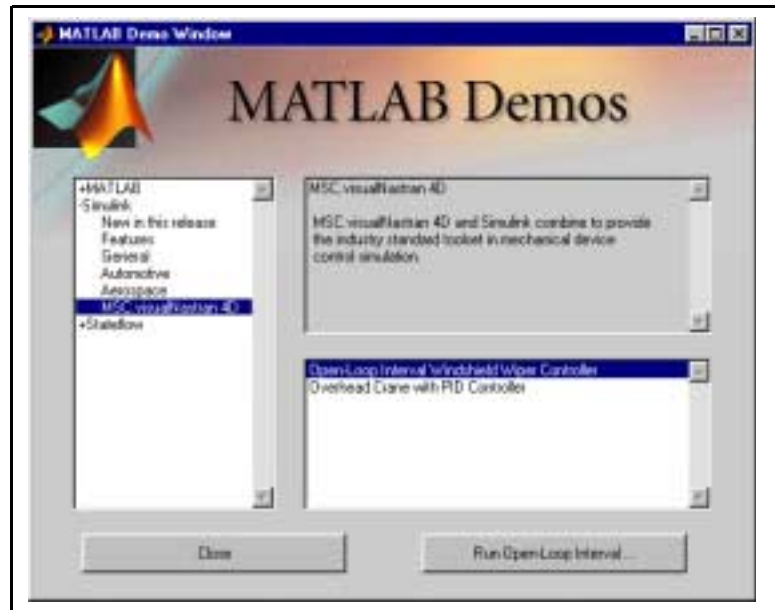


Simulink® Integration



Exercise 9-1

Open-Loop Controller for Interval Windshield Wipers

In this exercise, you will design a control system in Simulink to control an automotive windshield-wiper model. The physical model of the windshield wiper was constructed in a CAD package and imported into MSC.visualNastran 4D.

Objectives

In this exercise, you will learn to

- Connect MSC.visualNastran 4D (vN4D) with Simulink.
- Access the vNPlant block in Simulink.
- Pass variables from vN4D to Simulink.
- Pass variables from Simulink to vN4D.
- Run the integrated simulation.

Support Files

- Tutorials\Chapter 09\Exercise 9.1\wipers.wm3
- Matlab\vNPlantBlock.mdl

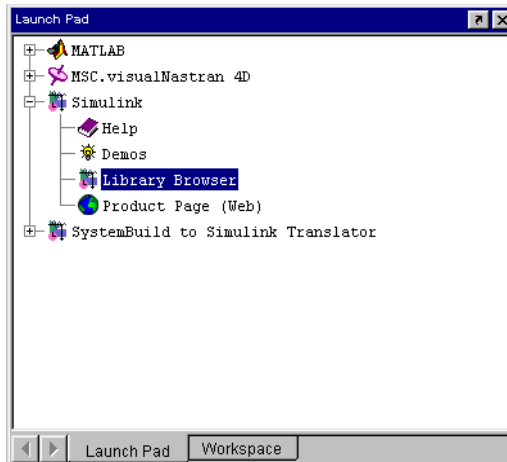
Launch Applications and Open Files

NOTE: Instructions for this section are divided into two subsections: **Instructions for Simulink 4 and MATLAB 6** and **Instructions for Simulink 3 and MATLAB 5.3**. Please choose the appropriate subsection based on your software version.

Instructions for Simulink 4 and MATLAB 6

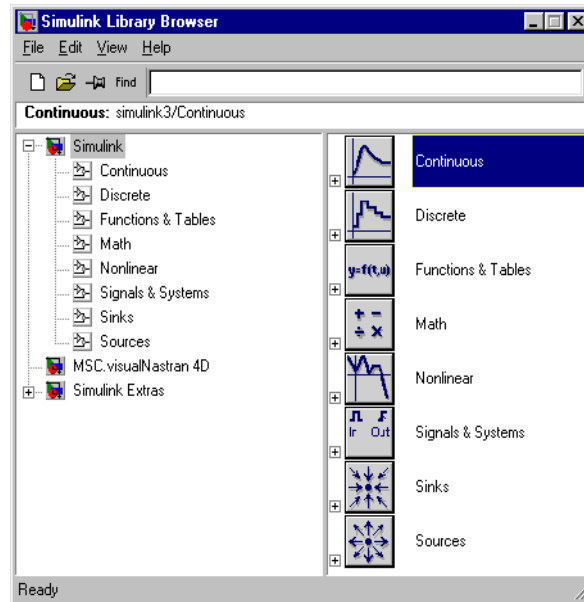
1. Launch MATLAB.
2. In the MATLAB **Launch Pad**, double-click **Simulink** to view its menu.

Figure 9-1
MATLAB's Launch Pad



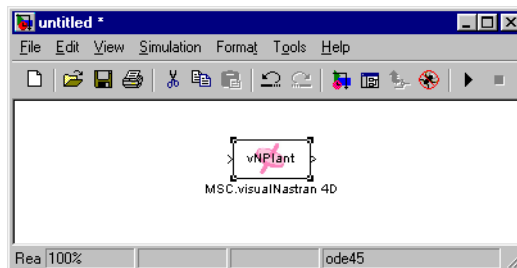
3. Double-click **Library Browser**.

Figure 9-2
Simulink Library Browser



4. In the **Simulink Library Browser** window, click **MSC.visualNastran 4D**. The **vNPlant** block appears in the right column.
5. From the **File** menu of the **Library Browser**, select **New**, then **Model**. This file will consist of the control system for the integrated simulation.
6. Drag and drop the **vNPlant** block from the right column of the **Simulink Library Browser** to the new model window.

Figure 9-3
vNPlant Block

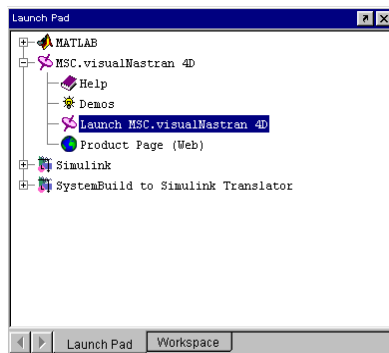


This block is the starting point for all vN4D and Simulink integration. Arrows (\triangleright) exist on both the left and right sides of the **vNPlant** block. The \triangleright on the left indicates the flow of information from Simulink to vN4D. The \triangleright on the right indicates the flow of information from vN4D to Simulink.

 The model file for the vNPlant block (**vNPlantBlock.mdl**) is located in the **Matlab folder of your installation directory**.

7. Save your Simulink model as **wipers.mdl**.
8. In the MATLAB **Launch Pad**, double-click **MSC.visualNastran 4D** to view its menu.
9. Double-click **Launch MSC.visualNastran 4D**.

Figure 9-4
MATLAB's Launch Pad



10. In vN4D, choose **Open** from the **File** menu.
11. Browse the **Tutorials\Chapter 9\Exercise 9.1** folder and open the file **wipers.wm3**.

Instructions for Simulink 3 and MATLAB 5.3

1. Launch MATLAB.
2. Launch the **Simulink Library Browser** from the toolbar or type `simulink` at the MATLAB command line.
3. In the **Library Browser** window, double-click **MSC.visualNastran 4D** to expand its menu. **vNPlant** is listed here.
4. Create a new Simulink model window by clicking the new model icon (blank white page) in the tool bar of the **Library Browser**. You will build the control system for the integrated simulation in this window.
5. Drag and drop **vNPlant** from the **Library Browser** to the new model window.

This block is the starting point for all vN4D and Simulink integration. Arrows (>) exist on both the left and right sides of the **vNPlant** block. The > on the left indicates the flow of information from Simulink to vN4D. The > on the right indicates the flow of information from vN4D to Simulink.

 The model file for the vNPlant block (**vNPlantBlock.mdl**) is located in the **Matlab folder of your installation directory**.

6. Save your Simulink model as **wipers.mdl**.
7. Launch vN4D from the MATLAB command line by typing **vN4D**, or separately from your Windows environment.
8. In vN4D, choose **Open** from the **File** menu.
9. Browse the **Tutorials\Chapter 9\Exercise 9.1** folder and open the file **wipers.wm3**.

vN4D and Simulink Integration

Your Simulink model will communicate with your vN4D model through the vNPlant block. For this to occur, you must first define in your vN4D model import and export parameters. These parameters are then made available to your vNPlant block in Simulink for you to select as appropriate.

Define import and export parameters in the vN4D model

1. Switch to the vN4D window to view **wipers.wm3**.

Figure 9-5
Wipers model

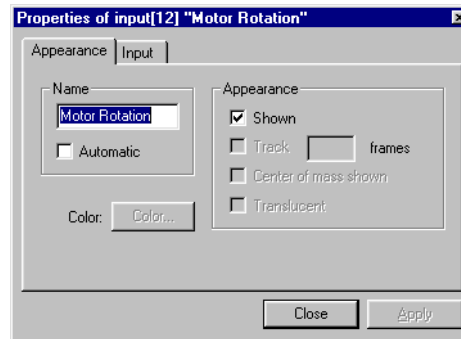


Simulink acts upon the values of Controls in vN4D. All information sent from Simulink to vN4D must be in the form of a Control. You will create a Control for the wipers model:

2. In the **Object List**, select the motor (**constraint 131**).

3. From the **Insert** menu, select **Control**, then **Rotation**. Choose the **Slider** format.
4. Click **OK**.
5. Double-click the input slider to display its **Properties** window. On the **Appearance** page, change the name to **Motor Rotation**.

Figure 9-6
*Properties of Input Slider for
Motor Rotation*

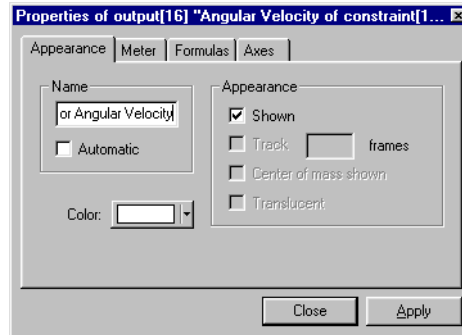


6. Close the **Properties** window.

Simulink reads Meter values from vN4D. All information sent from vN4D to Simulink must be measured by a Meter.

7. In the **Object List**, select the motor (**constraint 131**) once again.
8. From the **Insert** menu, select **Meter**, then **Angular Velocity**. Select **Tile vertically**.
9. In the **Connections List** of the **Object Manager**, right-click the meter you just created to view its **Properties** window. On the **Appearance** page, change the name to **Motor Angular Velocity**.

Figure 9-7
*Properties of Meter for Motor
 Angular Velocity*



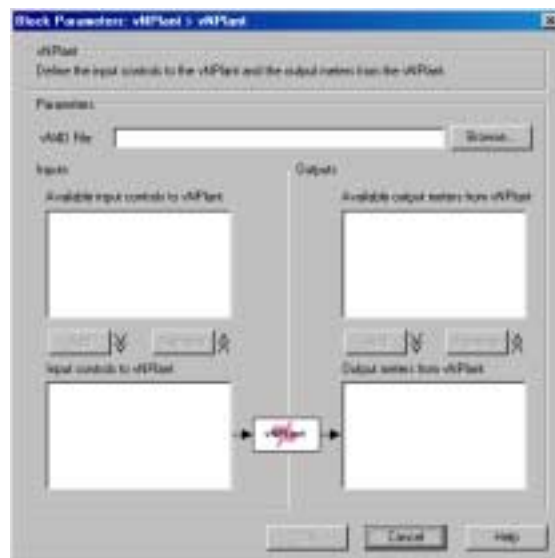
10. Close the **Properties** window.

Connect inputs and outputs to Simulink through vNPlant.

1. Open (double-click) the **vNPlant** block in the **wipers.mdl** model you created above.

The **Block Parameters** dialog opens. Using this dialog, the user will select the vN4D model with which Simulink will communicate, the input variables to vN4D, and the output variables from vN4D.

Figure 9-8
Block Parameters

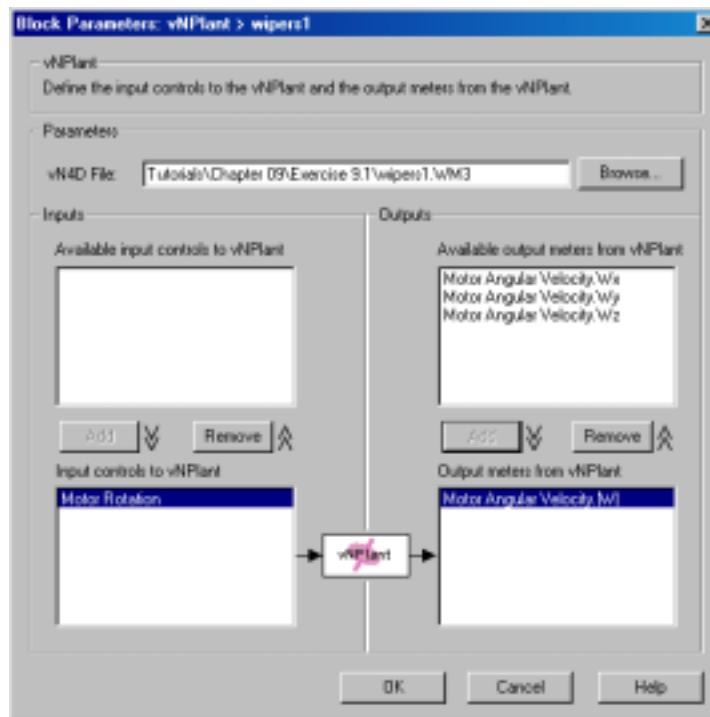


2. Use the **Browse...** button to navigate to the wipers model **Tutorials\Chapter 09\Exercise 9.1\wipers.wm3**.


All available **Inputs** and **Outputs** should now be visible. The **Inputs** are parameters that are passed from Simulink to vN4D via the vN4D Controls, and the **Outputs** are parameters that are passed from vN4D to Simulink via vN4D Meters.

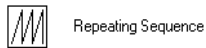
3. Select **Motor Rotation** from the box labeled “Available input controls to vNPlant.” Click **Add** to move it down to the box labeled “Input controls to vNPlant.”
4. Select **Motor Angular Velocity.[W]** (magnitude) from the box labeled “Available output meters from vNPlant.” Click **Add** to move it down to the box labeled “Output meters from vNPlant.”
5. Click **OK**.

Figure 9-9
*Block Parameters Page with
Inputs and Outputs defined*



NOTE: The vNPlant **Block Parameters** page must define at least one input and one output. Although you will not implement a feedback control system for this exercise, you will select **Motor Angular Velocity.[W]** as an output from vNPlant.

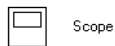
 Single input/output values are used in this example. If multiple values were to be used, then the input and output would be written as a “vector of quantities.” Such use is shown in Exercise 9-2.



Construct the control system

1. From the Simulink **Library Browser** select **Simulink-> Sources (or double-click Sources in Simulink 3)**.
2. Copy (drag and drop) **Repeating Sequence** to the left of **vNPlant** in the **wipers.mdl** Simulink model.

This is a Simulink element that will generate repeating signals. This will form the heart of the open-loop control system.

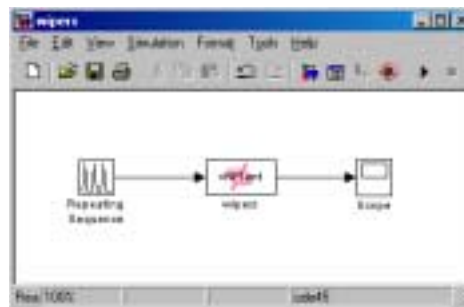


3. From the Simulink **Library Browser** select **Simulink-> Sinks**.
4. Copy (drag and drop) **Scope** to the right of **vNPlant** in the Simulink model.

This is a Simulink element that graphically displays measured quantities, much like a Meter in vN4D. This will be used to view the quantity input (angular velocity) to Simulink from vN4D.

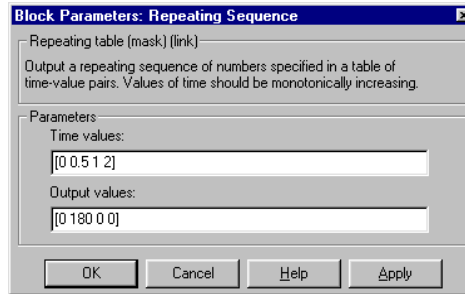
5. Using the mouse, “draw” signal lines connecting the **Repeating Sequence** to **vNPlant** and **vNPlant** to the **Scope**.

Figure 9-10
Control System



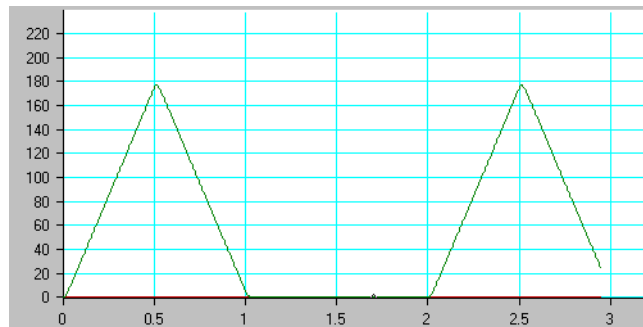
6. Open (double-click) the **Repeating Sequence** block.
7. Enter the sequence [0 0.5 1 2] for the time values.
8. Enter the sequence [0 180 0 0] for the output values.

Figure 9-11
Parameters for the Repeating
Sequence Block



You have defined the time-dependent orientation that will be sent to the wiper motor. The motor will start at zero degrees orientation. At one half second, the orientation will be 180 degrees. At one second, the orientation will again be zero degrees. Thus, the windshield wipers make one pass back-and-forth across the windshield in one second. They then remain at zero degrees (at rest) for another second. The sequence then repeats, as shown in the figure below:

Figure 9-12
Orientation (degrees) versus
Time (seconds)



This is your open-loop interval control system.

9. Click **OK** to close the Block Parameters dialog for the **Repeating Sequence** block.

The control system has been constructed. Although saving your **wipers.mdl** file is not necessary to run the control simulation, this is a good point at which to save the file.

Run the simulation




1. From the Simulink model page, click the **Run** button.

Simulink will control vN4D and run the combined simulation. The simulation will stop automatically after 10 seconds.



2. Click the stop button in the Simulink panel to stop the simulation.

 You can adjust the run time using the **Simulation->Simulation Parameters** drop-down menu from the Simulink model page.

You have now performed an integrated simulation using vN4D for the plant dynamics model and Simulink for the control system simulation.

Further Exploration

- Add a meter to measure the total force on the wiper pivot joints.
- Vary the wiper interval by changing the time values in the repeating sequence block.
- Vary the wiper speed by changing the time values in the repeating sequence block.
- Observe how the change in wiper speed affects the measured force on the wiper pivot joints.
- A real windshield wiper system doesn't command an angular orientation for the motor; instead, it commands a voltage, which is converted to a torque. Try changing the motor from a commanded orientation to a commanded torque and constructing an open-loop control system based on commanded torque. (You may find this exercise challenging.)

Exercise 9-2

Feedback Controller for an Overhead Crane

In this exercise, you will design a control system in Simulink to control an overhead crane model. The crane is supported by a horizontal beam 10m in length. Below it hangs a large hook for carrying loads. A horizontal force (e.g., via a cable) is applied to the crane to move it along the beam. The object of the control system is to move the crane from one point along the

beam to another in a timely fashion without exceeding performance limits. The crane model is free to move along the beam horizontally. The vertical position of the hook and its load will remain fixed in this exercise.

Objectives

In this exercise, you will learn to

- Create a proportional/derivative (PD) feedback system for positional control using a realistic model.
- Use multiple plant outputs (vN4D meters).
- Demonstrate model behavior for underdamped and critically damped systems.

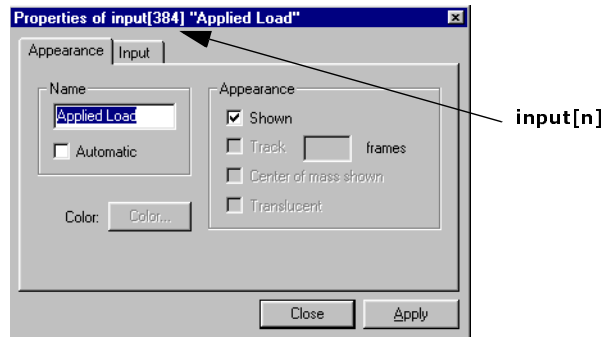
Support Files

- Tutorials\Chapter 09\Exercise 9.2\Crane.WM3
- Matlab\vNPlantBlock.mdl

Start vN4D

1. Launch MSC.visualNastran 4D.
2. Choose **Open** from the **File** menu.
3. Browse the **Tutorials\Chapter 9\Exercise 9.2** folder and open the file **Crane.wm3**.
4. From the **Insert** menu, select **Control**, then **Generic**. Choose **Slider** and click **OK**.
5. Double-click the input slider to display its **Properties** window.
6. On the **Appearance** page, change the name of the input slider for the concentrated force to `Applied Load`.
7. On the **Input** page, specify **N** (for Newtons) as the units and set the minimum and maximum force to `-1000` and `1000`, respectively.

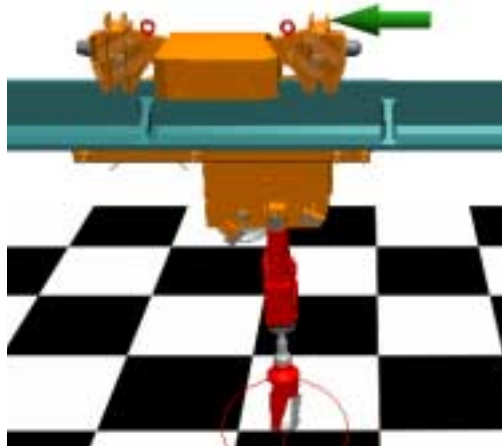
Figure 9-13
Properties of Input Slider



The input slider for the force has a formula language identifier (**input[n]**) listed in the **Properties** window label. You will use this identifier to “link” this input to the x-component of the concentrated force we use to move the crane.

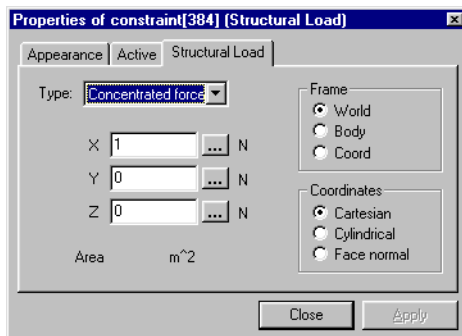
8. Click **Close**.
9. Select the concentrated force (green arrow) in the modeling window.

Figure 9-14
Concentrated Force in Crane model



10. Double-click the force to display its properties.

Figure 9-15
Properties of Concentrated Force



11. On the **Structural Load** page, enter the identifier for the input slider (**input[n]**) in the x-component field. The y- and z- components should be set to zero. The Frame should be set to **World**, and the Coordinates should be set to **Cartesian**. Close the **Properties** window.
12. Select the hook in the modeling window. (See the following tip.)



You may also select the hook from the **Object List**: Click the second tab to view all bodies in the model. Scroll to select **HOOK_1**. OR... Select the Coord attached to the hook in the modeling window. Then, select **HOOK_1** from the **Connections List** of the **Object Manager**.



13. Add a meter for the hook's position: From the **Insert** menu, select **Meter**, then **Position**. **Select Tile Vertically and click OK.**
14. Right-click the meter to view its **Properties** window. On the **Appearance** page, change the meter's name to `Position` of Hook. Close the **Properties** window.
15. Select the hook again and add a meter for the hook's velocity: From the **Insert** menu, select **Meter**, then **Velocity**. **Select Tile Vertically and click OK.**
16. Right-click the meter to view its **Properties** window. On the **Appearance** page, change the meter's name to `Velocity` of Hook. Close the **Properties** window.
17. Save the **crane.wm3** file.

Start Simulink

NOTE: Instructions for this section are divided into two subsection: **Instructions for Simulink 4 and MATLAB 6** and **Instructions for Simulink 3 and MATLAB 5.3**. Please choose the appropriate subsection based on your software version.

Instructions for Simulink 4 and MATLAB 6

1. Launch MATLAB.
2. In the MATLAB **Launch Pad**, **double-click Simulink to view its menu**.
3. Double-click **Library Browser**.
4. In the **Simulink Library Browser** window, click **MSC.visualNastran 4D**. The **vNPlant** block appears in the right column.
5. From the **File** menu of the **Library Browser**, select **New**, then **Model**. This file will consist of the control system for the integrated simulation.
6. Drag and drop the **vNPlant** block from the right column of the **Simulink Library Browser** to the new model window.
7. Save your Simulink model as **crane.mdl**.

Instructions for Simulink 3 and MATLAB 5.3

1. Launch MATLAB.
2. Launch the **Simulink Library Browser** from the toolbar or type `simulink` at the MATLAB command line.
3. In the **Library Browser** window, double-click **MSC.visualNastran 4D** to expand its menu. **vNPlant** is listed here.
4. Create a new Simulink model window by clicking the new model icon (blank white page) in the tool bar of the **Library Browser**. You will build the control system for the integrated simulation in this window.
5. Drag and drop **vNPlant** from the **Library Browser** to the new model window.
6. Save your Simulink model as **crane.mdl**.

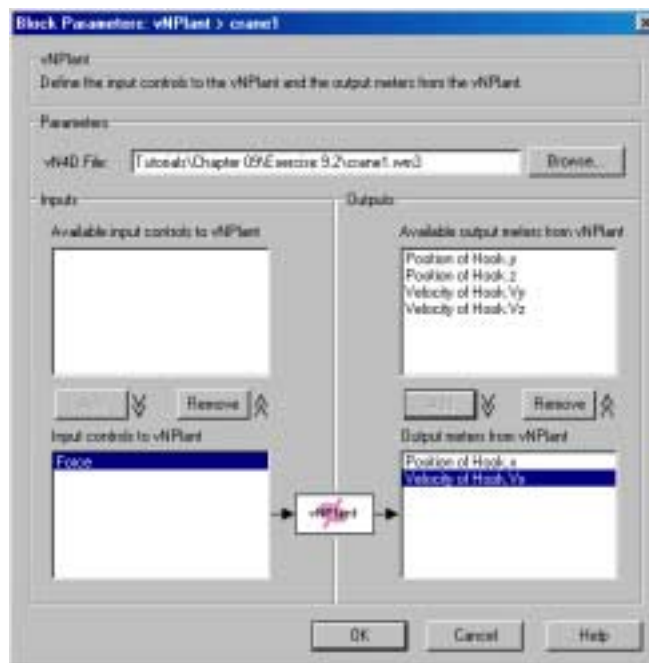
vN4D and Simulink Integration

1. Double-click the **vNPlant** block to display its **Block Parameters** page.
2. Use the **Browse...** button to navigate to the crane model **crane.wm3**.

Define import and export parameters

1. Select **Applied Load** from the box labeled “Available input controls to vNPlant.” Click **Add** to move it down to the box labeled “Input controls to vNPlant.”
2. Select **Position of Hook.x** and **Velocity of Hook.Vx** from the box labeled “Available output meters from vNPlant.” Click **Add** to move them down to the box labeled “Output meters from vNPlant.”

Figure 9-16
Block Parameters for the Crane model



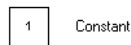
The two output meters are exported as a vector quantity. The vector components are listed in the order shown in the box labeled “Output meters from vNPlant.” In this case, the vector is (**Position of Hook.x, Velocity of Hook.x**).

NOTE: The order is alphabetical based on the input or output names. You can change these names in the controls and meters.

- Click **OK** to close the **Block Parameters** page for **vNPlant**.

Construct the control system

- From the Simulink Library Browser, select **Simulink-> Sources**.
- Copy (drag and drop) **Constant** to the left of **vNPlant** in your Simulink model.



This Simulink element will designate the end position of the crane.

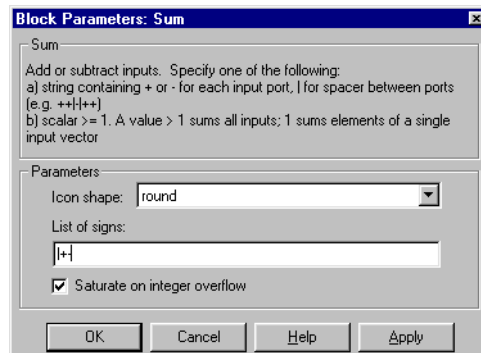
- Double-click the element and enter 8 for the constant value (to represent 8 meters).
- In the **Simulink Library Browser**, select **Simulink-> Math**, and drag **Sum** to the position between **Constant** and **vNPlant**.



The **Sum** tool allows you to create a feedback loop.

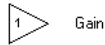
- Double-click the **Sum** icon and enter |+- in the field for **List of Signs**.

Figure 9-17
Block Parameters Dialog for the Sum tool



This will take the difference of two signals.

- Close the **Block Parameters** dialog for the **Sum** tool.



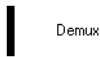
Gain

7. Select **Simulink-> Math-> Gain** and place it to the right of the **Sum** icon.
8. Double-click the **Gain** icon. In the **Block Parameters** dialog, set the **Gain** to **100**.

This is the positional gain. The implied units are (kg/s²) for this exercise.

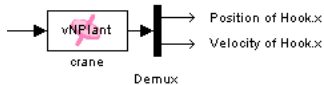
NOTE: The mathematics in Simulink are essentially unitless. All units are defined in the controls and meters in vN4D. vN4D passes or receives a pure number to and from Simulink.

9. Close the Gain dialog.
10. Control-click the **Sum** icon and drag it to the right of the **Gain** icon. This makes a copy of the first **Sum** icon, and the signs are correct for taking the difference of two signals.
11. In the **Simulink Library Browser**, select **Simulink-> Signals & Systems-> Demux** and drag it to the Simulink document. Place the **Demux** to the right of **vNPlant**.



Demux

Demux is the de-multiplexer. It splits a vector signal into scalar signals. You will use it to split the **vNPlant** output into position and velocity information.



NOTE: Once a **Demux** is attached to **vNPlant**, the order of scalar signals output from the **Demux** is given by the parameter order shown in “Output meters from vNPlant” box in the **Block Parameters** dialog for **vNPlant**. In this case, the upper signal is **Position of Hook.x**, and the lower signal is **Velocity of Hook.x**.

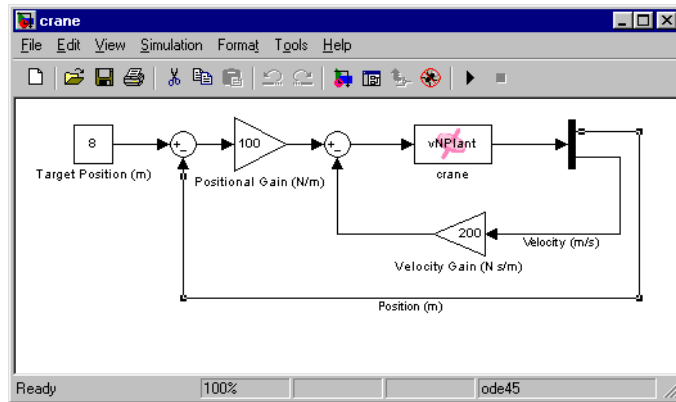


Gain

12. Place a second **Gain** directly below **vNPlant**. This will be the velocity gain.
13. Double click the icon and enter 200 in the **Gain** field of the **Block Parameters** dialog. The implied units are (kg/s).
14. Right-click the velocity **Gain** to display the context menu. Select **Format-> Flip Block** to orient the block leftward. This gain will be part of the feedback loop.

15. Use the mouse to draw signal lines to connect the blocks as shown.

Figure 9-18
Control System for Crane



Run the simulation

1. From the **Simulation** menu, select **Simulation Parameters** and set the **Stop time** to 20 seconds.
2. Run the simulation by clicking the **Run** button in the Simulink window. Go to the vN4D window to view the simulation.

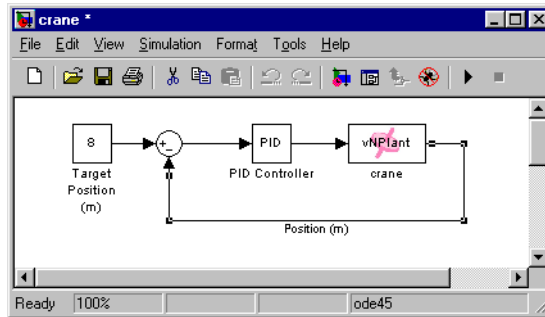
Note that the crane overshoots the target position of 8m. For the gains chosen, the control system is underdamped.

Further Exploration

- Experiment with different target positions and different gains.
- Set the velocity gain to 348.53 and the positional gain to 100. These gains make the control system critically damped, i.e., there should be no overshoot.
- Allow the hook to swing.
- Reduce control system complexity with a PID controller.

Simulink combines common gain controls into a single device called a PID controller. The control system that you constructed in this exercise can also be represented as follows:

Figure 9-19
Control System for Crane using
PID Controller



By using the PID block, you no longer need a separate output for the crane's velocity. The derivative component of the PID controller corresponds to derivative of the position, i.e. velocity. The integral component should be set to zero.

Figure 9-20
Block Parameters for PID
Controller

