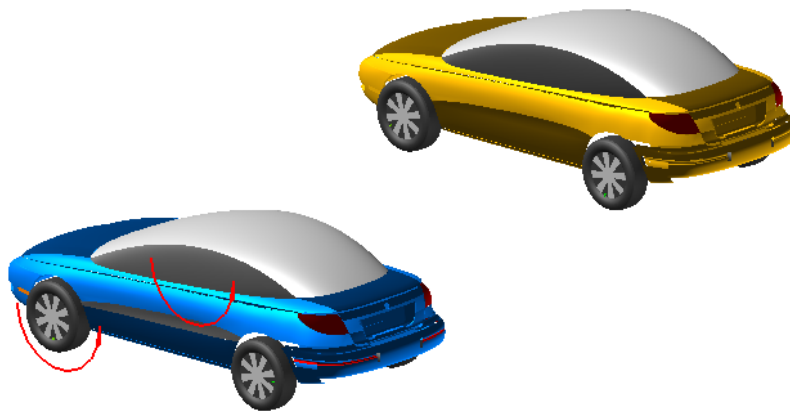




Car Tutorial (CoLink)



Copyright © 2020 FunctionBay, Inc. All rights reserved.

User and training documentation from FunctionBay, Inc. is subjected to the copyright laws of the Republic of Korea and other countries and is provided under a license agreement that restricts copying, disclosure, and use of such documentation. FunctionBay, Inc. hereby grants to the licensed user the right to make copies in printed form of this documentation if provided on software media, but only for internal/personal use and in accordance with the license agreement under which the applicable software is licensed. Any copy made shall include the FunctionBay, Inc. copyright notice and any other proprietary notice provided by FunctionBay, Inc. This documentation may not be disclosed, transferred, modified, or reduced to any form, including electronic media, or transmitted or made publicly available by any means without the prior written consent of FunctionBay, Inc. and no authorization is granted to make copies for such purpose.

Information described herein is furnished for general information only, is subjected to change without notice, and should not be construed as a warranty or commitment by FunctionBay, Inc. FunctionBay, Inc. assumes no responsibility or liability for any errors or inaccuracies that may appear in this document.

The software described in this document is provided under written license agreement, contains valuable trade secrets and proprietary information, and is protected by the copyright laws of the Republic of Korea and other countries. UNAUTHORIZED USE OF SOFTWARE OR ITS DOCUMENTATION CAN RESULT IN CIVIL DAMAGES AND CRIMINAL PROSECUTION.

Registered Trademarks of FunctionBay, Inc. or Subsidiary

RecurDyn is a registered trademark of FunctionBay, Inc.

RecurDyn/Professional, RecurDyn/ProcessNet, RecurDyn/Acoustics, RecurDyn/AutoDesign, RecurDyn/Bearing, RecurDyn/Belt, RecurDyn/Chain, RecurDyn/CoLink, RecurDyn/Control, RecurDyn/Crank, RecurDyn/Durability, RecurDyn/EHD, RecurDyn/Engine, RecurDyn/eTemplate, RecurDyn/FFlex, RecurDyn/Gear, RecurDyn/DriveTrain, RecurDyn/HAT, RecurDyn/Linear, RecurDyn/Mesher, RecurDyn/MTT2D, RecurDyn/MTT3D, RecurDyn/Particleworks I/F, RecurDyn/Piston, RecurDyn/R2R2D, RecurDyn/RFlex, RecurDyn/RFlexGen, RecurDyn/SPI, RecurDyn/Spring, RecurDyn/TimingChain, RecurDyn/Tire, RecurDyn/Track_HM, RecurDyn/Track_LM, RecurDyn/TSG, RecurDyn/Valve are trademarks of FunctionBay, Inc.

Edition Note

This document describes the release information of **RecurDyn V9R4**.

Table of Contents

Getting Started	5
Objective	5
Audience	6
Prerequisites.....	6
Procedures	6
Estimated Time to Complete	6
Opening the Initial Model.....	7
Task Objective	7
Estimated Time to Complete	7
Starting RecurDyn	8
Creating Joints and Couplers.....	9
Task Objective	9
Estimated Time to Complete	9
Creating the Revolute Joints	10
Creating the Translational Joint.....	11
Creating Couplers.....	12
Running a Simulation.....	14
Viewing the Results	15
Fixing the Revolute Joints.....	15
Refining the Model	17
Task Objective	17
Estimated Time to Complete	17
Creating the Second Car	18
Modifying the Second Car.....	20
Integrating CoLink	22
Task Objective	22
Estimated Time to Complete	22
Creating the Plant Input.....	23
Creating the Plant Outputs	24
Creating the CoLink Model	25
Creating the Proportional Feedback Control	27
Adding Derivative and Integral Control.....	29
Closing the Loop	32
Simulating the Model	33
Viewing the Results	34
Augmenting the CoLink Model	36
Task Objective	36
Estimated Time to Complete	36
Modifying the RecurDyn Model	37

Augmenting the CoLink Model	41
-----------------------------------	----



Getting Started

Objective

In this tutorial, you will simulate and control a dynamic system using CoLink, an interactive environment for designing, simulating, and testing time-varying systems. You will define a control system in CoLink to control a mechanical system that you defined in standard RecurDyn.

The system to be simulated is a simple one-dimensional problem. The scenario involves two cars, one slower car in front and one faster car coming up from behind. This situation could represent what happens when you drive around a turn and finding that someone is in front of you or when someone unexpectedly moves into your lane. The controller will automatically adjust the amount of torque being applied to the drive wheels to set the distance between the cars to a desired value. A simple PID controller will be used for this objective.

As a final part of the tutorial, you will a complex controller used to model an adaptive cruise control system.

Audience

This tutorial is intended for intermediate users of RecurDyn who previously learned how to create geometry, joints, and force entities. All new tasks are explained carefully.

Prerequisites

You should first work through the 3D Crank-Slider and Engine with Propeller tutorials, or the equivalent. We assume that you have a basic knowledge of physics.

Procedures

The tutorial is comprised of the following procedures. The estimated time to complete each procedure is shown in the table.

Procedures	Time (minutes)
Opening the initial model	5
Creating joints and coupler	15
Refining the model	15
Integrating CoLink	15
Argumenting CoLink	15
Total	65



Estimated Time to Complete

This tutorial takes approximately 65 minutes to complete.



Opening the Initial Model

Task Objective

Learn how to import a mechanical model in preparation for refining the model for use with a control system.



Estimated Time to Complete

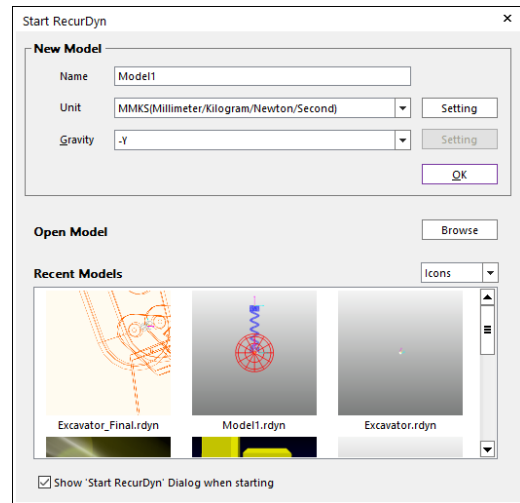
5 minutes

Starting RecurDyn

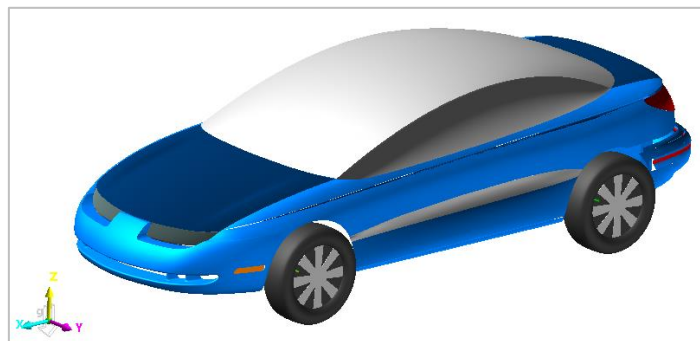
To start RecurDyn and open the initial model:



1. On your Desktop, double-click the **RecurDyn** icon.
2. When the Start RecurDyn dialog box appears, **Close** dialogue because you will not be creating a new model but using an existing one.
3. From the **File** menu, click **Open**.
4. From the **CoLink** tutorial directory (<Install Dir> \Help \Tutorial \Colink \Car), select the file **CarCruise_Initial.rdyn**.
5. Click **Open**.



Your model should look like the following.



To save the initial model:

1. From the **File** menu, click **Save As**.
2. Save the model different directory, because you cannot simulation in tutorial directory.

Chapter

3

Creating Joints and Couplers

Task Objective

In this chapter, you will create the joints for the wheels and couple them to the forward velocity of the car. You will also define the initial velocity of the car and the friction in the translational joint that represents the rolling drag resistance of the car. You will then run a simulation.



Estimated Time to Complete

15 minutes

Creating the Revolute Joints

In this section, you create revolute joints that attach the wheels to the car body.

To create the joints:



1. Change the working grid orientation to the **XZ plane**. shortcut is (shift + A).
This guarantees that the revolute joints will be oriented correctly.

Tip: Press C to center the car.



2. Press **A** to turn on Auto Operation mode to do one task continuously.

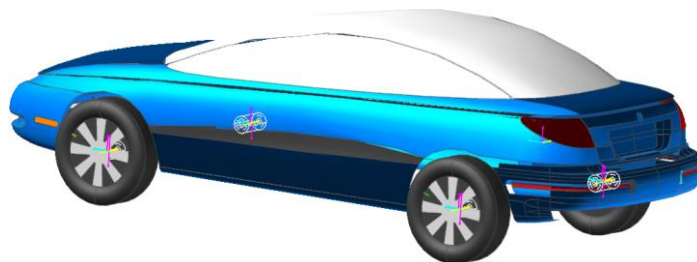
Tip: Auto Operation mode helps when you want to perform the same task repeatedly, by saving you from having to select the same task from the ribbon menu multiple times. Once you turn on Auto Operation and select a task from the ribbon menu, RecurDyn repeats that task after each completion, until you exit Auto Operation mode.



3. From the **Joint** group in the **Professional** tab, click the **Revolute**.
4. Set the **Creation Method** to **Body, Body, Point**.
5. For each of the four wheels, create revolute joints between the car body (**Car_Body**) and the wheel at the center of mass marker of the wheel.
 - Click **Car_Body**.
 - Click a **wheel**.
 - Click the marker at the center of that wheel.
 - Repeat these steps for each of the four wheels.
6. When you are done creating the four joints, press **A** again to turn off Auto Operation mode and then press **Esc** to cancel the creation of another revolute joint.
7. Rename the four newly-created joints to something meaningful, such as **Rev_Front_Right, Rev_Front_Left, Rev_Rear_Right, Rev_Rear_Left**.

Tip: In the Database window, right-click a revolute joint, click **Property**, and change the name.

Your model should now look like the following:

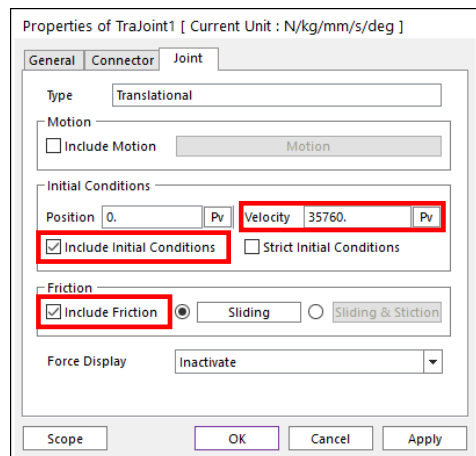
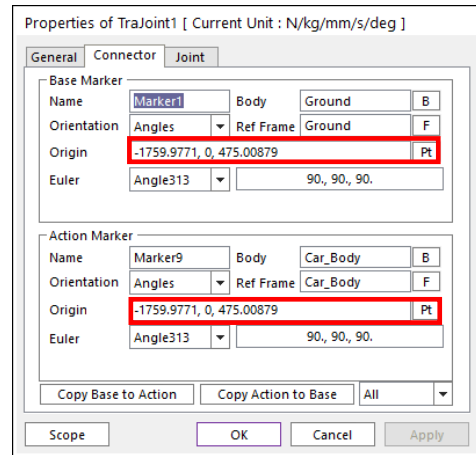
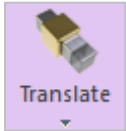


Creating the Translational Joint

Create a translational joint that defines the motion of the car.

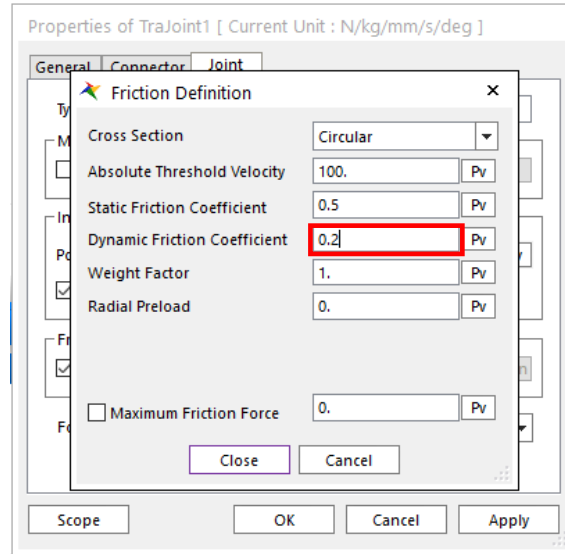
To create the joint:

1. From the **Joint** group in the **Professional** tab, click the **Translate**.
2. Using the **Body, Body, Point, Direction** creation method, select **Ground**, the **Car_Body**, and enter the following coordinates:
 - -1759.9771, 0, 475.00879
3. Click the **+X-axis** of **Ground InertiaMarker**.
4. Change the properties of the translational joint (right-click joint and click **Property**).
 - On the **General** page in the Properties dialog box, rename the translational joint to **Tra_Car_Ground**.
 - On the **Joint** page, provide the joint with an initial velocity:
 - a. Click **Include Initial Conditions**.
 - b. Set **Velocity** to **35760** mm/s (35.76 m/s or 80 mph).



You will also include friction in the model to simulate drag and friction that normally decelerate a car.

5. Click **Include Friction**.
6. Click **Sliding** and change **Dynamic Friction Coefficient** to **0.2** as shown below.



7. Click **Close** and then **OK** to accept changes and exit.

Now you are ready to create couplers between the wheels and the forward motion of the car.

Creating Couplers

Create couplers so the angular velocity of the tires matches the forward velocity of the car.

To create the couplers:



1. From the **Joint** group in the **Professional** tab, click the **Coupler**.
2. Using the **Joint, Joint, Joint** creation method, click **Tra_Car_Ground** then click **Rev_Front_Right** and **Rev_Front_Left**.
3. Right-click the newly-created coupler to display its Properties dialog box. Calculate the scale factors that are needed to define this coupler.

Tip: Defining the scale factors for the coupler

The coupler is defined by the following equation, where the d 's represent the scale factors and the u 's represent the velocities of the corresponding joint:

$$d_1 \times u_1 + d_2 \times u_2 + d_3 \times u_3 = 0$$

The forward velocity of the car is related to the rotation of the tires by the basic relationship, $v = w \times r$ where:

- v is the velocity of the vehicle.
- w is the angular velocity of the wheels.
- r is the wheel radius.

This, of course, assumes that the car is only moving forward in a straight line so the wheels are both rotating at the same velocity. This will be enforced by an additional coupler in the next step. For the provided geometry, the wheel radius is 336 mm so the equation above becomes the following:

$$d_1 \cdot r \times w + d_2 \times w + d_3 \times w = 0$$

$$d_1 \cdot r + d_2 + d_3 = 0$$

Because the wheels both have the same radius $d_2=d_3$, assume that $d_1=1$ because the scale factors are relative and one of them needs to be chosen. This leads to the following:

$$r + d_3 + d_3 = 0$$

$$d_3 = -r/2$$

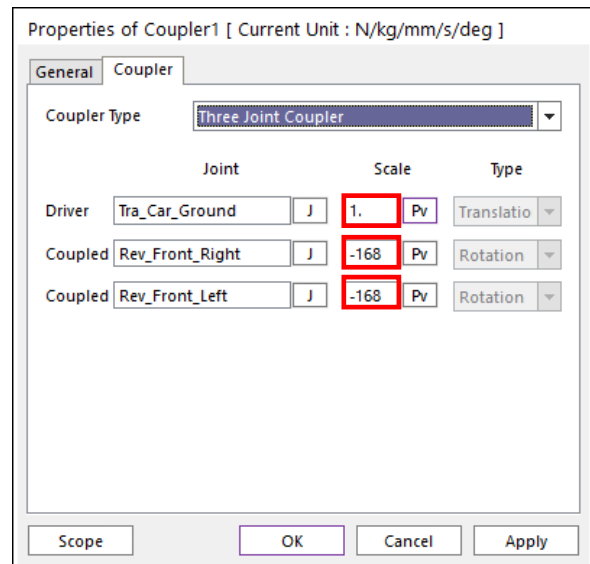
$$d_3 = -336/2$$

$$d_3 = -168$$

$$d_2 = -168$$

In summary, the scale factor for the translational joint is 1 and -168 for both revolute joints.

4. In the **Coupler** page of the Coupler Properties dialog box, enter the scale factors: **1, -168** and **-168**.
5. In the **General** page, change the name to **Coupler_Front** and click **OK**.
6. Repeat the procedure from Steps 1 through 5 for the rear wheels, naming the coupler **Coupler_Rear** and using the same scale factors.

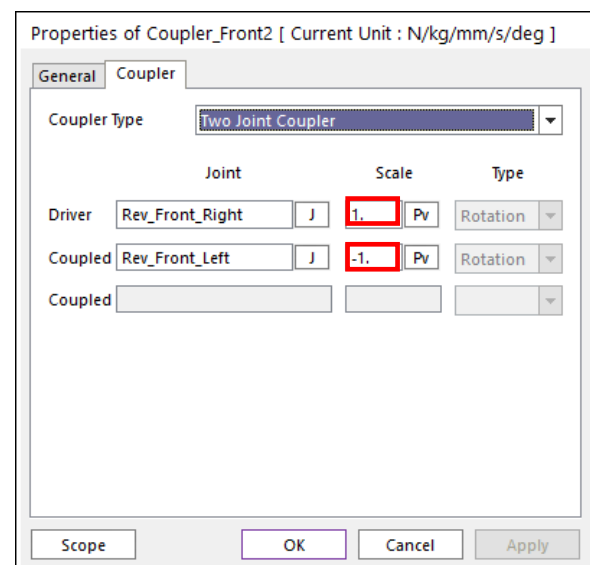


As mentioned previously, now you will create a coupler between the two front wheels to guarantee that they rotate at the same angular velocity.

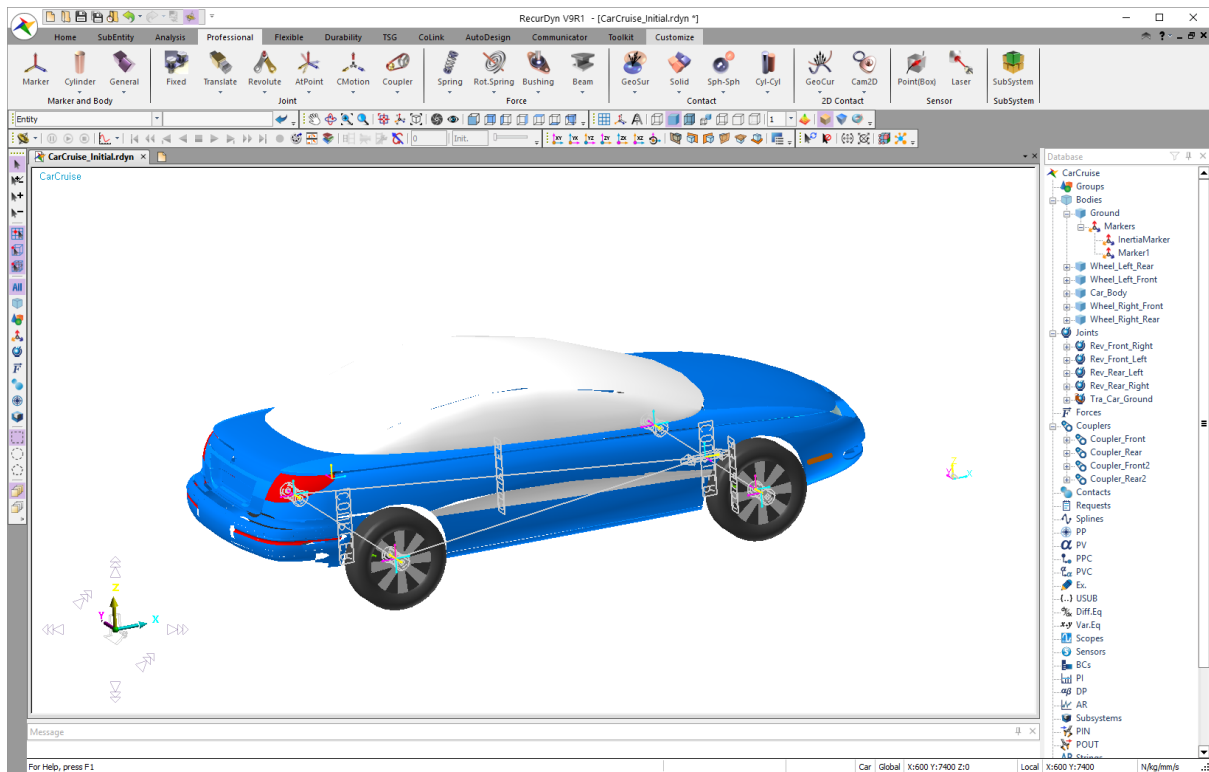
To create a coupler:



1. Click the **Coupler** of the **Joint** group in the **Professional** tab.
2. Using the **Joint, Joint** creation method, click **Rev_Front_Right** and then **Rev_Front_Left**.
3. Edit the properties of the newly created coupler, changing:
 - Its name to Coupler_Front2.
 - Scale factors to **1** and **-1**, essentially saying that the two rotations must be equal.
 - Click **OK**.
4. Repeat Steps 1 through 3 for the rear wheels, naming the coupler **Coupler_Rear2**, using the same scale factors (**1** and **-1**).



After completing these steps your model should appear as shown next.



Running a Simulation

You are now ready to run a simulation.

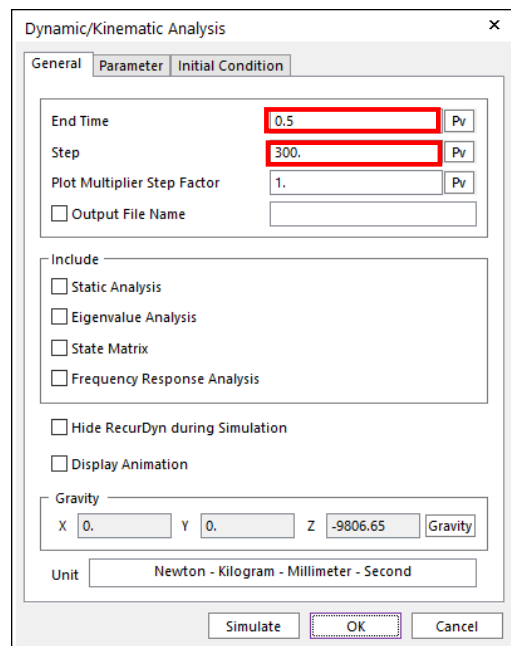
To run a simulation:



1. Click the **Dyn/Kin** of the **Simulation Type** group in the **Analysis** tab.
2. Set the simulation to run for **0.5** seconds with **300** steps as shown in the figure on the right.

This provides adequate time to view the motion of the wheels and the car.

3. Click the **Parameter** tab and set the **Maximum Time Step** to **1.e-003**.
4. Click **Simulate**.



Viewing the Results

To view the results:

- From the **Animation Control** group in the **Analysis** tab, click the **Play** button in the animation controls.

Tip: What if the car has no motion?

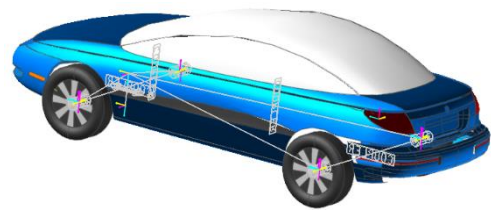
Carefully check the magnitude and sign of all of the coefficients that you set up in the couplers. For example, the car will not move if you have both coefficients set to +1.

Looking at the results of the simulation, notice that there is a flaw with this setup. Depending on how you set up the revolute joints, some of the wheels may be rotating in the wrong direction as the car drives forward. The solution is to make sure that the Z-axes of all of the revolute joints point in the same direction, which you will do next.

Fixing the Revolute Joints

To fix the joints:

- For the revolute joint of any wheel that is rotating in the wrong direction, in the Database window, right-click joint name, and click **Property** to display its Properties dialog box.



The steps below use the **Rev_Rear_Right** joint as an example.

- Change the 313 Euler angles from **0, 90, 0** to **0, -90, 0** for both the **Base** and **Action Markers** and click **Apply**.

Properties of Rev_Rear_Right [Current Unit : N/kg/mm/s/deg]

General				Connector				Joint							
Base Marker															
Name	Marker1	Body	Car_Body	B	Orientation	Angles	Ref Frame	Car_Body	F	Origin	-4361.9771, -873.16071, 475.00879	Pt			
Euler	Angle313	180,90,180.													
Action Marker															
Name	Marker2	Body	Wheel_Right_Rear	B	Orientation	Angles	Ref Frame	Body1	F	Origin	-4361.9771, -873.16071, 475.00879	Pt			
Euler	Angle313	180,90,180.													
Copy Base to Action				Copy Action to Base				All							
Scope				OK				Cancel				Apply			

Note: RecurDyn changes the Euler angles from **0, -90, 0** to **180, 90, 180**. This is not a problem because both combinations of Euler angles position the markers in the same orientation.

(Refer to the Tip on the next page for additional information about working with Euler angles.) This changes the orientation of the markers and the dialog box appears as shown on the right

3. Click **OK**.
4. Repeat these steps for the other joints as needed.
5. Simulate the model again to verify that the problem is corrected.

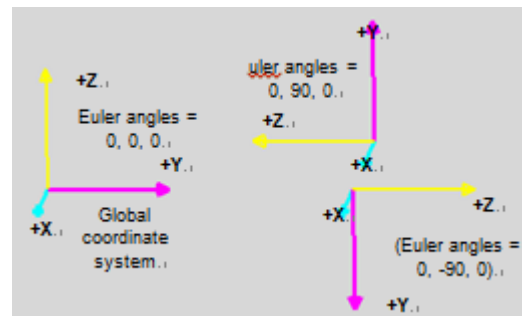
This car is now complete. Next, you will make a copy of the car and set up the model for integration with CoLink.

Tip: Working with Euler Angles

The standard Euler angles in RecurDyn are

3-1-3 Euler angles. This angular transformation is made by:

- First, rotating about the #3 (or +Z) axis.
- Then, rotating about the new #1 (or +X) axis.
- Finally, rotating about the new #3 (or +Z) axis.



The coordinate system in the figure to the right is the global coordinate system. A positive rotation about the +X-axis of 90 degrees results in the coordinate system shown in the upper right of the figure. A negative rotation about the +X-axis of 90 degrees results in the coordinate system shown in the lower right of the figure. Notice that changing the Euler angles from 0, 90, 0 to 0, -90, 0 reverses the direction of the Z-axis.

Chapter

4

Refining the Model

Task Objective

In this chapter, you will refine the model by copying the first car to make a second car. You will then move the second car and change it to a different color and to drive at a constant slower speed.



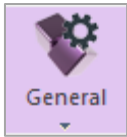
Estimated Time to Complete

15 minutes

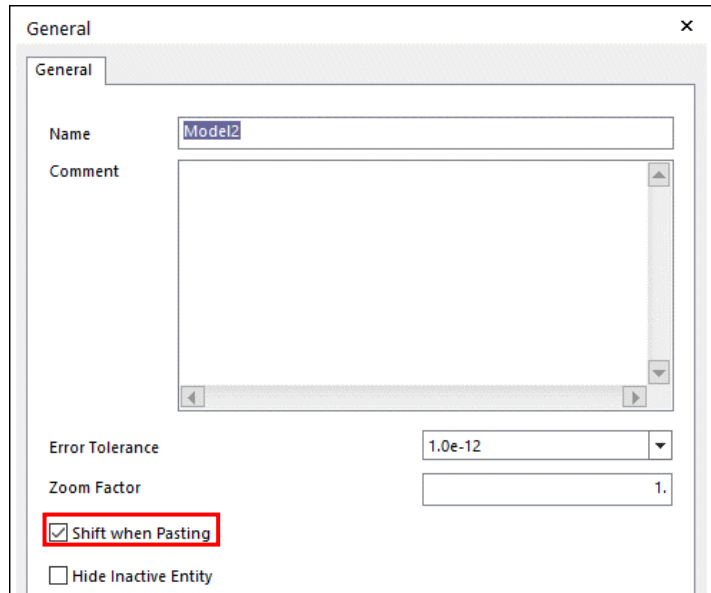
Creating the Second Car

To create the second car, you will make a copy of the first one, move it, and then modify it.

To copy the car:



1. Click the **General** of the **Setting** group in the **Home** tab and clear the selection of **Shift When Pasting** shown on the right.
2. Click **OK**.
3. To select all of the vehicle, draw a selection box around the entire vehicle.
4. Press **Ctrl-C** to copy and then **Ctrl-V** to make a duplicate of the car, joints, and couplers.

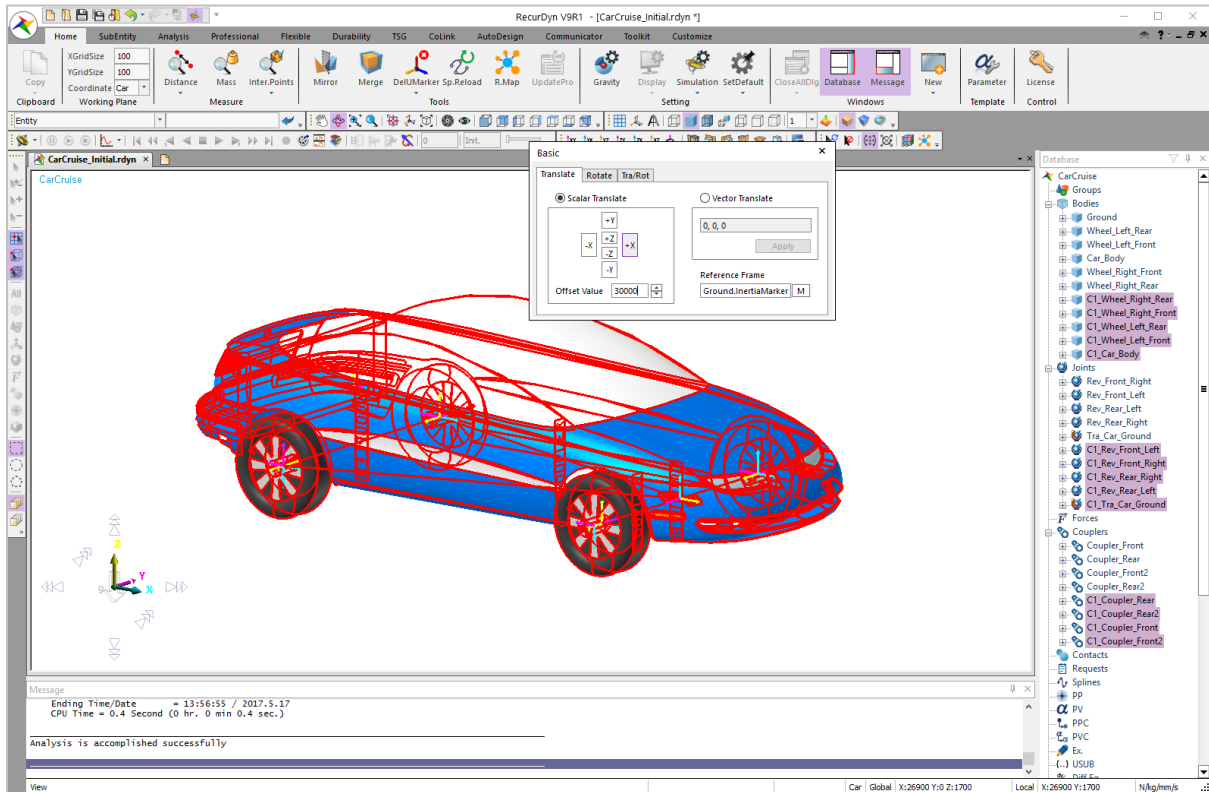


Tip: You can also click **Copy** of the **Clipboard** group in the **Home** tab and then **Paste** of the **Clipboard** in the **Home** tab.



5. Without clicking anywhere on the model so the copied model remains selected, from the tool bar, clicks the **Object Control** tool.

- Move the duplicated car **30,000 mm** in the **+X** direction as shown in the following figure.



Modifying the Second Car

To modify the duplicate car:

1. To display the Properties dialog box for the translational joint of the second car, in the Database window, right-click **C1_Tra_Car_Ground** and click **Property**.
2. Because this second car will be going at a constant speed, clear the selection of **Include Friction**.
3. Click **OK**.
4. You also need to decrease the speed at which it is traveling, so change the initial velocity (**X**) of the body **C1_Car_Body** to **22,350 mm/s**, which is 50 mph.

Tip: To change the initial velocity of C1_Car_Body

1. Click **Property** on the **C1_Car_Body** in the Database window.
2. Click the **Body** tab.
3. Click the **Initial Velocity** button.
4. Click **X** of **Translational Velocity**.
5. Enter **22350**.
6. Define **Reference Marker** of **Translational Velocity** to **Ground.InertiaMarker**.
7. Define **Reference Marker** of **Rotational Velocity** to **C1_Car_Body.CM**.
8. Click **Close**.
9. Click **OK**.

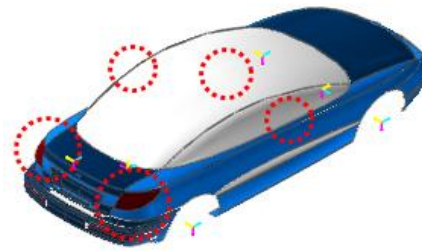
-
5. Now you will change the color of the car to be something more appropriate for a car going this slowly on a freeway.
 - To enter **Body Edit** mode, double-click the duplicate car's body (**C1_Car_Body**).

Tip: To enter Body-edit mode, you can also right-click **C1_Car_Body** in the database window and click **Edit**.

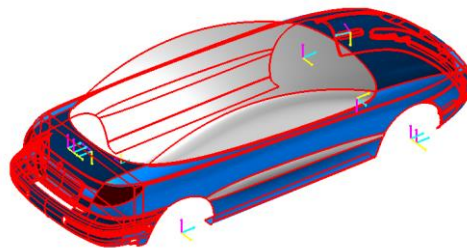
- Draw a selection box around the entire car body. Then, while holding down the **Ctrl** key, click the three white portions of the roof and windows, and the two red taillights to clear their selection.

Tip: Rotate the car so you see the rear or from the Database window, clear the selection of the following:

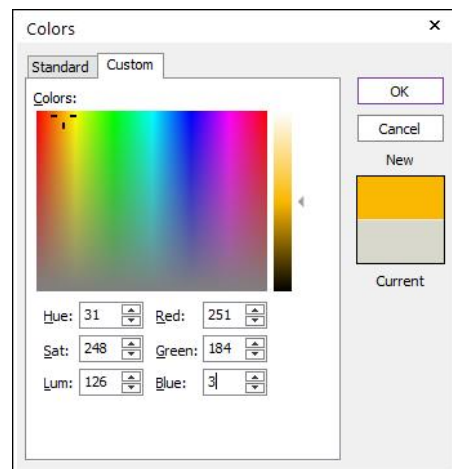
- C1_C1_C2_ImportSurface1
- C1_C1_C3_ImportSurface1
- C1_C3_ImportService1
- C1_C2_C1_ImportSurface1
- C1_C2_C2_ImportSurface1



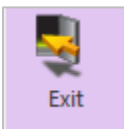
Your model should look like the following.



- Right-click anywhere on the selected body and click **Property**.
- Click the **Graphic Property** tab and set **Color** to other.
- In the **Custom** page of the Colors dialog box, change the **Red, Green, and Blue** values to **251, 184, and 3**, respectively.
- Click **OK** twice to accept and apply the color change.



You may have to wait a few minutes for the changes to take effect after clicking OK depending on the speed of your processor.



6. Click **Exit** to leave **Body Edit** mode and return to the main model level.

Your model is now ready for integration with CoLink.



Chapter

5

Integrating CoLink

Task Objective

In this chapter, you will set up the model for integration with CoLink and create the CoLink model. You will then simulate the system and plot the results.



Estimated Time to Complete

15 minutes

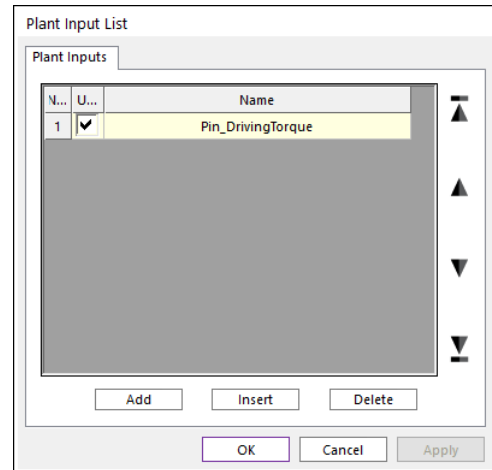
Creating the Plant Input

You will first create inputs to the model from the control system. This entity will be created as a placeholder that you will then go back and define.

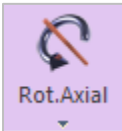
To create the Plant Input:



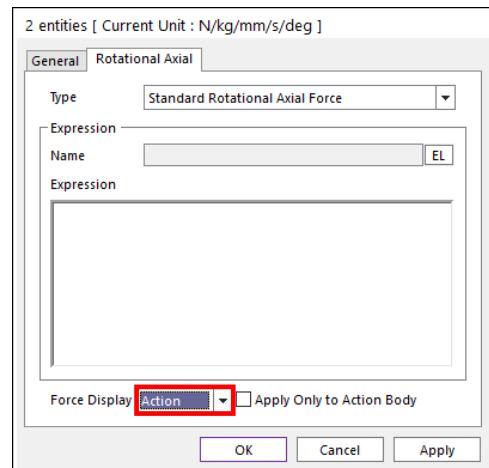
1. From the **Colink** group in the **Colink** tab, click the **Plant Input**.
2. Click **Add** and change the name of the plant input to **Pin_DrivingTorque**.
3. Click **OK**.



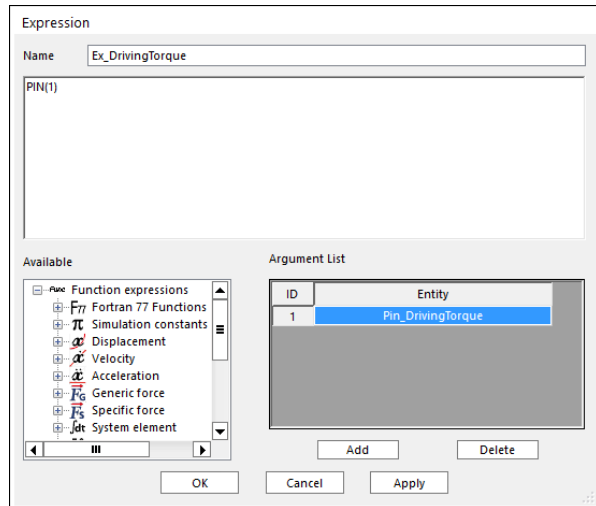
To create the torques:



1. From the **Force** group in the **Professional** tab, click the **Rotational Axial**.
2. Set the **Creation Method** to **Joint**, select **Rev_Front_Right**.
3. Repeat these steps for **Rev_Front_Left**.
4. Click **RotationalAxial1** and while holding down **Ctrl**, click **RotationalAxial2**. With both selected, right-click one of them and click **Property**.
5. To display the calculated torque in the animation, set **Force Display**, near the bottom of the dialog box, to **Action**.
6. Click **EL** to enter an expression for the driving torque.



7. In the Expression List dialog box, click **Create**.
8. Name the expression **Ex_DrivingTorque** and for the expression itself, type **PIN(1)**.
9. In the **Argument List**, click **Add** and then from the Database window, drag **Pin_DrivingTorque** to the first entity, as shown in the figure on the right.



Tip: Alternatively, you can double-click the yellow box and type **Pin_DrivingTorque**.

10. Click **OK** three times to accept all changes.

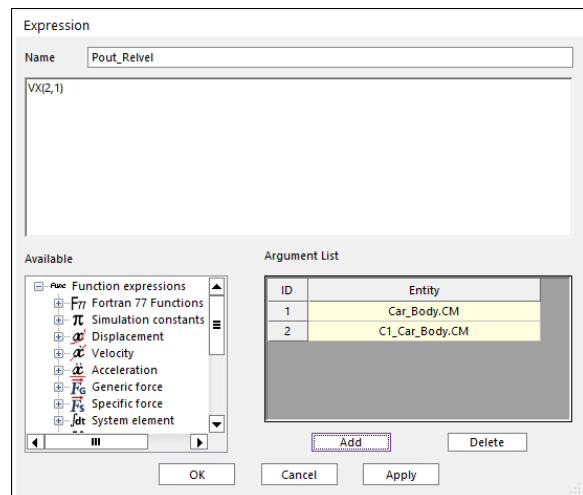
Creating the Plant Outputs

Output the relative velocity between the two cars, as well as the distance between them, from the model to the control system.

To create the plant outputs:



1. From the **Colink** group in the **Colink** tab, click the **Plant Output**.
2. In the Plant Output List dialog box, click **Add**.
3. In the Expression dialog box, change the name to **Pout_RelVel** and type **VX(2,1)** as the expression.
4. Add two entities to the **Argument List** and enter the CM markers for Car_Body and C1_Car_Body into the yellow boxes, as shown on the right.
5. Click **OK** to accept and return to the Plant Output List.
6. Click **Add** in the **Plant Output List** again and repeat Steps 3 through 5 using **DX** instead of **VX** and naming it **Pout_Distance** instead of **Pout_RelVel**.
7. Click **OK**



Creating the CoLink Model

You will now open CoLink and create the control system. This will involve creating the block diagram from scratch. Blocks will be created to represent the RecurDyn model, gains for a standard PID controller, summation blocks, and a constant reference signal.

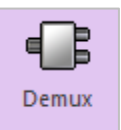
To create the CoLink model:



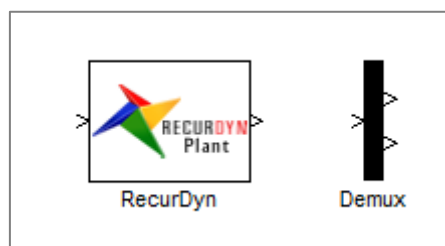
1. Click the **Run CoLink** of the **Colink** group in the **Colink** tab
CoLink opens with an empty model.



2. From **Link** group in the **Connector** tab, click the **RecurDyn** block and then, click into the model window, as shown below.



3. From the **Link** group in the **Connector** tab, click the **Demux** block and then, click to the right of the **RecurDyn** block in the model window, as shown below:

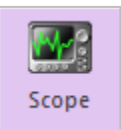


This splits the single output from the **RecurDyn** block into the two outputs you created in the RecurDyn model: **RelVel** and **Distance**.

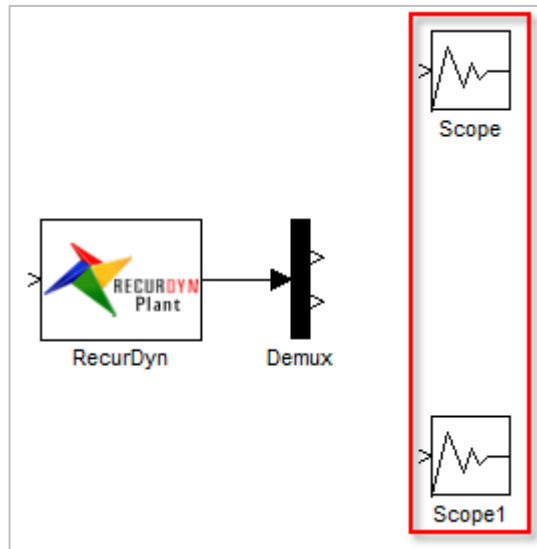
4. To connect the two blocks, click the **RecurDyn** block and hold down **Ctrl** key and click the **Demux** block.

This is how you connect blocks. Use the same procedure during the rest of the CoLink model creation.

You will want to view the results of your simulation, so create scopes for both the relative velocity and distance signals. Scopes draws x-y plots of simulation time (x) and input data (y).

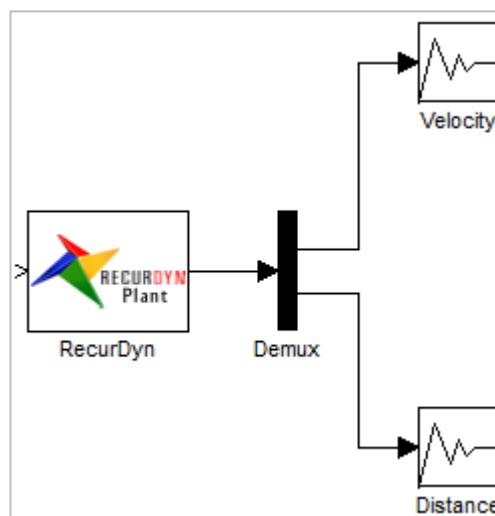


5. From the **Output** group in the **General** tab, click the **Scope** block and then, click into the model window twice. Place one on the top right and one on the bottom right, as highlighted in the figure below.



6. Connect the outputs from the **Demux** block to the two scopes using the same method described in Step 5.
7. Double-click the text below the top scope and change its name to **Velocity**. Change the name of the bottom scope to **Distance**.
8. Save the model as **CarCruise_Control**.

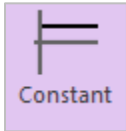
Your model should look like this.



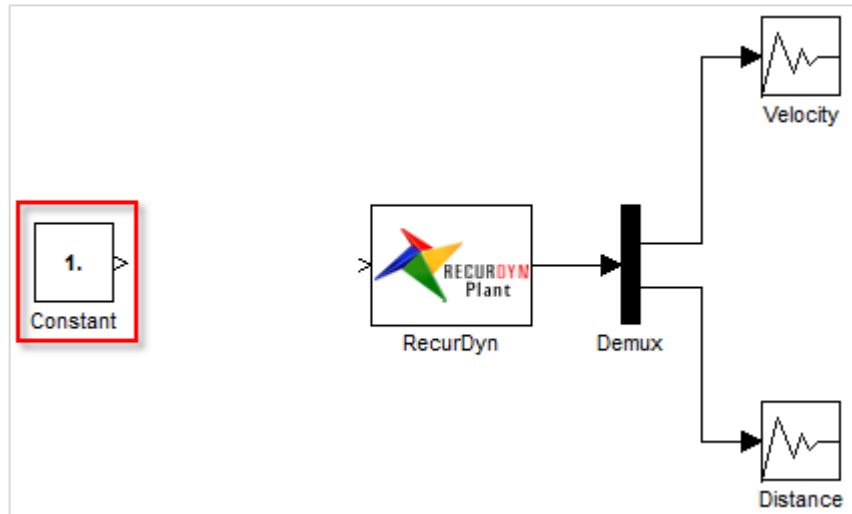
Creating the Proportional Feedback Control

You will now create a reference input signal and a feedback loop and add proportional control. This is the first step in building a PID controller.

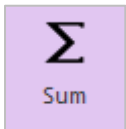
To create the proportional control:



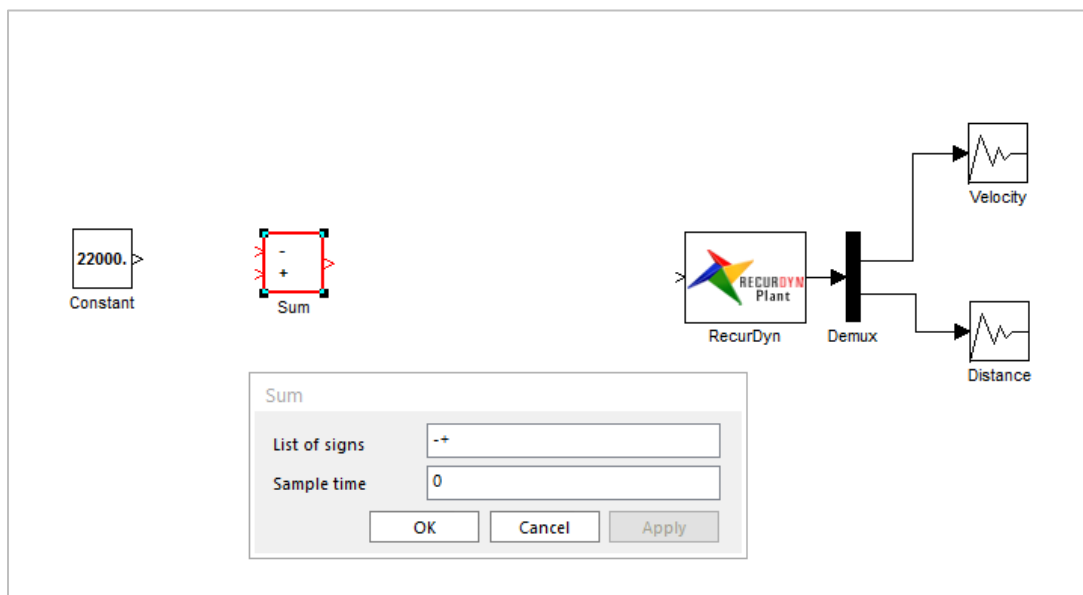
1. In the **General** tab, click the **Constant** block of the Source group and then, click into the far left middle of the model window, as highlighted in the figure below.



This constant will represent the desired distance between the cars.



2. Double-click the **Constant** block in your model and change the constant to **22,000**.
3. In the **Math** tab, click the **Sum** block of the Math group and then, click into the model window, placing it just to the right of the **Constant** block.
4. Connect the **Constant** block to the top input of the **Sum** block and then double-click the **Sum** block and change the list of signs to **-+** as shown below.

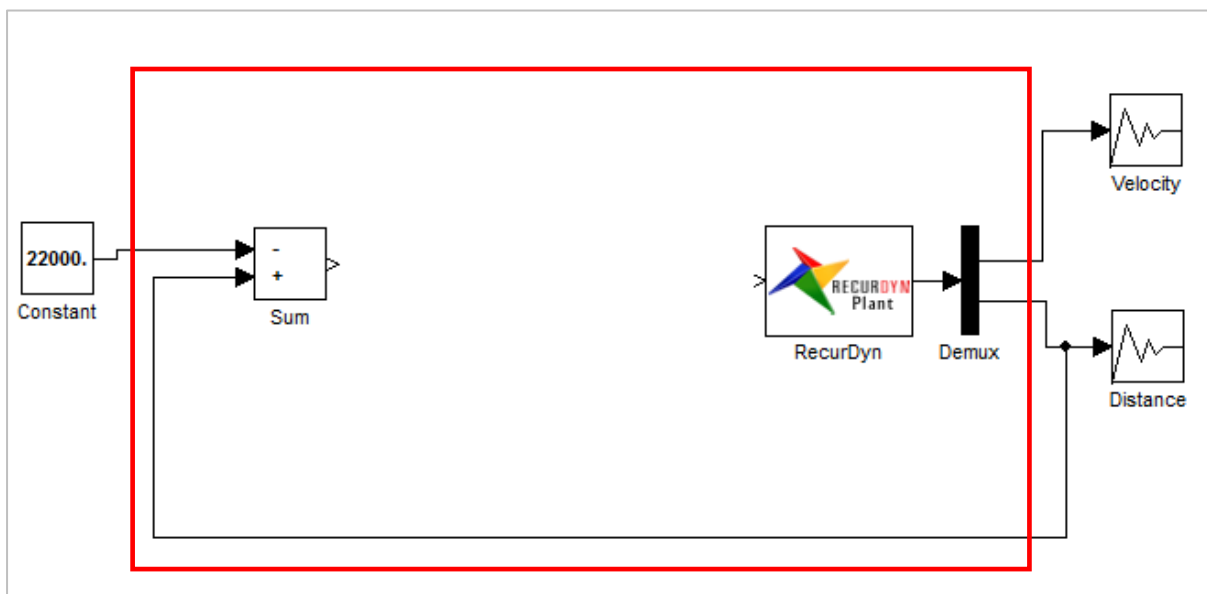


5. Click **OK**.

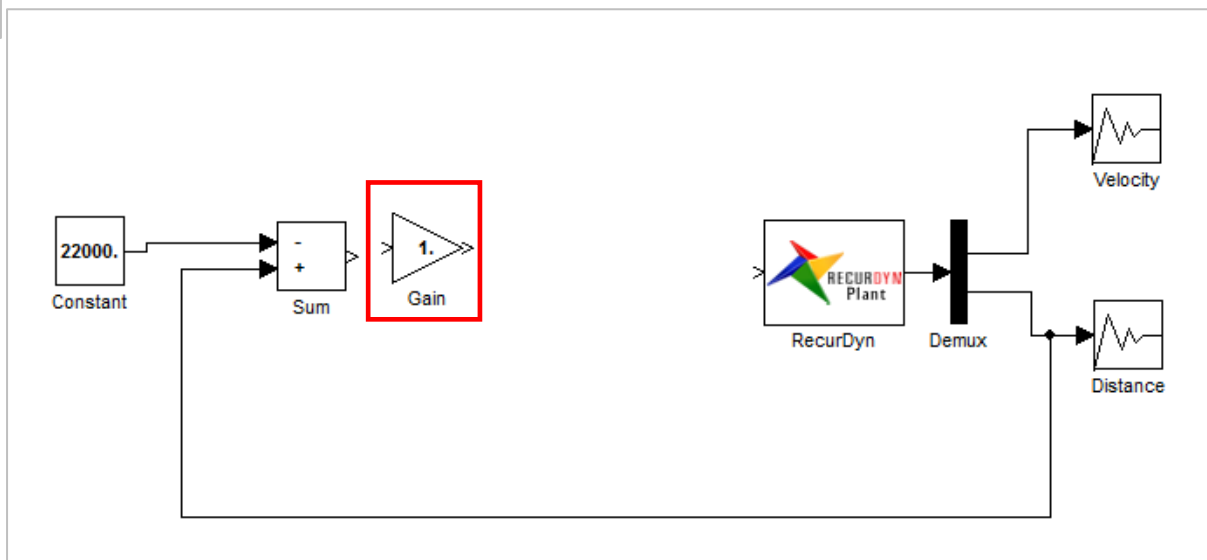
Now you need to connect the **Distance** signal to the bottom input of the **Sum** block. This creates an error signal representing the actual distance between the cars minus the desired distance, 22,000.

6. Connect the **Distance** output to the **Sum** block as follows:

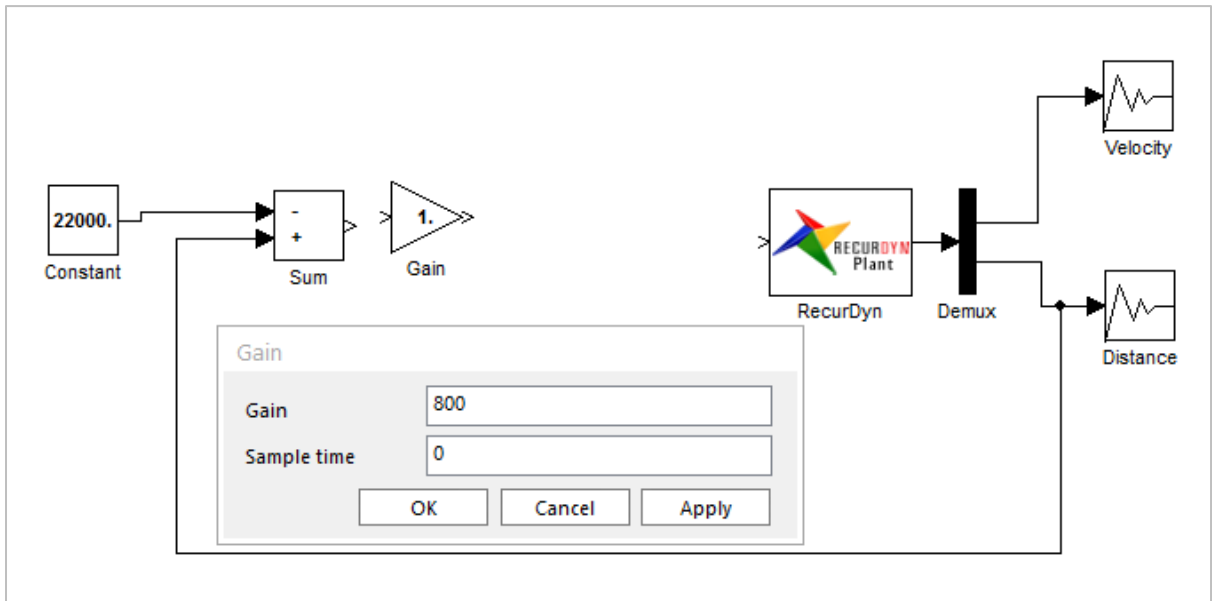
- Click once on the line connecting the **Demux** block to the **Distance Scope** to select it.
- Right-click the line where it bends just before entering the scope and drag your cursor to the bottom input of the **Sum** block, then release the right mouse button. This creates a junction at the point where you first right clicked. Your model should now look like this.
-



7. In the **Math** tab, click the **Gain** block of the Math group and then, click into the model window and place it to the right of the **Sum** block, as highlighted in the figure below.



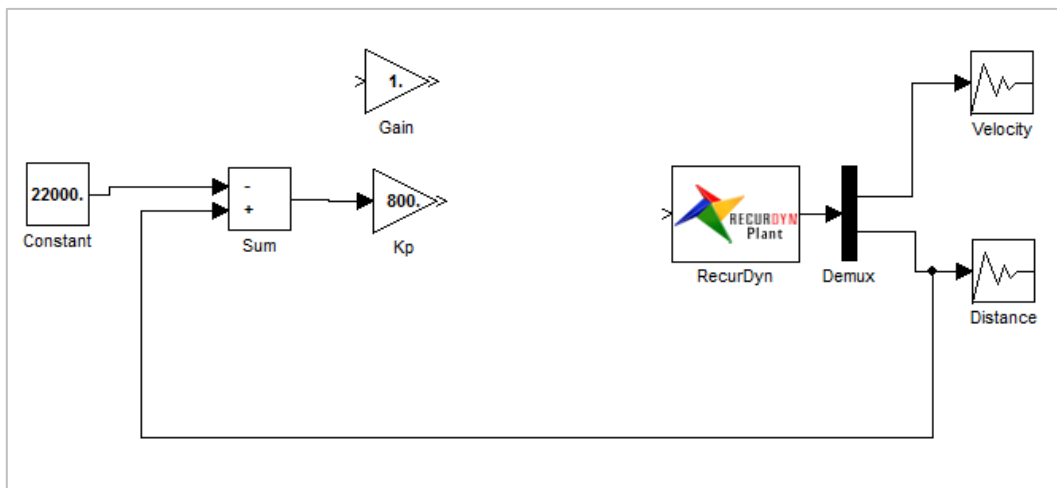
- Change the gain to **800** and connect the two blocks as shown below.



- Rename the Gain block **Kp** because it is the gain for the proportional part of your PID controller.

Adding Derivative and Integral Control

Now, you will add derivative and integral feedback to the controller. Because the Velocity signal is the derivative of the Distance, you will use it directly as the input to the derivative gain block. For the integral control, you will need to add an integrator block and use that as the input to the integral gain block.



To add derivative control:

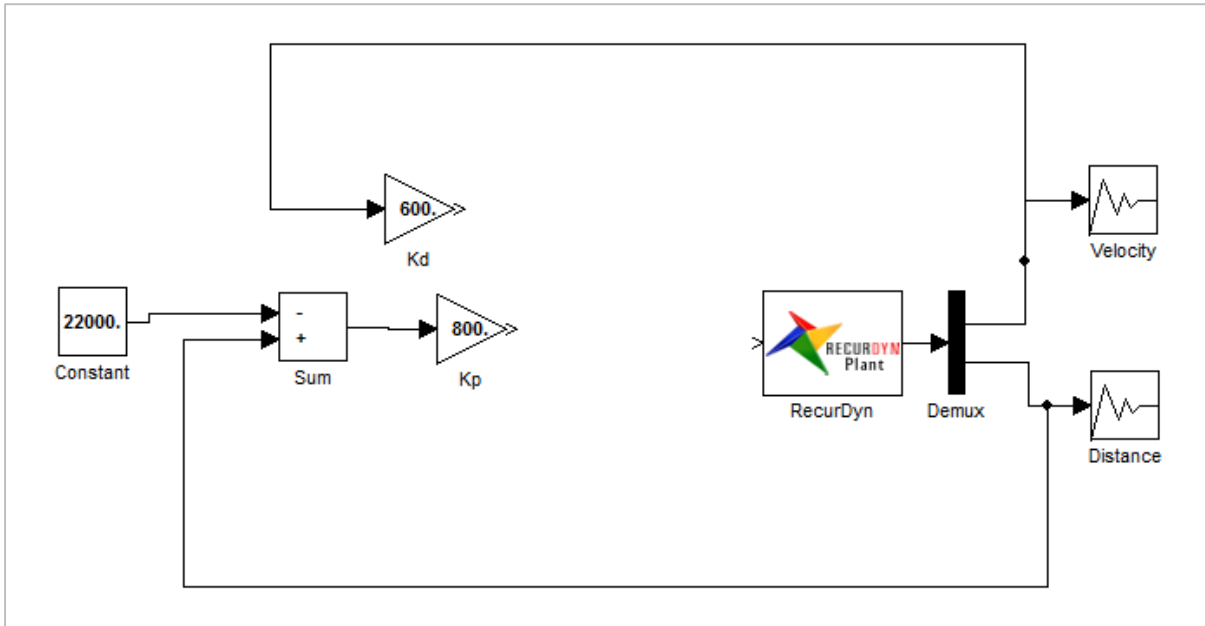
- In the **Math** tab, click the **Gain** block of the Math group and then, click into the model window and place it above the first one that is now labeled **Kp**, as highlighted in the figure below.

2. Rename the block **Kd** because it will be the gain for the derivative control.
3. Connect the **Velocity** output to the **Gain** block, as shown in the figure below:
 - Click once on the line connecting the **Demux** block to the **Velocity Scope** to select it.
 - Right-click the line where it bends just before entering the scope and drag your cursor to the input of the **Gain** block, then release the right-mouse button.



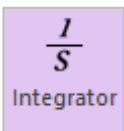
This creates a junction at the point where you first right-clicked.

4. Change the gain value to **600**.
- Your model should now look like the following:



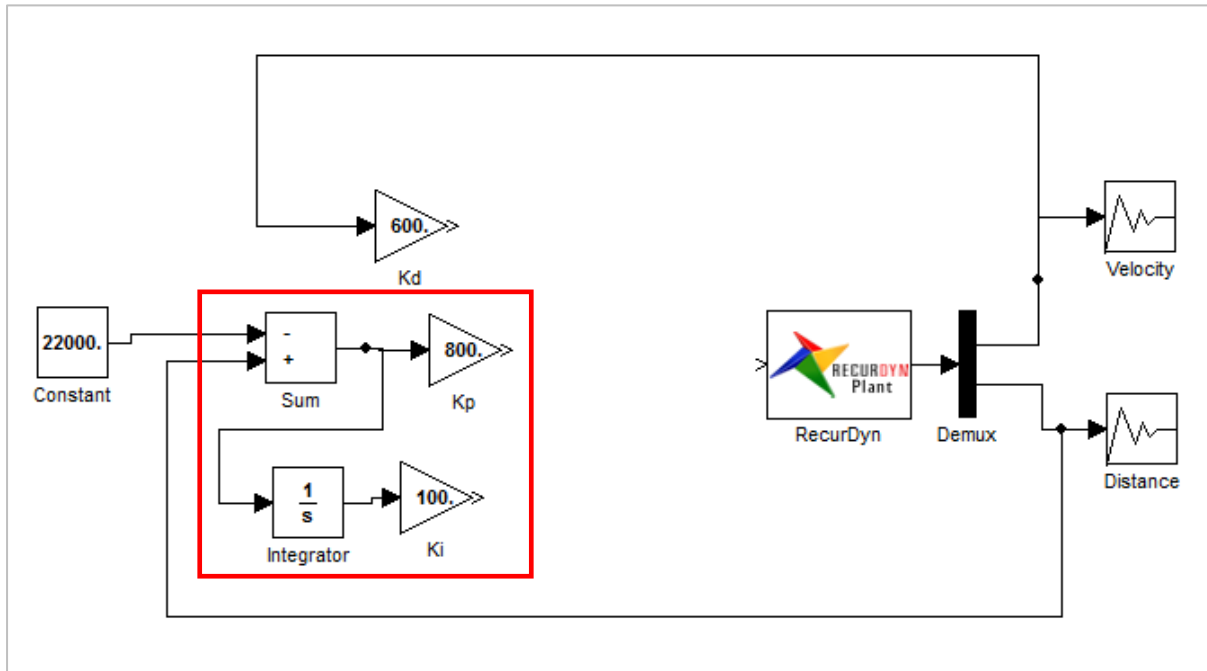
To add integral control:

1. In the **Math** tab, click the **Gain** block of the Math group and then click into the model window and place it below the **Kp** gain block, as highlighted in the figure below.
2. Rename the block **Ki** and change its value to **100**.
3. From the Continuous block in the **Continuous and Discrete** tab, click the **Integrator** block.



4. Right-click the signal connecting the **Sum** block and the **Kp** block and feed that into the **Integrator** block.
5. Connect the **Integrator** to the **Ki** block.

Your model should now appear as shown next.



Closing the Loop

Now, you will close the loop by adding all three feedback signals together and connecting the sum to the RecurDyn plant block.

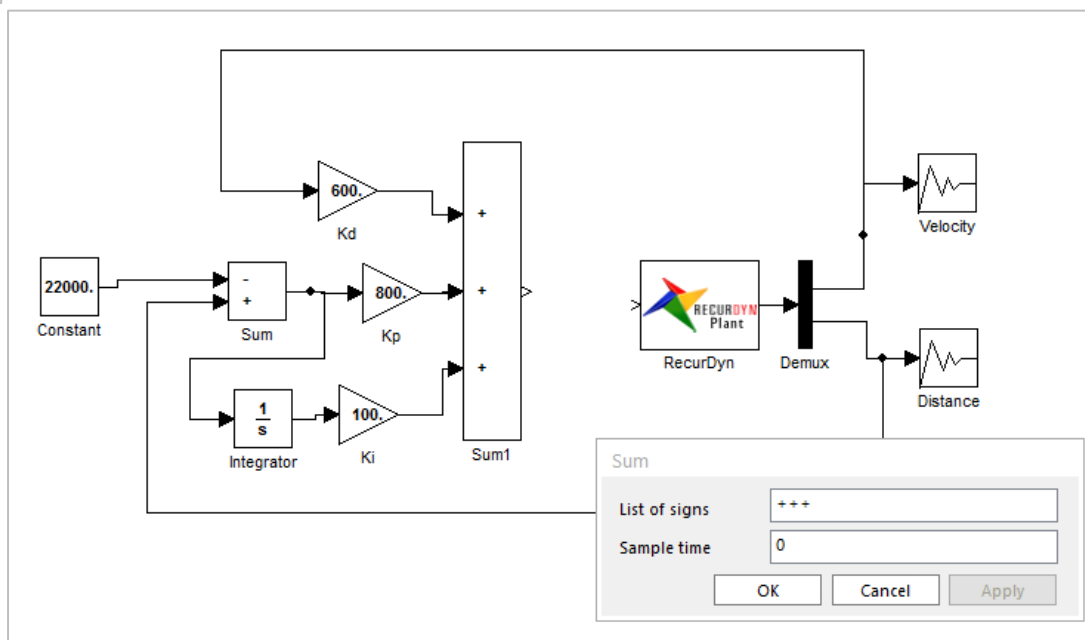
To close the loop:

1. In the **Math** tab, click the **Sum** block of the Math group and then click into the model window and place it between the **Gain** blocks and the **RecurDyn** block, as highlighted in the figure below.

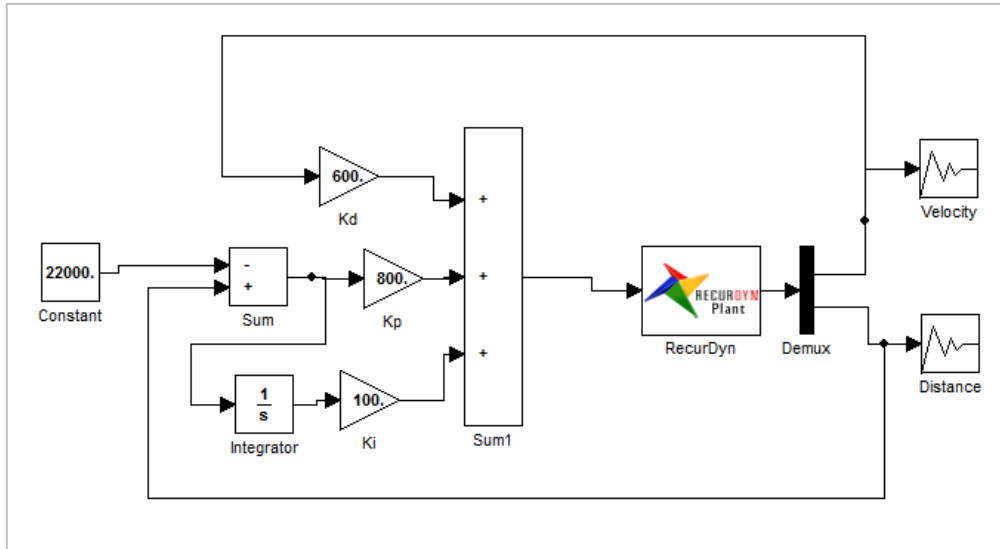
Σ

Sum

2. Double-click the newly added **Sum** block and change the list of signs to **+++**.
This adds a third input so all three elements of the feedback controller can be specified as inputs to the summation block.



3. Connect the **Gain** blocks to the summer and the summer to the RecurDyn plant.
Congratulations, you have finished building your controller model. It should now look like this.

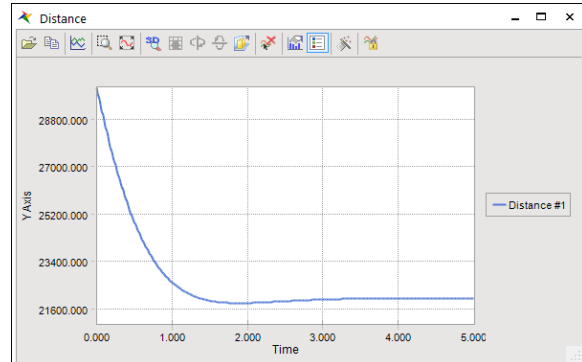
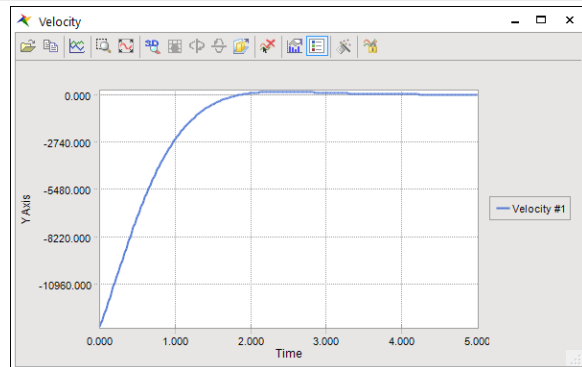
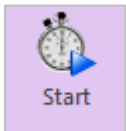
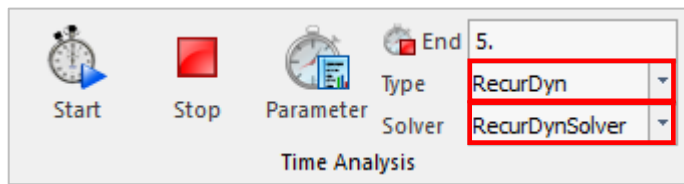


Simulating the Model

Now simulate the model and view the results.

To simulate the model:

1. Change the **Type** to **RecurDyn** and change the **Solver** to **RecurDynSolver**
2. From **Time Analysis** group in the **Simulation** tab, click the **Start**.
3. When the simulation has completed, double-click the **Velocity** and **Distance** scopes to view the response. The response plots should look like the figure on the right.



The Velocity converges to zero, indicating that the two cars maintain a fixed distance after the transient response has died out. The Distance plot shows that the final value is 22,000 mm (22 m) as well, as it should be. The response time is a bit slow, however, and the overshoot is too great. You will now fix these and run the simulation again.

Tip: What if the co-simulation does not run?

If the co-simulation does not run, a Server Busy dialog box may appear. If the dialog box appears:

- Click **Switch** to and review the message in the RecurDyn Output window.

If the error message is that the CoLink model could not be found, review the locations of your RecurDyn and CoLink models. Make sure that both models are located in the same directory. If not, place both models in the same directory and restart both RecurDyn and CoLink. The simulation should now proceed properly.

To adjust the control gains:

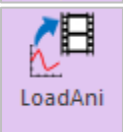
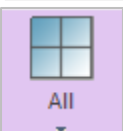
1. Double-click the **Kp** block and change the gain to **1000**. In addition, change **Kd** to **800**.
2. Run the simulation by clicking the **Start** tool.

Notice that the plots shown in the scopes did not change. These plots only update at the beginning of a simulation so to view the results from the current simulation, you need to run the simulation again. Then, the plots will be updated.

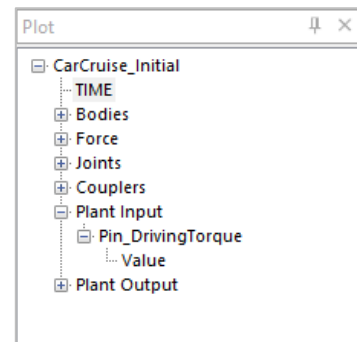
The system behavior now looks acceptable. Next, you will go back to the RecurDyn plot environment and plot the results.

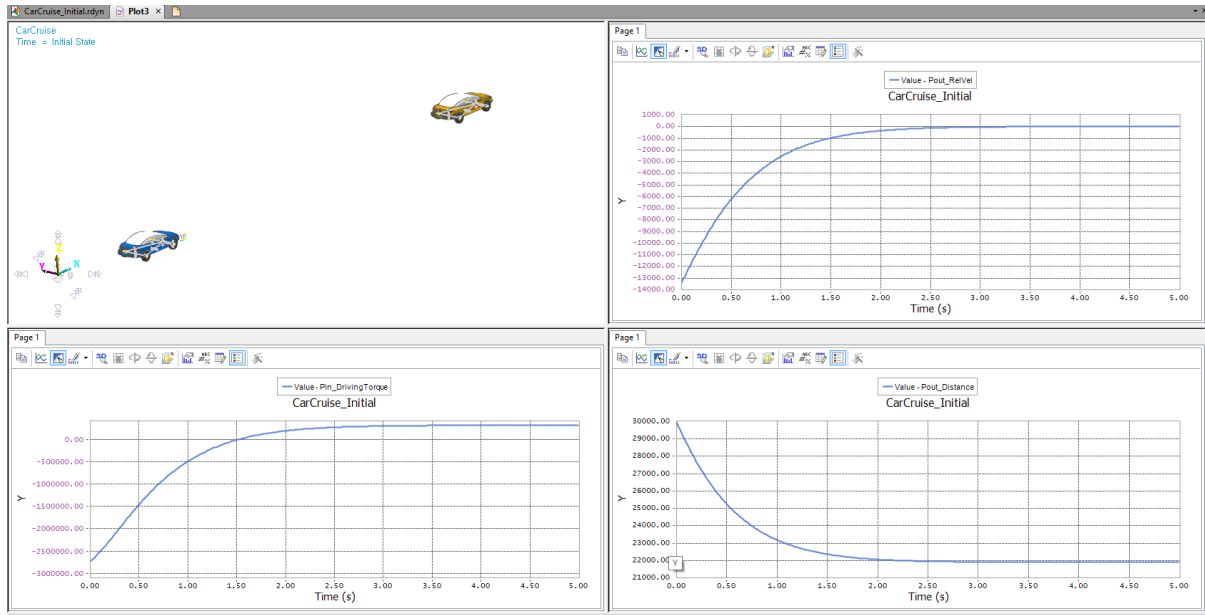
Viewing the Results

In this section, you will plot the driving torque, relative velocity, the distance between the cars, and include the animation.

To view the results:

1. Go back to the RecurDyn program, leaving the CoLink program open in case you want to explore your model further.
2. Open the **Plot window**.
3. In the upper row of tools in the toolbar, click the **Show All Windows** tool to divide the window into four panes.
4. Click the top left pane to activate it.
5. From **Animation** group in the **Tool** tab, load the animation by clicking the **Load Animation**.
6. A warning appears about deleting the view in that pane. Because there is nothing in the plot, click **Yes** to proceed.
7. Modify the view and rendering mode of the animation using the same commands and tools you would normally use in the modeling environment.
8. Click the bottom left pane to activate it and draw the **Driving Torque** (see the figure on right).
9. Click the top right pane to activate it and draw **Relative Velocity**, the first Plant Output
10. Click the bottom right pane to activate it and draw the **Distance** between cars, the second Plant Output. The next figure shows the resulting plot.





You have now finished the main steps of this tutorial. Additional steps include adjusting the control gains to further improve performance or increasing the complexity in the controller by making it adaptive based on the actual distance between cars or time to impact. In the next section, you will implement an adaptive control system that resumes driving at the desired speed when the car in front has moved.

Chapter

6

Augmenting the CoLink Model

Task Objective

In this chapter, you will modify the CoLink model to include an adaptive cruise control. You will then simulate the system and make plots of the results.



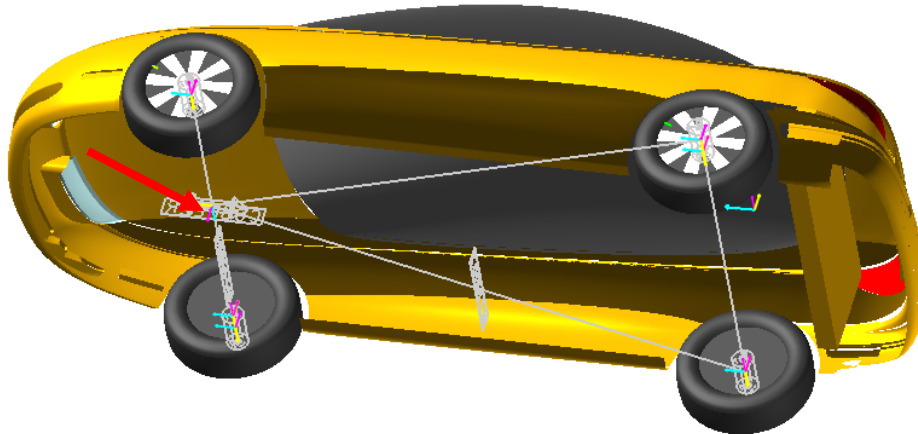
Estimated Time to Complete

15 minutes

Modifying the RecurDyn Model

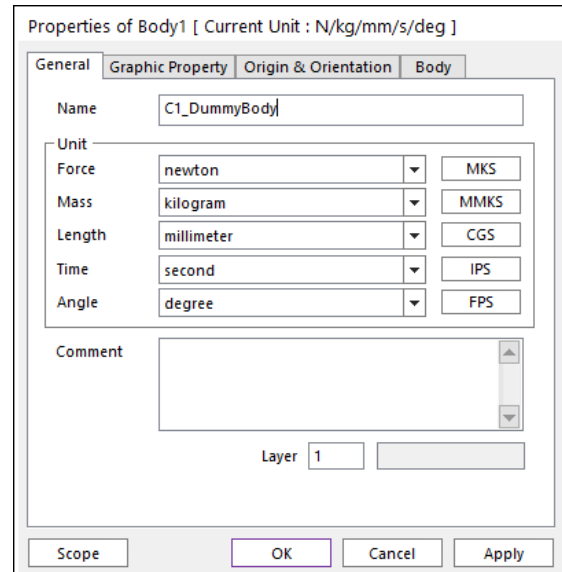
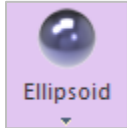
You will first modify the RecurDyn model to allow the implementation of the adaptive cruise control system. This will require an additional translational joint for the lateral movement of the front car and additional plant outputs to the control system from the model.

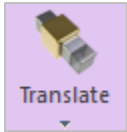
First, create a dummy body in the middle of the front axle of the front car.



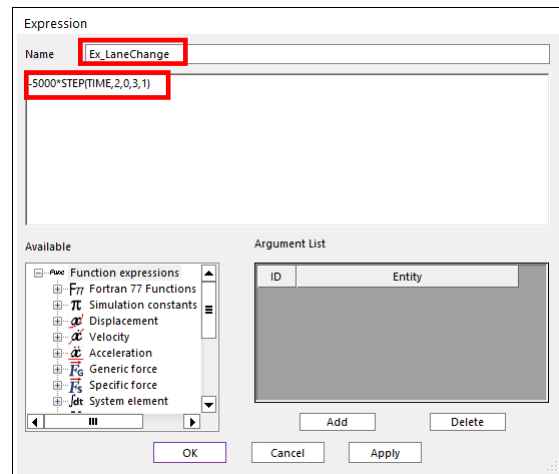
To create the dummy part:

1. From the **Professional** tab, click the **Ellipsoid** icon of the Body group.
2. Using the default **Point, Distance** creation method, select the markers at the existing translational joint and couplers of the front car, as shown in the following figure.
3. Enter **35** as the radius of the sphere.
4. Right-click the sphere and click **Properties**.
5. On the **General** page, change the name of the body to **C1_DummyBody** as shown in the figure to the right.
6. Click **OK**.



To create the translational joint:

1. From the **Joint** group in the **Professional** tab, click the **Translate** icon.
2. Using the **Body, Body, Point, Direction** creation method, and select:
 - Ground
 - C1_DummyBody
 - The center of the sphere
 - Enter the direction **0, 1, 0** (or click the **+Y-axis** of **Ground InertiaMarker**.)
3. Change the properties of the newly created joint:
 - Right-click the newly created joint and click **Property**.
 - On the **General** page, change its name to **C1_TraLaneChange**.
 - On the **Joint** page, create an expression for the lane change:
 - Check the box next to **Include Motion** and click **Motion**.
 - Click **EL** to open the expression window and then click **Create** to create a new expression.
 - **Name** the expression **Ex_LaneChange** and enter the following as the expression:
-5000*STEP(TIME,2,0,3,1)



This expression will move the car 5 meters to the right (the negative Y-direction) starting 2 seconds into the simulation and ending at 3 seconds

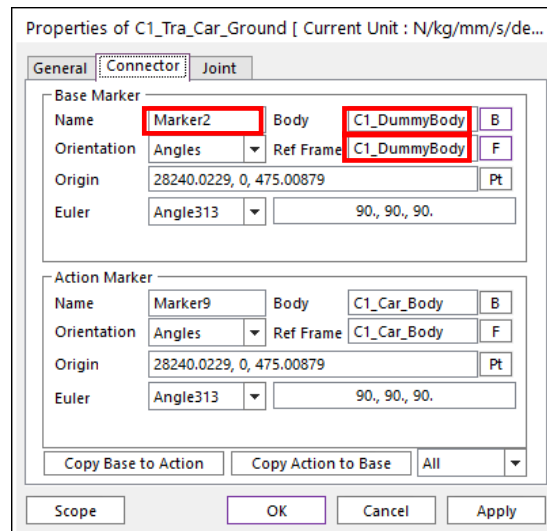
The Expression dialog box should look like the one on the right.

- Click **OK** four times to accept all changes and exit the dialog boxes.

Now you will modify the original translational joint so that it connects the car body (**C1_Car_Body**) to the sphere (**C1_DummyBody**) instead of ground.

4. In the Database window, right-click **C1_Tra_Car_Ground** and select **Property**.

- On the **Connector** page, change the **Base Marker Body** and **Ref Frame** to **C1_DummyBody** and change **Name** to **Marker2** because it is the next available one in the list of Markers associated with the dummy body as shown in the Database window in the figure below.



- Run the simulation to verify that it is working properly.

The five-second simulation should run as before with the exception that now the lead car should change lanes halfway through the simulation.

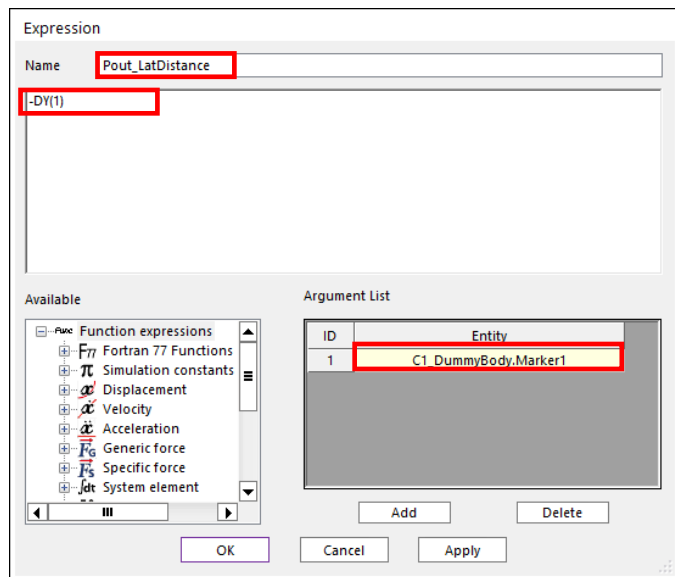
Next, you will define the additional plant outputs so the **CoLink** control system can be augmented to include them.

To define the plant outputs:

1. Double-click an existing plant output to display the Plant Output List dialog box.

2. Click **Add** create a new output.

3. In the Expression dialog box, add an entity to the Argument list by clicking **Add** and then either typing in **C1_DummyBody.Marker1** or dragging it in from the Database window.



4. Change **Name** to **Pout_LatDistance** and enter the expression as shown below, where **1** refers to the entry in the Argument List:

-DY(1)

5. Click **OK**.

6. In the Plant Output List dialog box, click **Add** again to enter one more plant output.

7. Name the expression **Pout_AbsVel**, enter **Car_Body.Marker5** in the Argument List, and enter the expression **VX(1)**.

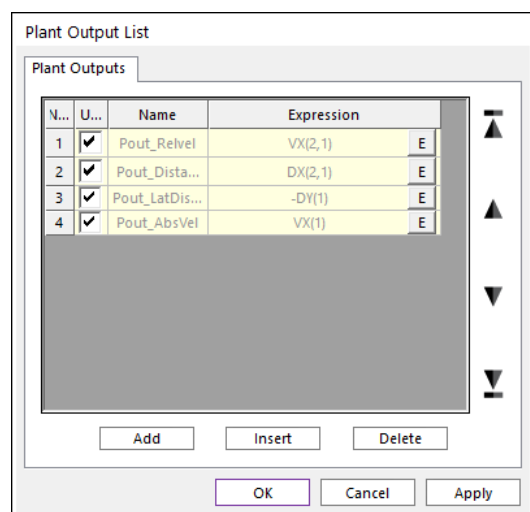
Marker5 on **Car_Body** is the marker associated with the translational joint **Tra_Car_Ground**.

Tip: You may need to enter a different marker number if you took extra steps (for example, created joints erroneously then deleted and created them again) and your numbering is different than that shown here.

Find the correct marker by expanding the entry for **Tra_Car_Ground** in the Database window and looking for the marker associated with **Car_Body**.

8. Click **OK** twice to accept changes and exit the dialog boxes.

9. The Plant Output List should look like the one shown on the right. Then click **OK** again.

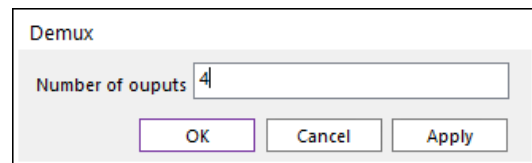


Augmenting the CoLink Model

You will now modify the CoLink model to implement the adaptive cruise control system. This will require a switch based on lateral movement of the front car and an additional feedback control system to maintain desired speed once the front car has moved out of the way.

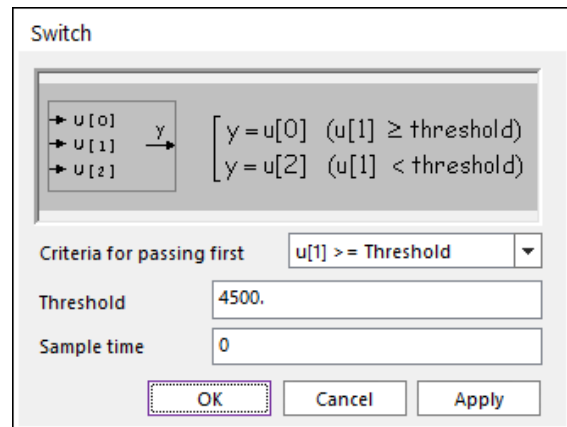
To modify the CoLink model:

1. Display the CoLink model window of the model that you created previously.
2. Save the CoLink model as a new file by clicking the clicking the **Save As** in the **File** menu and entering a new name.
3. Double-click the **Demux** block and change the number of outputs to **4**.
4. In the **Math** tab, click the **Switch** block of the Nonlinear group and then, click into the block diagram.
5. Double-click the **Switch** block and change the **Threshold** value to **4500**.

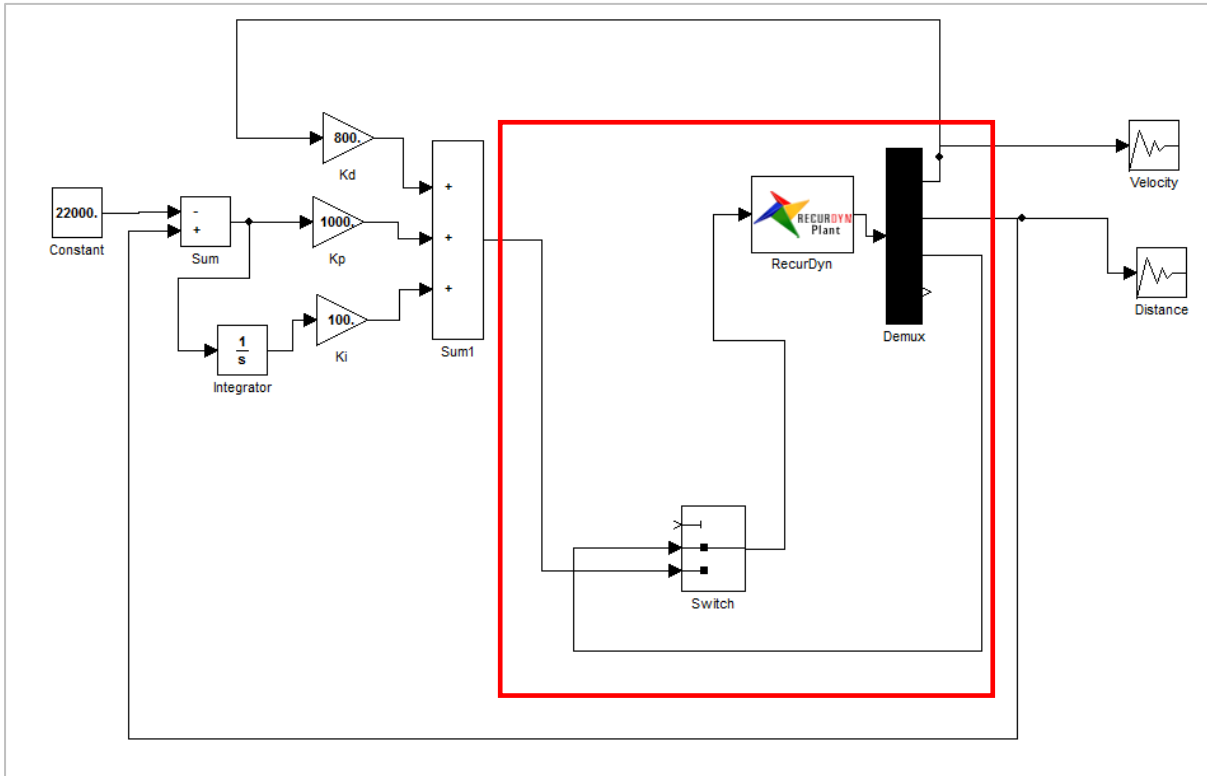


Setting the Threshold to 4500 means that when the second input, **u[1]**, is greater than or equal to 4.5 meters, the first input, **u[0]**, will be passed to the output of the block, **y**. This is why:

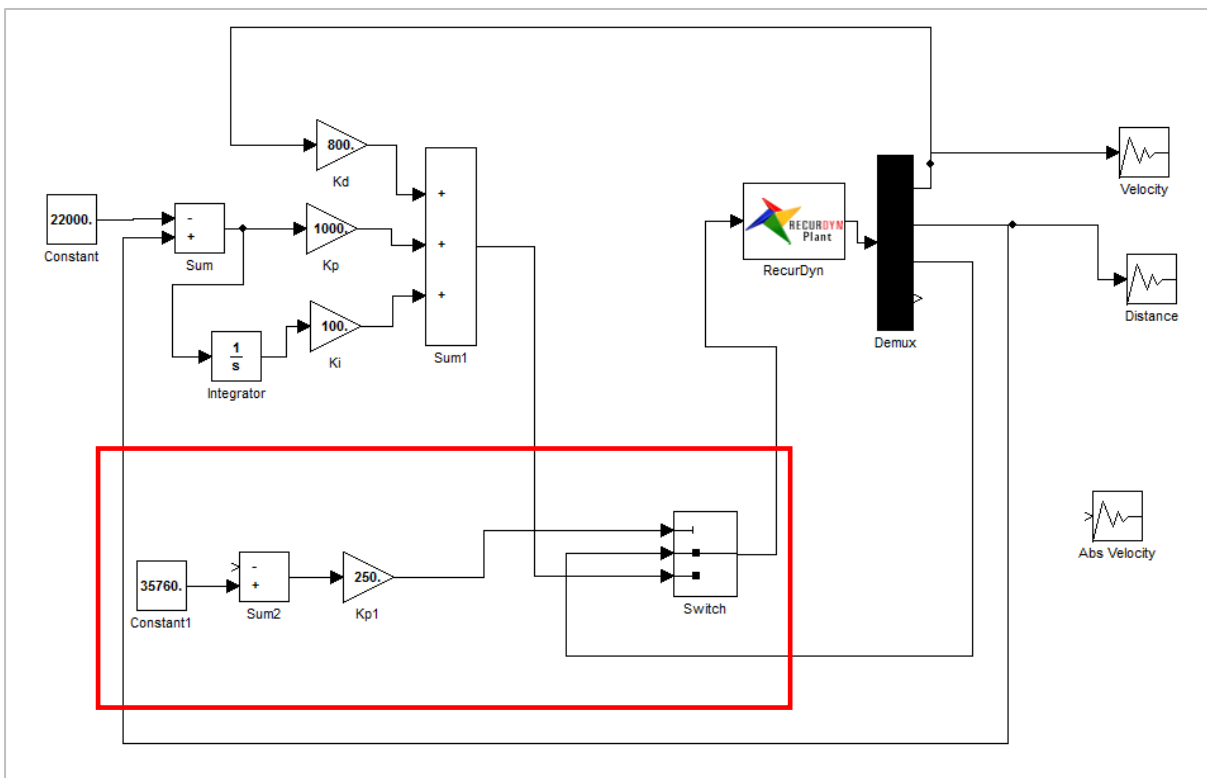
- **u[1]** is set as the lateral distance between the two cars in the simulation.
- **u[2]** is the control signal for the original system when the two cars are in the same lane.
- **u[0]** is the control signal for the new control system that sets a desired driving speed.



6. Using the figure below as guidance, do the following:
 - Delete the connection between **Sum1** block and the **RecurDyn** block and, connect the **Sum1** block to the bottom port of the **Switch** block.
 - Connect the output of the **Switch** to the **RecurDyn** block and the third port on the **Demux** block to the second port of the **Switch** block.

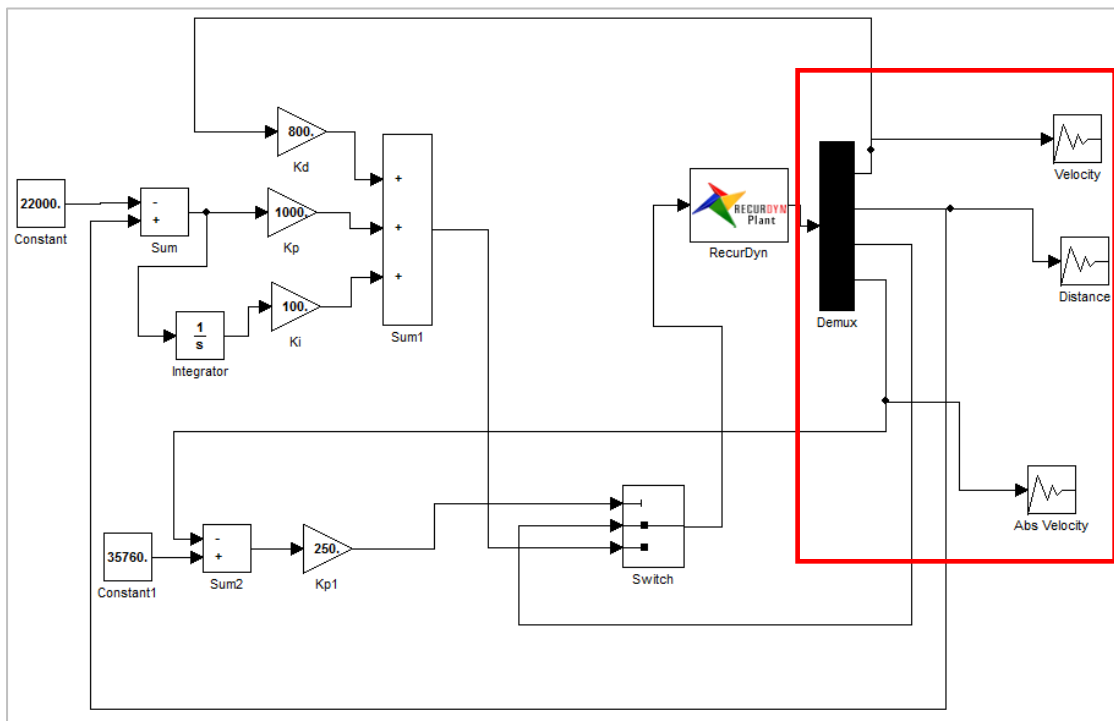


7. Now, create a new constant/sum block, highlighted in the figure below:



Click the **Sum** block and press **Ctrl+C** to copy the block. Then, press **Ctrl-V** to paste a copy and drag it down to the lower section of the block diagram to the left of the **Switch** block.

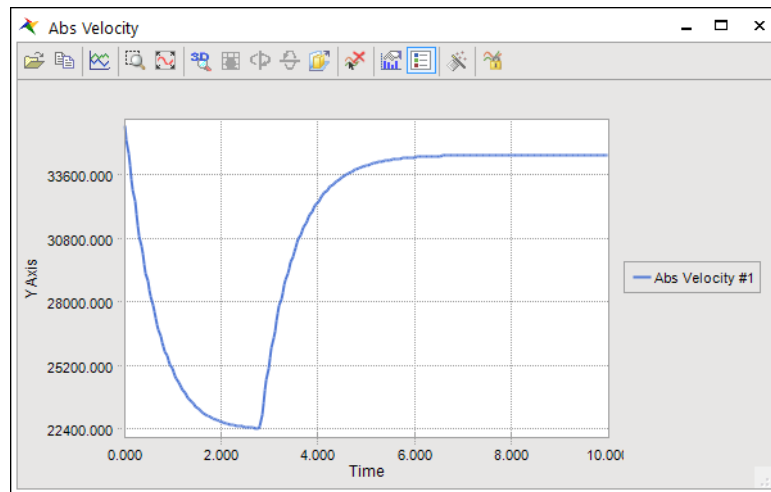
- a. Change the signs of the new **Sum2** block to $-+$ so it appears as shown above.
 - b. Connect the fourth output port from the **Demux** block to the negative terminal on the new summation block, **Sum2**.
 - c. Similarly, copy the **Constant** block and the **Kp** block and place them to the left and right of the **Sum2** block. Then, do the following:
 - Change the constant value to **35760** (mm/s, or 80 mph).
 - Connect the **Constant1** block to the positive port on **Sum2**
 - Connect the output of **Sum2** to the input of **Kp1**.
 - Change the gain for **Kp1** to **250** and connect its output to the first input port on the **Switch** block.
8. Add a scope to the fourth output from the **Demux** block and label it **AbsVelocity** because it represents the actual forward speed of the blue car, as highlighted in the figure below.



9. Run a simulation of your system for 10 seconds to verify that it is working correctly.

Tip: Change the simulation time by either:
 In the toolbar bar next to the animation controls, entering **10**.
 Clicking **Simulation** → **Parameter** and changing **End Time** to **10**.

The velocity of the second car should match the following plot

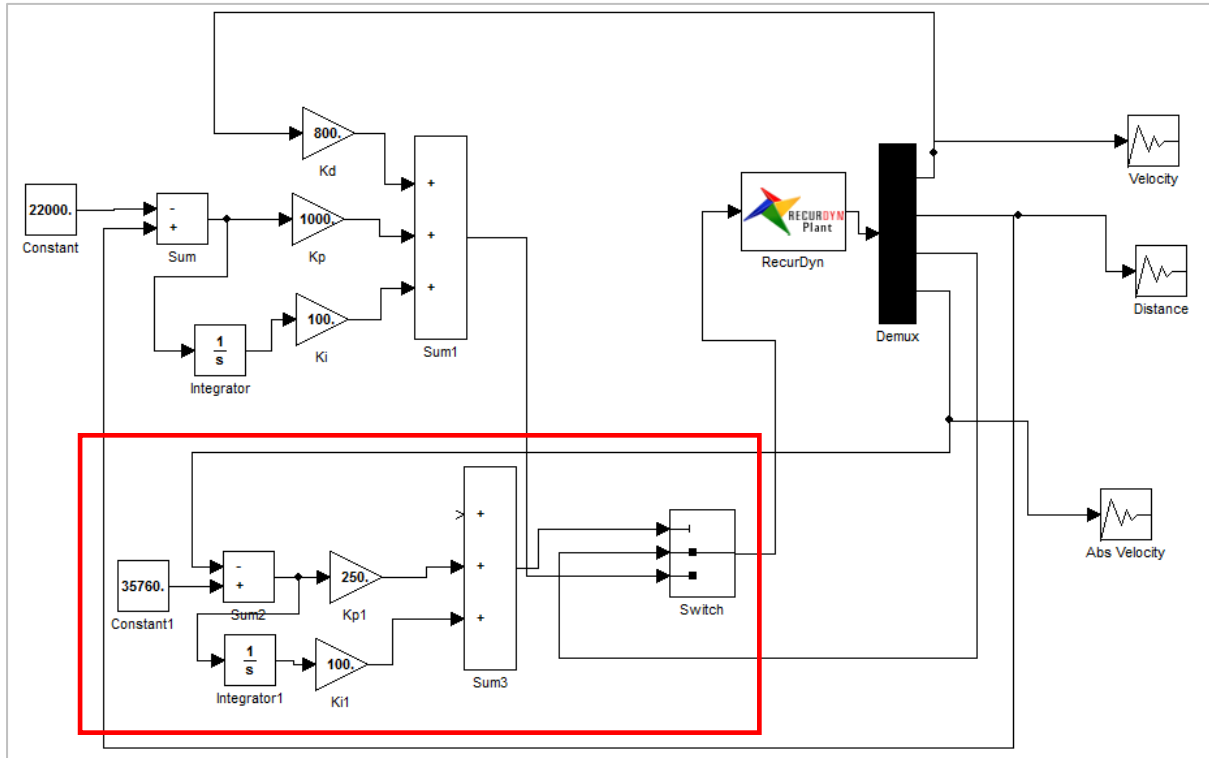


Note that the forward speed does not ever reach the value at the beginning of the simulation. To rectify this, you will add integral control to the control system. You will also add derivative control and a rate limiter to control the rapid acceleration 3 seconds into the simulation.

To add integral and derivative control, highlighted in the figure below:

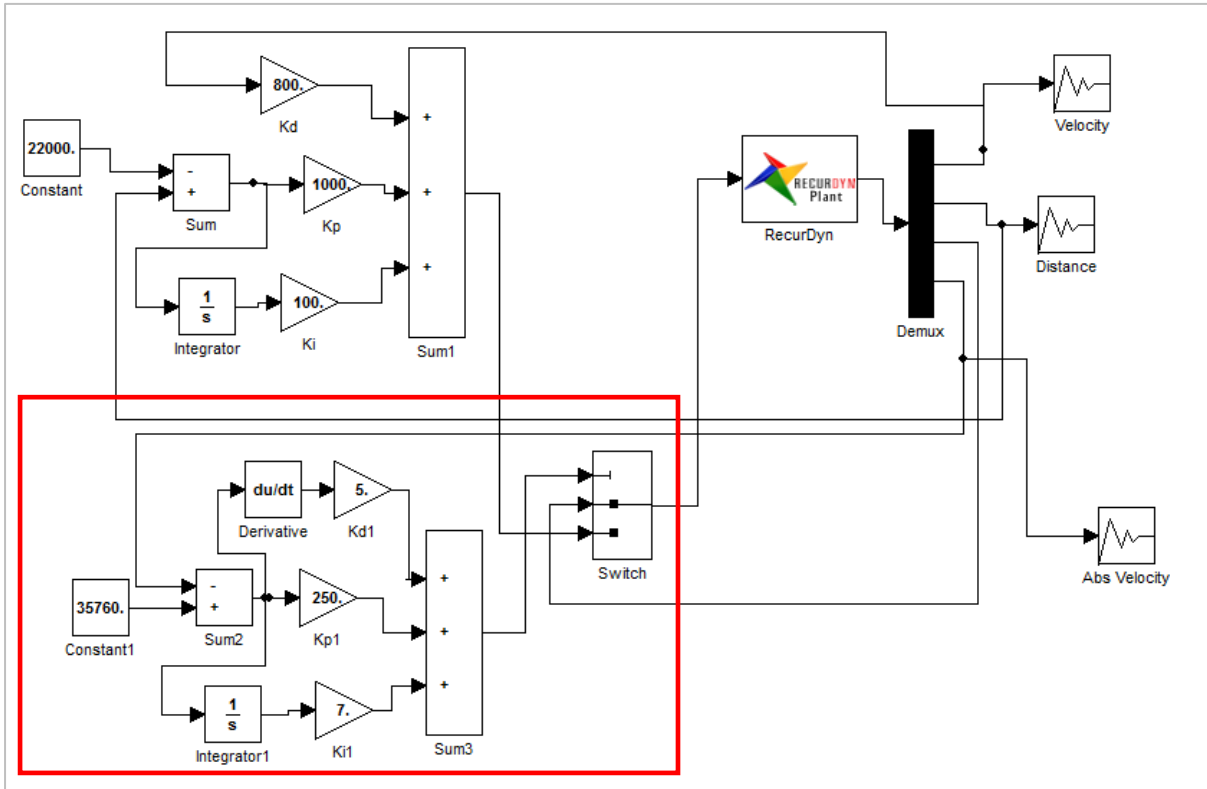
1. Delete the connector between the **Kp1** block and the **Switch**.
2. Copy the **Sum1** block and paste it just to the right of **Kp1**.
3. Change the signs of the new **Sum3** block to **+++** so it appears as shown below.
4. Connect **Kp1** to the second port of the new summer, **Sum3**.
5. Connect the output of **Sum3** to the open first port on the **Switch** block.
6. Copy the **Integrator** and **Ki** blocks from the original control system and paste them into the augmented portion in the same place. Connect the blocks in the same manner in which they were previously done.

When this step is complete your model should look like the following:



7. In the **Continuous and Discrete** tab, click the **Derivative** block of the Continuous group and then, click into the control system and place it above the **Sum2** block.
8. Make a copy of the **Kd** block and paste it above the **Kp1** block.
9. Connect the following:
 - The signal running between **Sum2** and **Kp1** to the **Derivative** block.
 - The **Derivative** block to the **Kd1** block.
 - The **Kd1** block to the first input port on the **Sum3** block.
10. Change the gains for **Kd1** and **Ki1** to **5** and **7**, respectively.

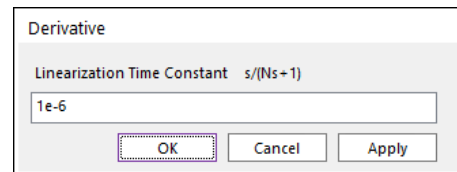
Your model should now look like this:

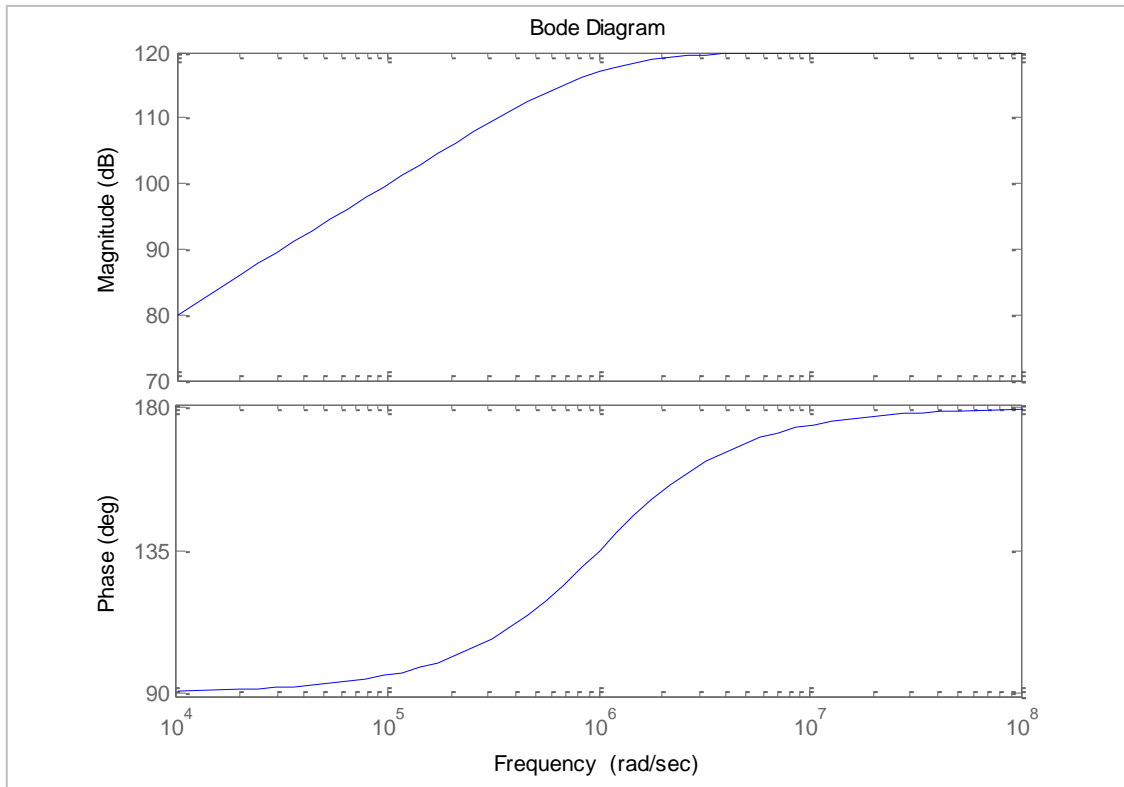


- Double-click the **Derivative** block and change the **Linearization Time Constant** to **-1e-6**.

As seen in the Derivative dialog box, the transfer function for this differentiator is now $s/(-1e-6s + 1)$.

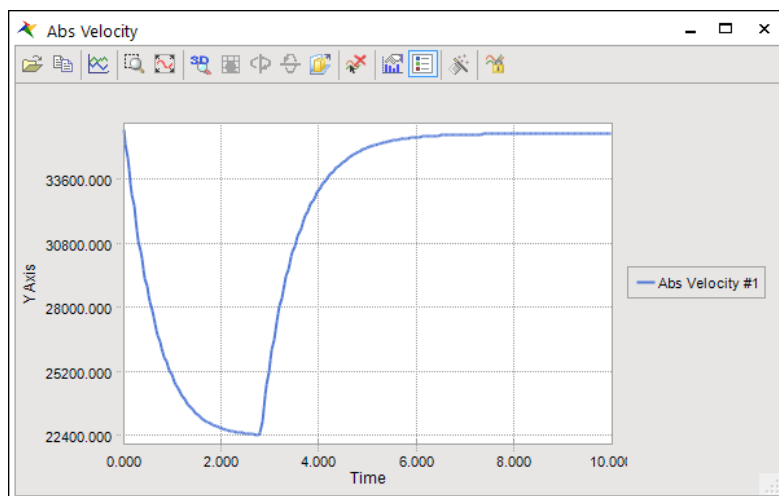
A perfect differentiator would have a transfer function of simply s , or $s/1$. This is not physically realizable though because at high frequencies, the gain would be infinite. Instead, we choose the Linearization Time Constant, so the block approximates a differentiator for frequencies less than about $1e6$ rad/s, as shown in the following Bode diagram:





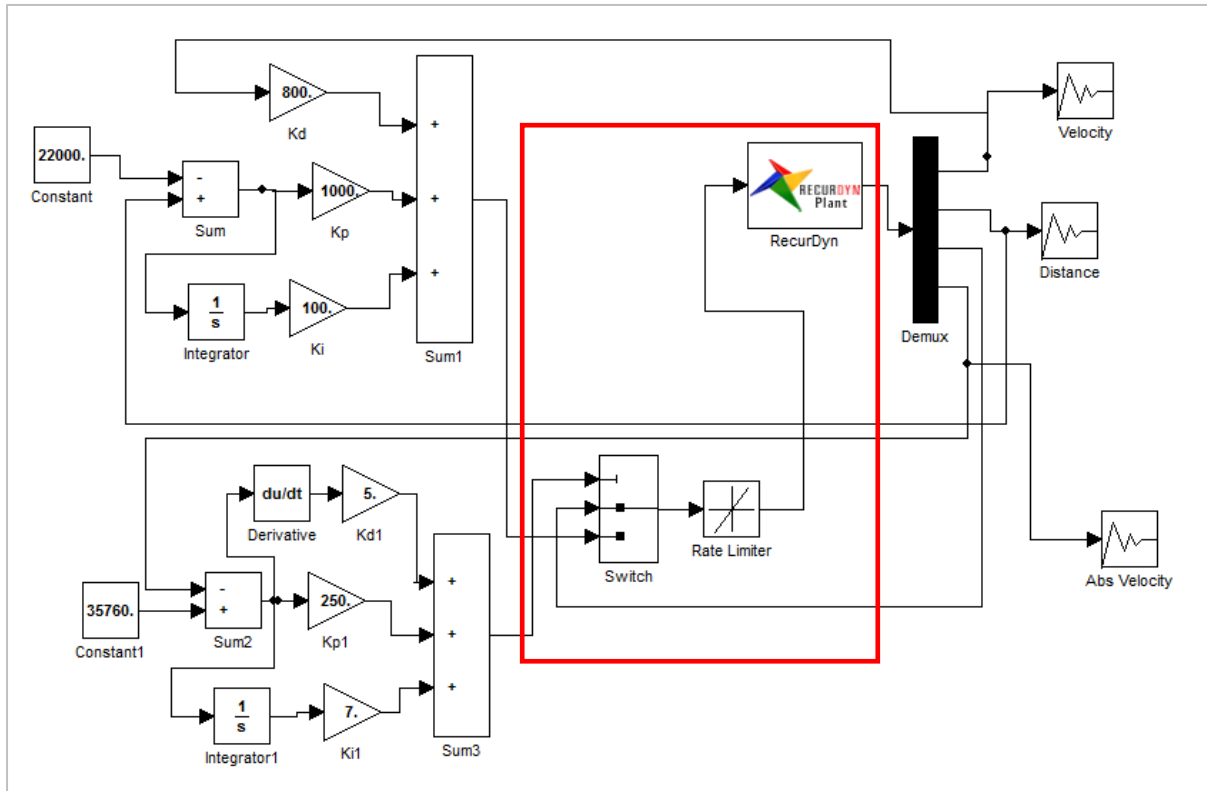
12. Simulate the system again to verify that the system is working properly, and that the final velocity is now the desired value. The car velocity should appear as shown on the right.

Next, you will add a rate limiter to the control system, so the rate of acceleration is not so high. Specifically, as soon as the front car moves over, the back one changes suddenly from a steady speed to a rapidly increasing velocity. This would feel like the car is flooring the gas pedal and would be uncomfortable for passengers. The rapid decrease in velocity is not a problem because the car needs to maintain the ability to slow down quickly if an object is in its way.

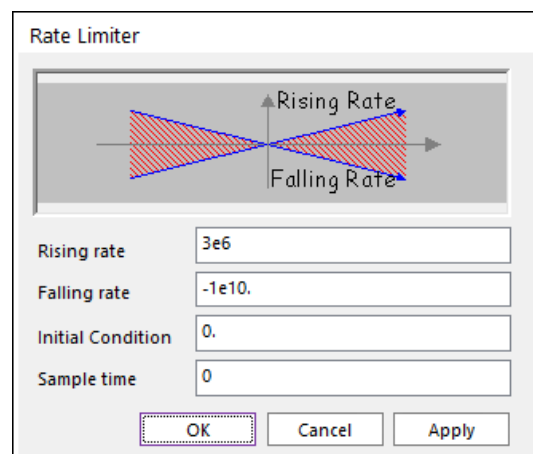


To add the rate limiter, highlighted in the figure below:

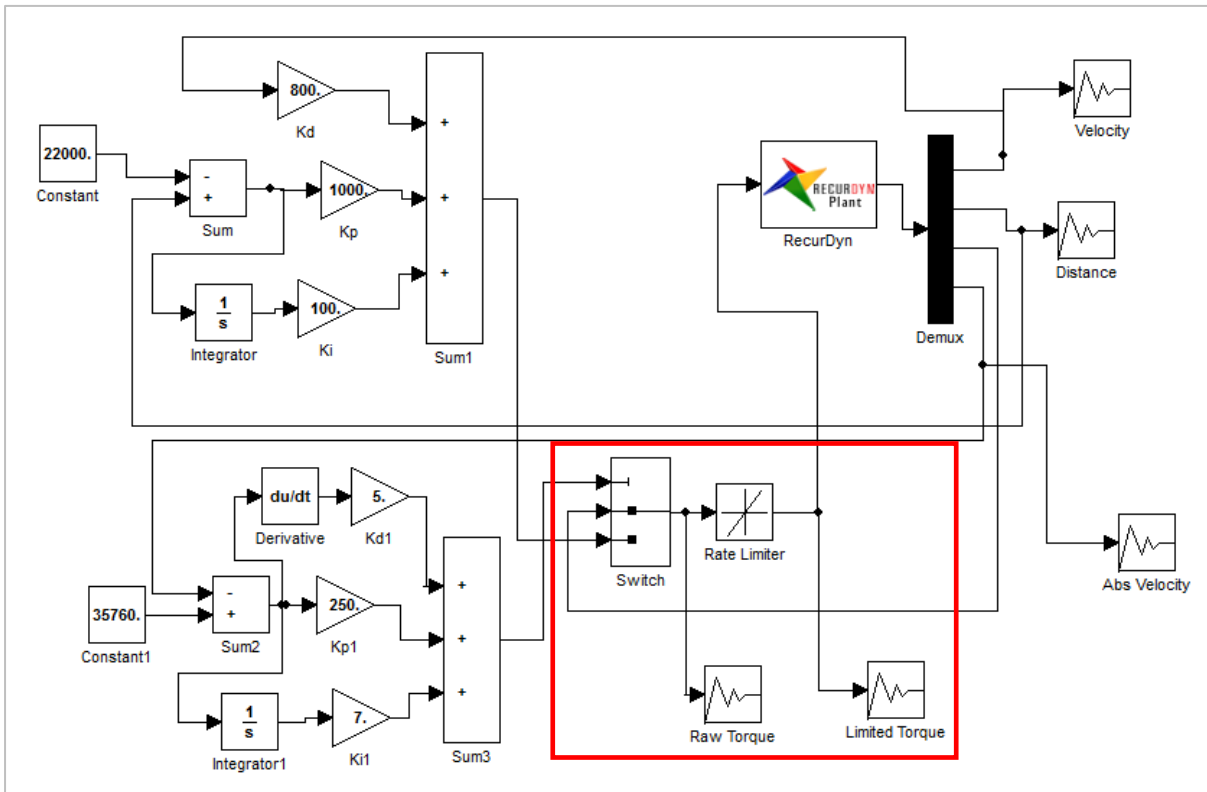
1. Delete the connector between the **Switch** and **RecurDyn** blocks.
2. In the **Math** tab, click the **Rate Limiter** block of the **Nonlinear** group and then, click into the model and place it to the right of the **Switch** block.
3. Connect the **Switch** to the **Rate Limiter** and the **Rate Limiter** to the **RecurDyn** block as shown below.



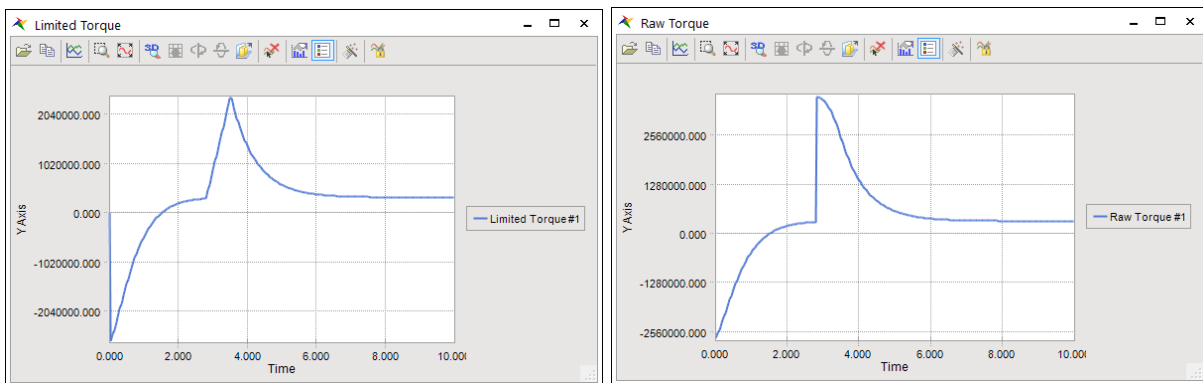
4. Double-click the **Rate Limiter** and change the **Rising rate** to **3e6** and the **Falling rate** to **-1e10**. Leave the initial conditions as they are.



- To see the effect of the rate limiter, add two scopes to the model. Connect one of them before the rate limiter and the other one after it.. Change the name of **Scope** and the name of **Scope1** to **RawTorque** and **LimitedTorque**.

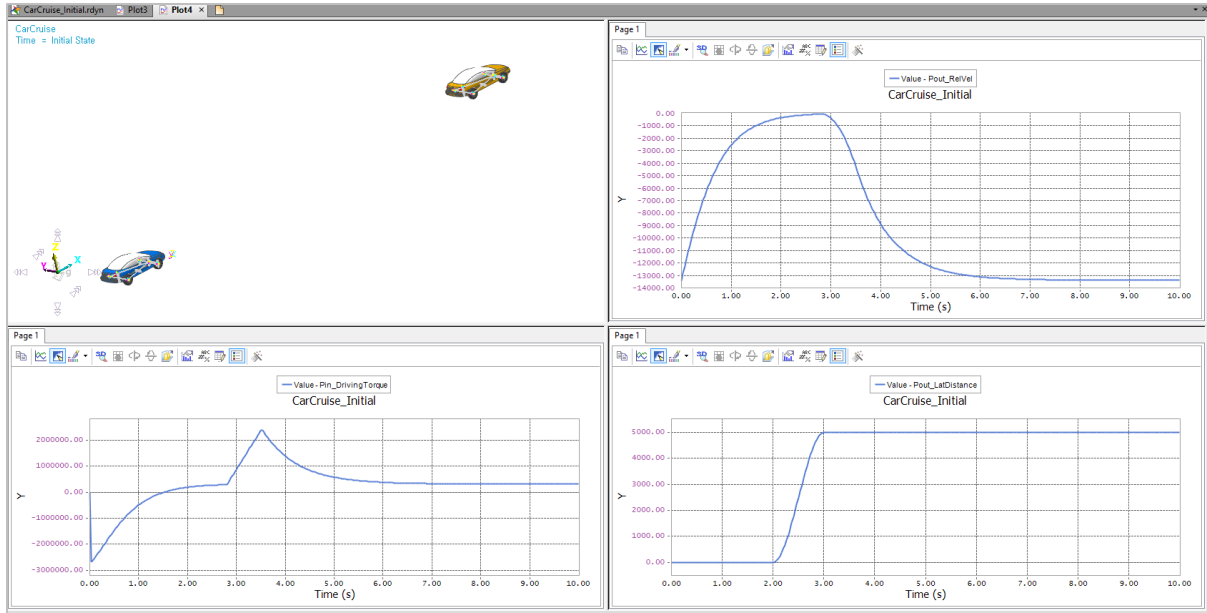


- Simulate your model again, double-click the two new scopes, and look at the results. They should appear as follow.



- Display the RecurDyn model, press the **Play** button on the Animation Control group in the Analysis tab, and watch the simulation.

When you are satisfied that your model is working properly, display the plotting environment and make the same plot as you did at the end of Chapter 5, but rather than plotting the distance between cars in the lower-right pane, plot the lateral distance, **Pout_LatDistance**. This plot is shown below.



If you are interested in investigating this model further, try these additional steps:

- Adjust the control gains to improve performance even more.
- Set **Kd** and **Ki** to zero and tune the controller from scratch.

Thanks for participating in this tutorial!