template and an NI VeriStand and LabVIEW SIT Component Model Generation template in the form of Maple worksheets for manipulating and exporting MapleSim subsystems. These templates contain pre-built embedded components that allow you to generate LabVIEW blocks from a MapleSim subsystem, export the subsystem as a LabVIEW block and Microsoft® Visual Studio® project, and save the source code.

Using either of these templates, you can define inputs and outputs for the system, generate the source code and library code.

Example models are available in the **NI Connector Examples** palette in MapleSim. To access them, from the Help menu, select **Examples > NI connector Examples**.

Viewing Examples

To view an example:

1. From the **Help** menu, select the **Examples** > **NI Connector Examples** menu, and then click the entry for the model that you want to view.

Some models include additional documents, such as templates that display model equations or define custom components.

2. In the Attached Files tab, expand Documents. You can open any of these documents by right-clicking its entry in the list and clicking View.

After you add a template to a model, it becomes available from this list.

1.4 Example: RLC Circuit Model

In this example, you will generate a LabVIEW EMI or SIT block, or a block for NI VeriStand using an RLC circuit model that was created in MapleSim.

To generate a LabVIEW block

- 1. From the Help menu, select Examples > NI Connector Examples, and then select the RLC Parallel Circuit example.
- 2. Select the Add Apps or Templates tab (🔠).
- 3. In the **Templates** palette, double-click on either **NI LabVIEW EMI Component Block Generation** to generate a LabVIEW EMI block or **NI VeriStand and LabVIEW SIT Component Model Generation** to generate a block for the LabVIEW Simulation Interface Toolkit or NI VeriStand.
- 4. Enter **RLC Circuit** as the worksheet name.
- 5. Click Create Attachment (). Your MapleSim model opens in Maple, in the template that you select.
- 6. Browse to the **RLC Parallel Circuit 1** subsystem by selecting the subsystem name from the drop-down menu in the toolbar above the model diagram. This menu displays all of the subsystems and components in your MapleSim model.
- 7. In the EMI Block Generation section of the template, click Load Selected Subsystem. All of the template fields are populated with information specific to the subsystem displayed in the model diagram. You can now specify which subsystem parameters will be kept as configurable parameters in the generated block.
- 8. In the **EMI Component Options** section, set the **Code Optimization** option to **Full (3)**. This option specifies the degree of simplification applied to the model equations during the code generation process. This option eliminates redundant variables and equations in the system.
- 9. If you plan to generate a LabVIEW EMI block, follow the steps in the **Generating a LabVIEW EMI Block** section below. If you plan to generate a block for NI VeriStand or the LabVIEW Simulation Interface Toolkit, follow the steps in the **Generating a LabVIEW Block for NI VeriStand or SIT** section below.

Generating a LabVIEW EMI Block

To generate a LabVIEW EMI block

- 1. In the Generate EMI Component Code section of the template, specify the LabVIEW and Visual C++ directories.
- 2. Click Generate and Compile EMI Component to LabVIEW to generate the Visual Studio project and dynamiclink library (.dll) file for the EMI block.
- 3. In LabVIEW, open a new VI and open the block diagram window by selecting Windows>Show Block Diagram.
- 4. Right-click the canvas and select Control Design & Simulation > Simulations > Control and Simulation Loop. Click the canvas and draw a simulation loop box.

- 5. Right-click the simulation loop box and select **Control Design & Simulation>Simulations>Utilities>External Model.** Click a point in the box to position the model.
- 6. In the **Select an External Model Library** window, browse to the **Release** subfolder located in the default directory that you specified in the **LabVIEW EMI Block Generation** template and open the .dll file that you generated.
- 7. Click **OK**. You can now use the RLC circuit block in a LabVIEW EMI diagram. To view a complete example that describes how to prepare and export a slider-crank model as a LabVIEW EMI block, see *Example: Exporting a Model as a LabVIEW EMI Block (page 10)*.

Note: Generating a block may require a few minutes.

Generating a LabVIEW Block for NI VeriStand or the LabVIEW SIT

To generate a LabVIEW block for NI VeriStand or the LabVIEW SIT

- 1. In the Generate SIT Component Code section of the template, specify the SIT and Visual C++ directories.
- 2. Click Generate and Compile SIT Component to VeriStand to generate the Visual Studio project and dynamiclink library (.dll) file for the SIT block.
- 3. In LabVIEW, open a new VI and open the block diagram window by selecting Windows>Show Block Diagram.
- 4. Right-click the canvas and select Control Design & Simulation > Simulations > Control and Simulation Loop. Click the canvas and draw a simulation loop box.
- 5. Right-click the simulation loop box and select **Control Design & Simulation > Simulations > Utilities > External Model.** Click a point in the box to position the model.
- 6. In the **Select an External Model Library** window, browse to the **Release** subfolder located in the default directory that you specified in the **LabVIEW EMI Block Generation** template and open the .dll file that you generated.
- 7. Click OK.

Note: Generating a block may require a few minutes.

For more information about preparing your block for either the NI VeriStand or LabVIEW SIT environment, see *Working with Your Block in NI VeriStand or LabVIEW SIT (page 17).*

2 Example: Exporting a Model as a LabVIEW EMI Block

2.1 Preparing a Model for Export

In this example, you will perform the steps required to prepare a slider-crank mechanism model and export it as a LabVIEW EMI block.

- 1. Convert the slider-crank mechanism model to a subsystem.
- 2. Define subsystem inputs and outputs.
- 3. Define and assign subsystem parameters.
- 4. Export the model using the LabVIEW EMI Block Generation template.
- 5. Implement the EMI block in LabVIEW.

To open the slider-crank mechanism example:

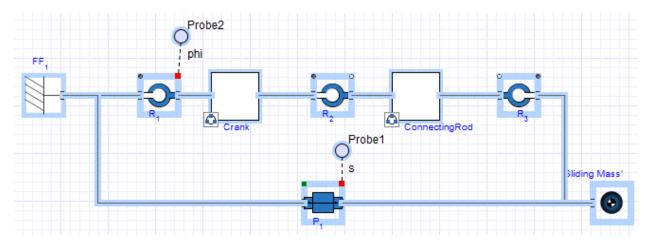
- 1. In MapleSim, click the Help menu item.
- 2. Select Examples > User's Guide Examples > Chapter 6, and then select Planar Slider-Crank Mechanism.

Converting the Model to a Subsystem

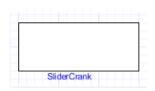
By converting your entire model or part of your model into a subsystem, you identify which parts of the model that you want to export. In this example, you will group all of the components into a subsystem.

To convert the model to a subsystem:

1. Draw a box around all of the components in the model by dragging your mouse over them.



- 2. From the Edit menu, select Create Subsystem.
- 3. In the Create Subsystem dialog box, enter SliderCrank as the subsystem name.
- 4. Click OK. A SliderCrank subsystem block appears in the Model Workspace.



Defining Subsystem Inputs and Outputs

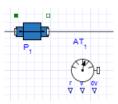
MapleSim uses a topological representation to connect interrelated components without having to consider how signals flow between them, whereas traditional signal-flow modeling tools require explicitly defined system inputs and outputs. Since LabVIEW only supports data signals, properties on acausal ports, such as mechanical flanges and electrical pins, must be converted to signals using the appropriate components. The resulting signals are directed as inputs and outputs for the subsystem in MapleSim and for the EMI block.

Note: Currently, code generation is limited to subsystems with defined signal input (*RealInput*) and signal output (*RealOutput*) ports.

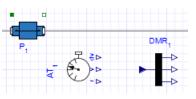
In this example, you will convert the displacements of the slider and the joint between the crank and connecting rod to output signals. The input signal needs to be converted to a torque that is applied to the revolute joint that represents the crank shaft.

To define subsystem inputs and outputs:

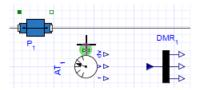
- 1. Double-click the subsystem block to view its contents. The broken line surrounding the components indicates the subsystem boundary, which can be resized by clicking and dragging its sizing handles.
- 2. Delete the probes that are attached to the model.
- 3. In the Library Components tab (() on the left side of the MapleSim window, expand the Multibody palette and then expand the Sensors submenu.
- 4. Drag the **Absolute Translation** component to the **Model Workspace** and place it below the **Prismatic Joint** component.



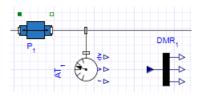
- 5. Right-click (Control-click for Mac®) the Absolute Translation component and select Rotate Counterclockwise.
- 6. From the Signal Blocks > Routing > Demultiplexers menu, drag a Real Demultiplexer component to the Model Workspace and place it to the right of the Absolute Translation component.



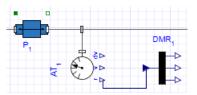
7. To connect the **Absolute Translation** component to the model, click the frame_b connector. The frame is highlighted in green when you hover your pointer over it.



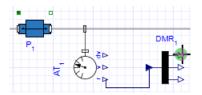
8. Draw a vertical line and click the connection line directly above the component. The sensor is connected to the rest of the diagram.



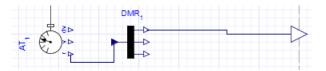
9. In the same way, connect the **r** output port (*TMOutputP*) of the **Absolute Translation** component to the input port of the demultiplexer. This is the displacement signal from the sensor in x, y, and z coordinates. Since the slider only moves along the x axis, the first coordinate must be an output signal.



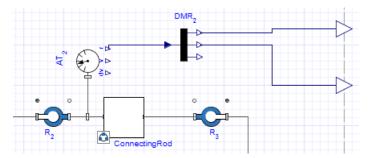
10. Hover your pointer over the first demultiplexer port and click your mouse button once.



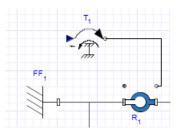
11. Drag your pointer to the subsystem boundary and then click the boundary once. A real output port is added to your subsystem.



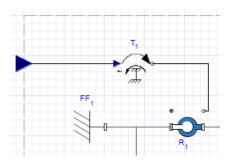
- 12. Add another Absolute Translation component above the Connecting Rod subsystem.
- 13. Right-click (Control-click for Mac) the Absolute Translation component and select Flip Vertical. Right-click the Absolute Translation component again and select Rotate Clockwise.
- 14. Add a **Real Demultiplexer** component to the right of the sensor and connect the components as shown below. Since the crank is moving in the x, y plane, you only need to output the first two signals. You are now ready to add a real input port to your subsystem to control the torque on the crank shaft.



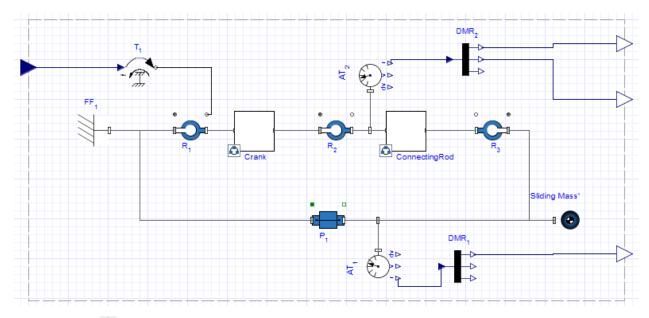
- 15. From the 1-D Mechanical > Rotational > Torque Drivers menu, add a Torque component to the Model Workspace and place it above the Fixed Frame component.
- 16. Connect the white flange of the Torque component to the white flange of the leftmost Revolute Joint.



17. Click the input port of the **Torque** component, then drag your pointer to the subsystem boundary and click the boundary once. A real input port is added to your subsystem.

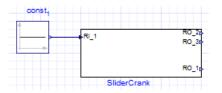


The complete subsystem appears below.



18. Click Main () in the Model Workspace toolbar to browse to the top level of the model.

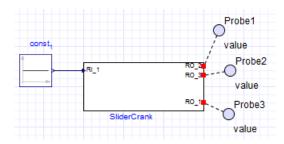
19. From the Signal Blocks > Sources > Real menu, drag a Constant source into the Model Workspace and connect its output port to the input port of the SliderCrank subsystem as shown below.



20. Click Attach Probe () above the Model Workspace toolbar and then click the top output port of the SliderCrank subsystem.

21. In the Model Workspace, click the probe once to position it.

22. In the same way, add probes to the other SliderCrank output ports as shown below.



2.2 Defining and Assigning Subsystem Parameters

You can define custom parameters that can be used in expressions in your model to edit values more easily. To do so, you define a parameter with a numeric value in the parameter editor. You can then assign that parameter as a variable to the parameters of other components; those individual components will then inherit the numeric value of the parameter defined in the parameter editor. By using this approach, you only need to change the value in the parameter editor to change the parameter values for multiple components.

To edit parameters

- 1. Double-click the **SliderCrank** component on the Model Workspace to see the detailed view of the **SliderCrank** subsystem, and then click **Parameters** (**SII**) in the **Model Workspace** toolbar. The parameter editor appears.
- 2. In the first Name field, type CrankL and press Enter.
- 3. Specify a default value of 1 and enter Crank length as the description.
- 4. In the second row of the table, define a parameter called ConRodL and press Enter.
- 5. Specify a default value of 2 and enter Connecting Rod Length as the description.

Standalone Subsystem default settings					
Name	Type	Default Value	Default Units	Description	
CrankL	Real	• 1		Crank length 🛛 😵	
≡ ConRodL	Real	• 2		Connecting Rod Length 🛛 😣	

- 6. Click **Diagram View** (T) to switch to the diagram view, and then click **Main** (1).
- 7. Select the SliderCrank subsystem. The parameters are defined in the Properties tab ().

Properti	es	
Name	SliderCrank	
Type	Standalone Subsystem	1
Paramet	ers	-
CrankL	1	ير 🖸
ConRod	L 2	*

8. Double-click the SliderCrank subsystem, and then select the Crank subsystem.

9. In the **Properties** tab (), change the length value (L) to **CrankL**. The **Crank** subsystem now inherits the numeric value of **CrankL** that you defined.

Proper	ties	
		B
Name	Crank	
Type	Link	1
V Param	eters	-
L	CrankL	94
		5

10. Select the ConnectingRod subsystem and change its length value to ConRodL.

Proper	ties	
Name	ConnectingRod	
Type	Link	بر
▼ Parame	eters	<u> </u>
L	ConRodL	ير 🖸
		S.

11. Click **Main** (a) in the **Model Workspace** toolbar to navigate to the top level of the model. You will include these parameter values in the model that you export. You are now ready to convert your model to an EMI block.

2.3 Exporting Your Model Using the LabVIEW EMI Block Generation Template

After preparing the model, you can use the LabVIEW EMI Block Generation template to set export options and convert the model to a LabVIEW EMI block.

To generate an EMI file:

- 1. Select the Add Apps or Templates tab (👬).
- 2. Double-click on the NI LabVIEW EMI Component Block Generation entry in the Templates palette.
- 3. Enter Slider Crank EMI as the worksheet name, and click Create Attachment (✓). The slider-crank subsystem is opened in the LabVIEW EMI Block Generation Template in Maple.
- 4. Use the navigation controls above the model diagram to select the **SliderCrank** subsystem and then click **Load Selected System**. All of the template fields are populated with information specific to the subsystem.
- 5. Click Generate to LabVIEW to generate the block.
- 6. Set the LabVIEW and Visual C++ directory paths.
- 7. At the bottom of the template, click **Generate and Compile EMI Component to LabVIEW** to generate the Visual Studio project and dynamic-link library (.dll) file for the EMI block.
- 8. In LabVIEW, open a new VI and open the block diagram window by selecting Windows > Show Block Diagram.
- 9. Right-click the drawing canvas and select **Control Design & Simulation** > **Simulation** > **Control and Simulation Loop**. Click the canvas and draw a simulation loop box.

- 10. Right-click the simulation loop box and select **Control Design & Simulation** > **Simulation** > **Utilities** > **External Model.** Click the simulation loop box to position the model.
- 11. In the **Select an External Model Library** window, browse to the **Release** subfolder located in the default directory that you specified in the **LabVIEW EMI Block Generation** template and open the .dll file that you generated.
- 12. Click OK.
- 13. Connect the output of the block to a scope and the input to a sine wave.

3 Working with Your Block in NI VeriStand or LabVIEW SIT

Using the RLC circuit block that you generated in *Example: RLC Circuit Model (page 8)*, this chapter describes how to work with your block in NI VeriStand or the LabVIEW SIT environment.

- Preparing Your MapleSim Model to Run in NI VeriStand (page 17)
- Importing a MapleSim Model to the LabVIEW SIT Environment (page 25)

3.1 Preparing Your MapleSim Model to Run in NI VeriStand

Creating a New Project File

- 1. Open NI VeriStand.
- 2. From the File menu, select New Project.
- 3. In the New Project Name field, enter RLC Circuit.
- 4. Click OK.
- 5 In the Create New Project window, select the System Definition tab.
- 6. Select the Create a new NI VeriStand System Definition file radio button.

😻 Create	New Project				X
Project	System Defintion	Properties			
() Creat	e a new NI VeriStand S	System Defini	tion File		
-	se project name				
Syste	em Definition File name				
RLC	Circuit				
O Use a	n existing NI VeriStand	l System Defi	nition Fil	2	
-	m Definition File Path	-,		-	
			_		
		OK		Cancel	Help

7. Click **OK**. A new project file is created, along with a system definition file.

Adding the MapleSim Model to the System Definition File

1. From Start > All Programs > National Instruments > NI VeriStand 2010, select System Explorer.

2. In the System Explorer window select File, then Open,

3. Navigate to the project you created in the previous section, *Preparing Your MapleSim Model to Run in NI VeriStand (page 17)*.

4. Open RLC Circuit.nivssdf.

Select a system	n definition to o	pen				? 🔀
Look <u>i</u> n:	RLC Circuit		*	3 🦻	جي 🥙	
My Recent Documents	Logs					
Desktop						
My Documents						
My Computer						
	File <u>n</u> ame:	RLC Circuit.nivssdf			~	ОК
My Network	Files of type:	System Definitions (*.nivssdf)			*	Cancel

5. In the left pane, expand Controller, click Simulation Models then click Models.

🧏 System Explorer - RLC Circuit.nivssdf	
<u>E</u> ile Edit <u>T</u> ools <u>H</u> elp	
🏂 🗃 🖬 X 🖣 🐧 X 🛤 🗃 🛕	
Custon Devices Custon Devices Custon Devices Simulation Models Custon Devices Simulation Models Custon Order Colored Control Standus Aims Procedures Aims Aims Calculated Channels Custon Devices System Chanels Calculated Channels Calculated Channels Aims Procedures Aims Aims System Initialization	System Explorer Window We this window to create and/or modify a system definition file. You configure a system definition file by adding, running, and modifying options in the configuration tree, located on the left of the System Explorer window, run can define various components of the system. Including alarms, calculated channels, hardware I/O, system mappings, procedures, simulation models, system channels, user channels, and some VeriStand Engine execution settings. For more information about using the System Explorer window, refer to the Configuring and Running a Project book for detailed descriptions of system definition files and the VeriStand Help. Refer to the Components of a Project book for detailed descriptions of system definition files and the VeriStand Engine. System Definition File Settings Name RLC Circuit Creator Creator Description Window Window Mark Name RLC Circuit Corcuit Version Maine Namine RLC Provision Revision history Image: Revision Revision Revision Revision Revision Revision

6. Click the Add a Simulation Model button located above the right pane. The Add Simulation Model window appears.

😼 Add Sin	nulation N	lodel		X
General	Settings	Parameters and Signals	Inports and Outports	
Name	·			
RLCF	ParallelCircuit	.1		
Path				
C:\D	ocuments an ParallelCircuit	d Settings\maplesoft\Deskt :1.dll	op \akorobkine \RLC \Release \	
NIVer 6012		r Port (MDLs only)		
Simula	ation model ir	nfo		
Path RLCP Modi File s File v Prod Inter Com Lega	arallelCircuit	nts and Settings\maplesoft\ 1.dll /time: 9:14:31.000 AM 17/0	Desktop \akorobkine \RLC \Rel	ease\
			OK Can	cel Help

7. In the General tab, click Browse () and open the .dll that you created in *Example: RLC Circuit Model (page 8)*.

- 8. In the Parameters and Signals tab, select Import all Signals.
- 9. In the Inports and Outports tab, select Segment into scalar channels.
- 10. Click OK.
- 11. Save your changes.

Running the Project

1. In the NI VeriStand Getting Started window, click Run Project.



- 2. In the blank workspace that you opened, from the Screen menu, select Edit Mode.
- 3. Click the Workspace Controller tab on the left side of the workspace.
- 4. In the Workspace Controls menu, expand Model.



5. From the **Workspace Controls** pane, drag the **Model Control** label into the workspace to add a model control component.

M	Worksp	oace - RLC Circuit		
Eile	Tools	<u>S</u> creen <u>V</u> iew <u>W</u> indow <u>H</u> elp		
Œ		Screen	M 1 of 1	🔛 💽
S				
Workspace Controls		+ Controller/RLCParallelCircuit1		
pace	#	12.2		
Vorks		Model running		
F	ΗĽ.			

6. From the Item Properties window, select RLC.

😻 Item Prop	erties	
Model:	Controller/RLCParallelCircuit1	~
Delete	ОК	Cancel

7. Click OK.

Adding a Dial to the Workspace

- 1. Click the Workspace Controller tab.
- 2. In the Workspace Controls menu, expand Numeric Control.



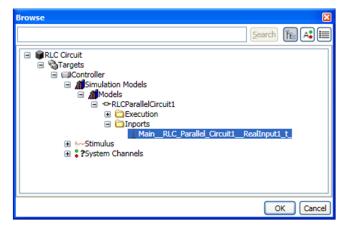
3. Drag the **Dial** component into the workspace.

₩ Workspace - RLC_Tutorial	
Eile <u>T</u> ools <u>S</u> creen <u>V</u> iew <u>W</u> indow <u>H</u> elp	
Displayed Screen: 1 of 1	
Empty Screen	
8	
RLC +	
RLC + RLC 35.1 Model running	
Model running	
S Proder romming	
40 50 60	
30,70	
1090	
0 100	
100	
	-

The Item Properties window appears.

Item Propertie	s	
General Format	& Precision Limits & Warnings	
Channel:		
	ion Models/Models/RLC/Inports/)
Control Label:		
MainRLC_Parall	el_Circuit1RealInput1_t_	
Default Units:	New Units:	
	Use Default Units?	
Scale Factor:	Offset:	
1	0	
Delete	OK Car	ncel

- 4. In the Item Properties window, click Channel (
- 5. Expand Controller>Simulation Models>Models>RLCParallelCircuit1>Inports.
- 6. Select Main_RLC_Parallel_Circuit1__RealInput1_t_



7. Click OK.

8. Click OK.

Adding a Graph to the Workspace

1. Click the Workspace Controller tab.

2. In the Workspace Controls menu, expand Graph and drag the Simple label into the workspace.

M Workspace Controls			
÷	Auto		
÷	Boolean Control		
÷	Boolean Indicator		
÷	Decoration		
.	Graph		
-	FFT		
	Simple		
.	Model		
	Model Control		
÷	Model Calibration Control		
0	Numeric Control		
	Dial		
	Medium		
÷	Numeric Indicator		

3. In the Graph Channel Selection window, in the left pane, expand Controller > Simulation Models > Models > RLCParallelCircuit1 > Outports.

W Graph Channel Selection		
General Format & Precision		
Graph Title: Graph V Allow Resize?		History Length: Update Rate: 10.0 5 Hz
Search RLC Circuit RLC Circuit Controller Controlle	Legend Text Variab Main_RLC_Parallel_Circuit1_R01 Target	ole ts/Controller/Simulation Models/Models/RLCParall
Delete		OK Cancel

4. Select Main_RLC_Parallel_Circuit1__RO1_t_

- 5. Click the right-pointing arrow () to include the outport quantity in the graph.
- 6. Click OK.
- 7. From the Screen menu, clear the Edit Mode option.

8. In the workspace, rotate the dial to change the input behavior. The results are shown in the graph.

1 Workspace - RLC Circuit		
<u>Elle T</u> ools <u>S</u> creen <u>V</u> iew <u>W</u> indow <u>H</u> elp		
Screen	✓ 1 of 1	🔛 💽
Controler, RLCParaleCrout 1 Nodel running	Graph Gr	Setup Hide Legend Hold Autoscale X On Once Deta Lost

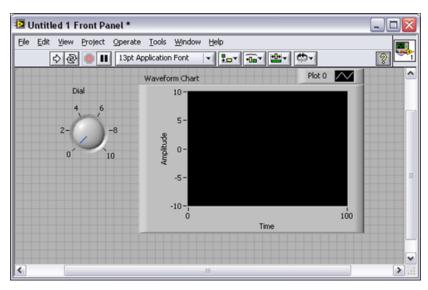
3.2 Importing a MapleSim Model to the LabVIEW SIT Environment

Creating a LabVIEW SIT Interface

- 1. Open a new VI file.
- 2. In the Front Panel, right-click to open the Numeric Controls panel.



- 3. Select Numeric Controls and then select Dial.
- 4. Drag the **Dial** component into the Front Panel
- 5. Right-click the Front Panel
- 6. Select Graph Indicators
- 7. Select Chart.
- 8. Drag the chart component into the Front Panel

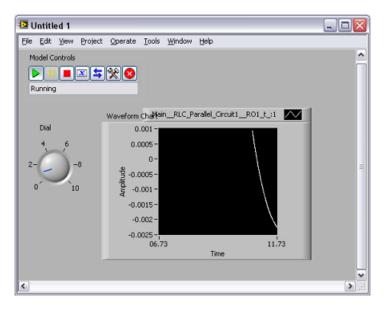


Connecting the MapleSim Model and the LabVIEW SIT User Interface

- 1. From the Tools menu, select SIT Connection Manager.
- 2. In the SIT Connection Manager window, select Driver VI on Localhost.
- 3. In the Current Model DLL section, browse to and select the .dll that you created.

Category Model and Host Mappings	Execution Host		
File Playback File Playback Hardware I/O	Simulation Environment Real-Time Target Oriver VI on Localhost	Target Localhost Port 6011 (2)	ि Change
	Current Model DLL C:¥Documents and Settings¥wwo	ng¥Desktop¥Release¥RLC.dll	
	Project Directory C:WDocuments and Settings\#www	ngVDesktopVRelease	
			OK Cancel Heb

- 4. In the left pane of the SIT Connection Manager, select Mappings.
- 5. In the Current Mappings table, double-click the first row which corresponds to the dial that you inserted.
- 6. Expand rlcparallelcircuit1 > Input_param.
- 7. Select Main_RLC_Parallel_Circuit1__RealInput1_t_.
- 8. Click OK.
- 9. Below Current Mappings, double-click Waveform Chart.
- 10. Expand rlcparallelcircuit1 > Output > Main_RLC_Parallel_Circuit1_RO1_t_.
- 11. Select Port 1 Main_RLC_Parallel_Circuit1_RO1_t_.
- 12. Click OK. The SIT Connection Manager will now build the model.
- 13. Click **Run** (**>**) to run the simulation. When the simulation is complete, you can rotate the knob to change the output.



4 Running a Simulation on a LabVIEW Real-Time Target Machine

You can run a simulation on a LabVIEW real-time target machine by using any .dll file that you generated using the MapleSim Connector for LabVIEW and NI VeriStand Software. In this chapter, the steps for running a real-time simulation are demonstrated using the slider-crank .dll file that you generated in *Example: Exporting a Model as a LabVIEW EMI Block (page 10)* in Chapter 2 of this guide. These steps can also be applied to any .dll file for which you want to run a real-time simulation.

4.1 Preparing the LabVIEW Real-Time Project

To prepare the LabVIEW Real-Time project

- 1. From the LabVIEW Getting Started window, click Real-Time Project.
- 2. Keep the project type as **Continuous communication architecture**, change the project name to **RTSliderCrank**, and click **Next**.

Select project type, name, and folder	
Project type	
Continuous communication architecture	
Application includes deterministic components	
Checking this box will restrict options on the next page to ensure the project can run a deterministic loop	
Project name RTSliderCrank	
Project folder	
C:\Documents and Settings\wwong\My Documents\ LabVIEW Data\RTSliderCrank	

3. Keep all of the default architecture options values and click Next.

Create new LabVIEW Real-Time project	E
Customize architecture options Target Configuration One loop Target VI runs one timed loop Two loops Target VI runs two timed loops at different priorities to separate deterministic and non-deterministic tasks	
Include file I/O Host Configuration	
 Host VI Use a VI on the host to communicate with target Remote Panel Use a web browser to communicate with target 	
< Back Next >	Finish Cancel Help

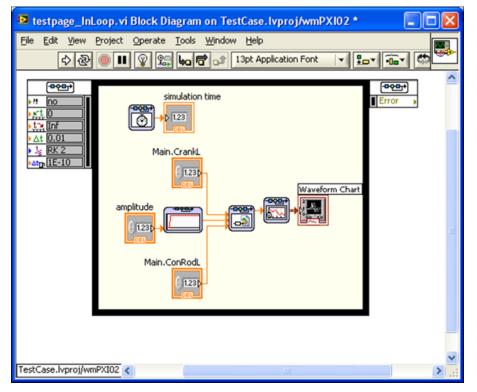
4. Click Browse... From the drop-down menu, browse to locate the real-time target platform. Click Next.

Create new LabVIEW Real-Time project	
Browse targets Add a networked target, such as PXI, Compact FieldPoint, FieldPoint, CompactR10, Single-Board R10, Compact Vision System Browse Selected target wmPXI02	
< Back Next >	Finish Cancel Help

- 5. Click **Finish** to create and display the model.
- 6. From the **Project Explorer**, right-click the entry of the target platform and select **Add >File..**..

Project Explorer - RT	SliderCrank.lvproj	
Elle Edit View Project Q	perate <u>T</u> ools <u>W</u> indow <u>H</u> elp	
🍢 🗃 🖬 🞒 🗴 🖣) 🗅 🗙 🍤 🝽 🛛 💕 🔩	: 🛅 = 📽 🛕]
Items Files		
By Project: RTSliderCra My Computer P Dependencies		
Build Specifica		1
- 🛋 target - sing - 🛋 support - ac	New >	
- 🍟 Dependencia	Add	File
	Connect Disconnect Utilities	Folder (Snapshot) Folder (Auto-populating) Hyperlink
	Deploy Deploy All	
	Arrange by Expand All Collapse All	
	Remove from Project Rename F2	
	Help Properties	

- 7. Browse to the **Release** subfolder located in the default directory that you specified in the **LabVIEW EMI Block Generation** template and open the .dll file that you generated. Click **OK**.
- 8. Navigate to the block diagram of the VI. Double-click the **Simulation Parameters** window to the left of the simulation loop. The **Configuration Simulation Parameters** window appears.



9. Click the Time Parameters tab and select Synchronize loop to time source. Click OK.

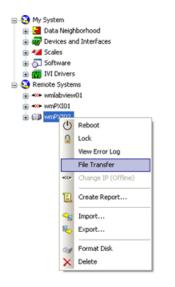
Configure Simulation F	Parameters 🛛 🔀
Simulation Parameters Timin	ng Parameters
- Enable Synchronized T	iming
Synchronize Loop to T	-
Synchronize coop to 1	ining source
ze the simulation loop to a timing	source
Source type	
1 kHz Clock	~
1 MHz Clock	
1 kHz <reset at="" structure<br="">1 MHz <reset at="" structure<="" td=""><td></td></reset></reset>	
Synchronize to Scan Engi	
Other <defined by="" source<="" td=""><td>e name or terminal></td></defined>	e name or terminal>
Source name	
1 kHz	
Calculated Period	
1	Auto Period
Offset / Phase	Priority
0	100 😂
Deadline	Timeout (ms)
-1	-1
-Processor Assignment	
Mode	Processor
Automatic 🔽	-2
OK C	Cancel Help

10. Save the file.

4.2 Moving the .dll File to the Target Real-Time Machine

To move the .dll File to the target real-time machine

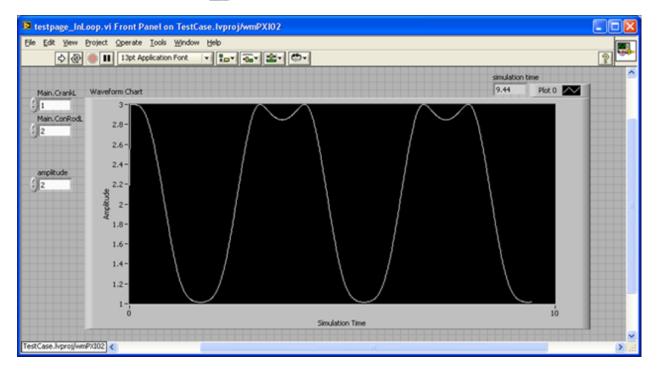
- 1. From the start menu, select Measurement and Automation Explorer.
- 2. In the Measurement and Automation Explorer window, expand Remote Systems.
- 3. Right-click the entry for your target machine.
- 4. Select File Transfer.



- 5. Browse to the directory that contains the .dll file you created.
- 6. Select the .dll file.

Target IP Address 10.10.9.2					
Current Remote Dir	ectory				
1				OK	Delete
Folders	Nan	ne	Size	Date 🔨	Delece
≡n /	6				
\$RECYCLE.	BIN	\$RECYCLE.BIN		Jul 11 C 💷	New Folder
😥 ni-rt	6	ni-rt		Jul 13 1	
主 temp	6	temp		Jul 13 1	Rename File
		hdmon.bin	242732	Jul 20 1	
		LABVIEWRT	0	Jul 10 2 🥃	Select All
<	2 3		1.002	S	
a.		To Remote	To Local	¥	Cancel
Current Local Direc	tory				
Folders	Nan		Size		Delete
			2126	Date 🔨	
≓ C:/		3DOFHEL		05/10/2	New Folder
E- C:\ B- 3DOFHEL	0	3DOFHEL 3DOFHEL EMI		05/10/; 05/10/;	New Folder
≓ C:/	EMI O	3DOFHEL_EMI	c4970bf35	05/10/2	New Folder
C:\ - 3DOFHEL - 3DOFHEL_I	EMI 13679e384	3DOFHEL_EMI 3ed0092993679e384695	c4970bf35	05/10/; 08/01/;	
C:\ - 3DOFHEL - 3DOFHEL_I - 3ed009299	EMI 13679e384	3DOFHEL_EMI 3ed0092993679e384695 Config.Msi	c4970bf35	05/10/2 08/01/2 16/10/2	Rename File
C:\ OOFHEL OOFHEL_ OOFHEL	EMI 13679e384	3DOFHEL_EMI 3ed0092993679e384695	c4970bf35	05/10/; 08/01/;	

- 7. Click **To Remote** to move the .dll file from your local machine to the target machine in the **ni-rt/system** directory.
- 8. Click **Close** and then click **Run** (**>**) in the front panel of the VI. The following graph appears.



Index

D

DLL file generating, 8, 9 moving to target machine, 30

Ε

EMI Component Options, 3 Baumgarte Constraint Stabilization, 4 Constraint Handling Options, 3 Discretization, 4 Event Handling Options, 3 Optimization Options, 3 Examples RLC circuit model, 8, 17 slider-crank model, 10 Exporting, iv External Libraries, iv

G

Generate EMI Component Code, 7 External Libraries, 7 SIT Component Code, 7

L

LabVIEW EMI block exporting, 10 generating, 8 LabVIEW SIT block, 25 generating, 9

Μ

Models using external libraries, iv

Ν

NI Connector Examples, 7 NI VeriStand, 17 NI VeriStand model generating, 9

Ρ

Port and Parameter Management, 1

R

Real-time simulations, 27

S

SIT Component Options, 4

Subsystem Preparation, 1 Selection, 1 Subsystem parameters, 14

Т

Templates, 1 NI LabVIEW EMI Block Generation, 7, 15 NI VeriStand and LabVIEW SIT Model Generation, 7

V

View EMI or SIT Component Code, 7