Basic ADAMS Full Simulation Training Guide

VERSION 11.0

PART NUMBER
110VIEWTR-03

Visit us at: www.adams.com

CONTENTS

Contents...

Contents...

Contents...

Contents...

WELCOME TO BASIC ADAMS TRAINING

ADAMS Full Simulation Package is a powerful modeling and simulating environment that lets you build, simulate, refine, and ultimately optimize any mechanical system, from automobiles and trains to VCRs and backhoes.

Basic ADAMS Full Simulation Package training teaches you how to build, simulate, and refine a mechanical system using Mechanical Dynamics, Inc.'s ADAMS Full Simulation Package.

What's in this section:

About Mechanical Dynamics

Find a list of ADAMS products at:

- http://www.adams.com/mdi/product/modules.htm

Learn about the ADAMS—CAD/CAM/CAE integration at:

- http://www.adams.com/mdi/product/partner.htm

Find additional training at:

- http://support.adams.com/training/training.html
- Or your local support center

After taking this course you will be able to:

- Build ADAMS models of moderate complexity.
- Understand ADAMS product nomenclature and terminology.
- Understand basic modeling principles and extend your proficiency by creating progressively more complex models.
- Use the crawl-walk-run approach to virtual prototyping.
- Debug your models for the most common modeling challenges (for example, redundant constraints, zero masses, and so on).
- Use and be informed about all methods of ADAMS product support.
- Use the product documentation optimally.

Organization of guide

This guide is organized into modules that get progressively more complex. Each module focuses on solving an engineering-based problem and covers mechanical system simulation (MSS) concepts that will help you use ADAMS most optimally. The earlier workshops provide you with more step-by-step procedures and guidance, while the later ones provide you with less.

Each module is divided into the following sections:

1. Problem statement
2. Concepts
3. Workshop
4. Module review

Online guides

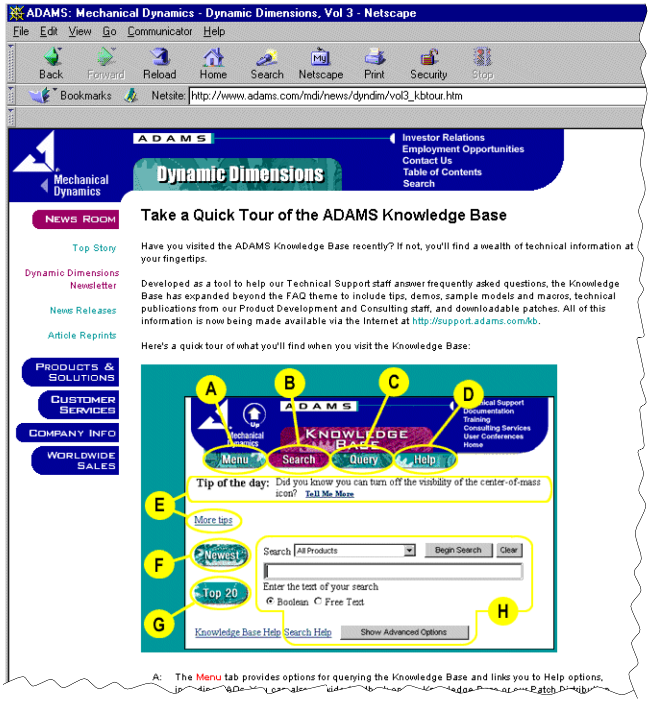
Access help on help from

- **Help** menu of any ADAMS product
- **Help** tool  on the Documentation Road Map

Knowledge base

Go to http://support.adams.com/kb

For a quick tour, go to http://www.adams.com/mdi/news/dyndim/vol3_kbtour.htm



Ask email-based users group

Go to http://support.adams.com/support/tech_ask.html

Consulting services

Go to http://support.adams.com/support/cnsltsrv.html

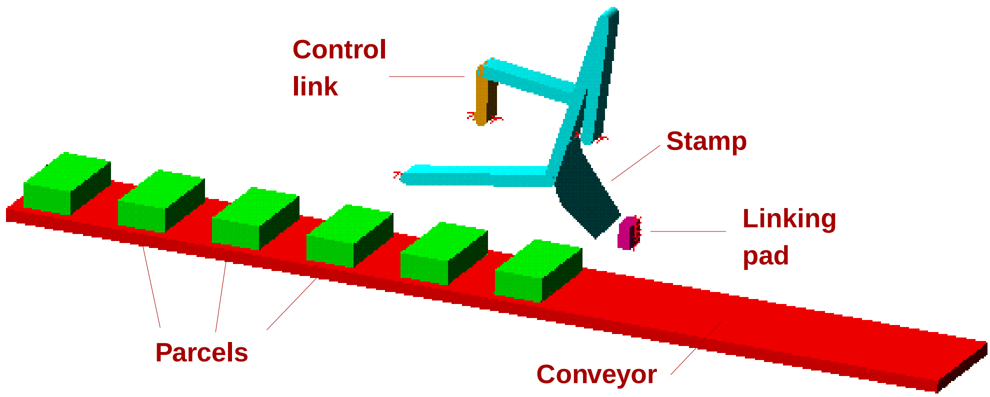
Technical support

To find your support center, go to http://support.adams.com/support/suppcent.html

To read the Service Level Agreement, go to http://support.adams.com/support/sla_agree.html

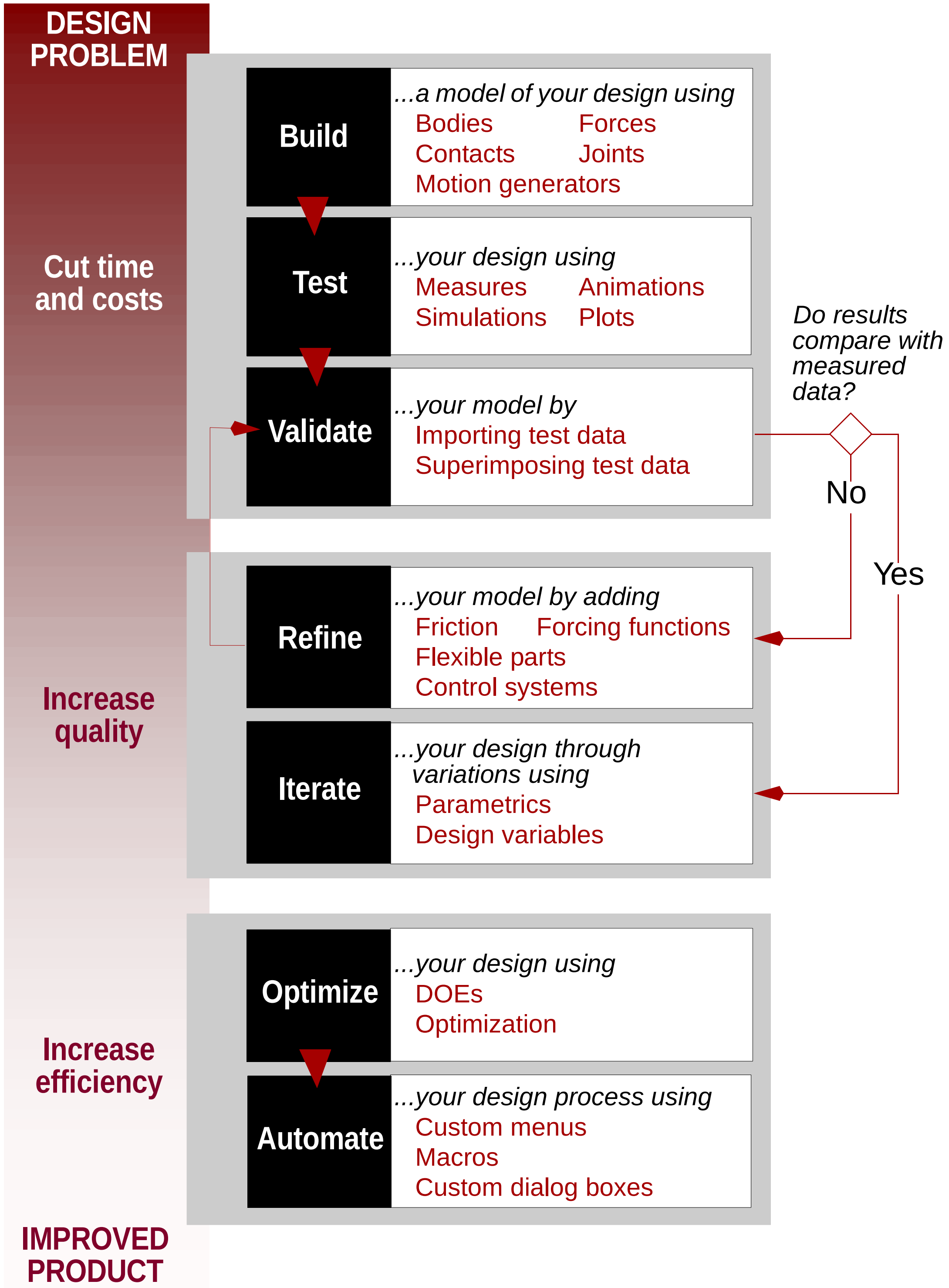
1 STAMPING MECHANISM

Understand the virtual prototyping process by improving the design of the stamping mechanism shown next:



What's in this module:

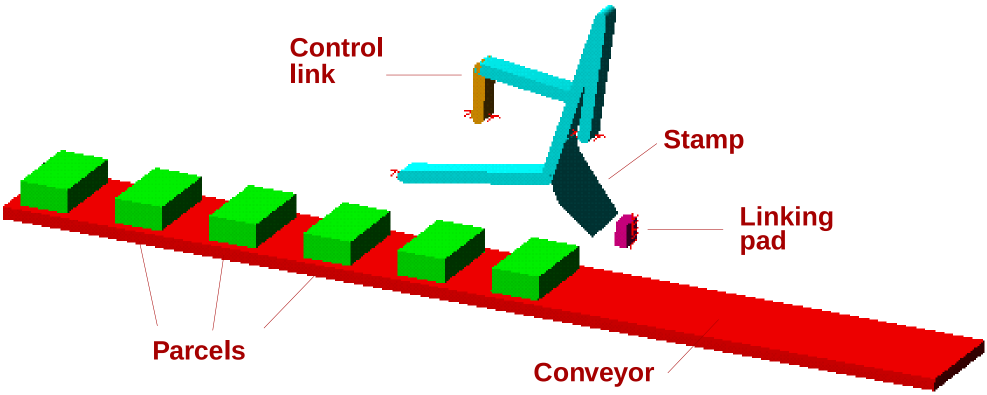
Virtual Prototyping Process



Workshop 1—Stamping Mechanism

Problem statement

Understand the virtual prototyping process by improving the design of the stamping mechanism shown next:



Model description

- This model represents a mechanism for stamping parcels that are moving along a conveyor belt.
- During the work cycle, the stamp does not contact the parcels that it is supposed to label.
- To fix this design flaw, modify the length of the control link.

Start the workshop

Your first step will be to start ADAMS/View from the directory *exercise_dir*/mod_01_stamper. When you start ADAMS/View from that directory, it automatically builds the model stamp and a fully customized version of ADAMS/View.

There are separate instructions for starting ADAMS/View in UNIX and Windows NT. Follow the set of instructions below depending on the platform you are using.

If you are using UNIX, to start ADAMS/View:

1. At the command prompt, enter the command to start the ADAMS Toolbar, and then press **Enter**. The standard command that MDI provides is adams*x*, where *x* is the version number, for example adams11, which represents ADAMS 11.
2. From the ADAMS toolbar, right-click the **ADAMS/View** tool .
3. Select **Change Settings for A/View**.

   The Change Settings for A/View dialog box appears.
4. Select **Working directory**.
5. Right-click the **Working directory** text box, and then select **Select a Directory**.

   The Select a Directory dialog box appears.
6. Select the directory **mod_01_stamper** (*exercise_dir*/mod_01_stamper).
7. Select **OK**.
8. From the **Change Settings for A/View** dialog box, select **OK**.
9. From the ADAMS toolbar, select the **ADAMS/View** tool.

   ADAMS/View starts and automatically imports the commands to build:

   - Model named stamp.
   - Fully customized version of ADAMS/View.

   After importing the commands, an Information window appears.
10. Read the information describing the model, and then, in the upper right corner, select **Close**.

If you are using Windows NT, to start ADAMS/View:

1. From the **Start** menu, point to **Programs**, point to **ADAMS 11.0**, point to **AView**, and then select **ADAMS - View.**

   ADAMS/View starts and the Welcome dialog box appears.

2. From the Welcome dialog box, select **Import a file**.
3. Click the file folder.

   The Find Directory dialog box appears.

4. Find and select the directory **mod_01_stamper** (*exercise_dir*/mod_01_stamper).
5. Select **OK.**

   The File Import dialog box appears.

6. Set **File Type** to **ADAMS/View command file (*.cmd)**.
7. Right-click the **File to read** text box, and then select **Browse**.

   The Select File dialog box appears.

8. Select the file, **aview.cmd**, and then select **Open**.
9. Select **OK.**

   ADAMS/View imports the commands to build:

   - Model named stamp.
   - Fully customized version of ADAMS/View.

   After importing the commands, an Information window appears.

10. Read the information describing the model, and then, in the upper right corner, select **Close**.

Change the model

In this section, you see how you can change the length of the control link (control_link).

To change the model:

1 From the **Stamper** menu, select **Setting Up Model**.

The Stamper_Setup dialog box appears.

2 Use the arrow buttons to modify the length of the control_link.

- The buttons shift the location of the top of the control_link upward and downward 3 mm at a time.
- The parts connected to the control_link are parameterized in such a way as to move the appropriate amount automatically whenever you adjust the length of control_link.

3 Watch the model change as you press these buttons.

4 To reset your model to the original configuration, select **Reset**.

Leave the Stamper_Setup dialog box open and continue with the next step.

Simulate the model

Now, you'll simulate the model to see how it behaves.

To simulate the model:

1 From the **Stamper** menu, select **Simulate**.

The Stamper_Simulate dialog box appears.

2 To simulate the current design variation, ensure that **Single** is selected.

3 To solve the equations of motion for the current design, select **Apply**.

Note: You selected to display the model at every output step. If you were to change **Model Update** from **At Every Output Step** to **Never**, the model would not update on the screen but solves faster.

When a single simulation is completed, ADAMS/View tells you what the penetration was during the simulation. A positive number indicates penetration.

4 To continue, select **OK**.

5 Leave the Stamper_Simulate dialog box open and continue with the next step.

Investigate results

Now you'll look at the results of the simulation as an animation and a plot.

To investigate results:

1. From the **Stamper** menu, select **Investigate Results**.

   The Stamper_Investigate dialog box appears.

2. To see the motion resulting from your last simulation, select **Animate Results**.

   If necessary, use the stop sign in the lower right corner of the window to stop an animation before it has completed.

3. To plot the vertical travel of the stamper with respect to the parcel tops versus time, as calculated from your last simulation, select **Measure Stamp Height Above Parcels**.

   A strip chart appears, in which ADAMS plots the height of the stamp above the parcels.

4. To save an existing curve so that the next simulation curve will not overwrite the existing curve, but will be superimposed on the saved curve, select **Save Curve**.

Manually find the correct height

Now change the model again to find the correct height at which the stamp makes minimal contact with the parcels.

To find the correct height:

- Repeat the steps on the previous page until you can identify the control_link length at which the stamp makes contact with the parcels, using 3 mm increments. Use this value to answer Question 1 in Module review on page 21.

  If stamp_height > 0, stamper does not make contact with parcels.

  If stamp_height < 0, stamper makes contact with parcels.

Perform a design study

Now you'll perform a design study. The design study automatically analyzes the model using the specified upper and lower limits for control_link length, and the specified number of runs. Default values are given, but you can modify them if desired.

To perform a design study:

1. On the Stamper_Simulate dialog box, select **Design Study**.
2. To speed up the simulation, set **Model Update** to **Never**.
3. Select **Apply** to submit the design study.

   The design study automatically analyzes the model and a strip chart and Information window appears when the study is complete.
4. From the Information window, identify the range of the control_link length values within which the stamp makes contact with the parcels. Use this range to answer Question 2 in Module review on page 21.
5. Close the Information window.

Perform an optimization study

Now, you'll perform an optimization study. During an optimization study, ADAMS/View systematically varies the control_link length and runs a number of simulations until the specified penetration is achieved to within a set tolerance.

To perform an optimization study:

1. On the Stamper_Simulate dialog box, select **Optimization**.
2. Set the **Desired Penetration** to **4 mm**.

   Note: Notice that ADAMS wraps the 4 mm in parentheses () to denote an expression. If you did not enter units, ADAMS uses the default units set for the model.
3. Set **Model Update** to **Never**.
4. Select **Apply** to submit the optimization study.

   The Information window appears, displaying the control_link length for a maximum penetration of 4.00.
5. From the displayed value of the control link length, note the maximum penetration. Use this value to answer Question 3 in Module review on page 21.

6 Select **OK**.

The value on the Stamper_Setup dialog box also updates to the optimized value.

7 Exit ADAMS/View:

- From the **File** menu, select **Exit**.
- From the dialog box that appears, select **Exit, don't Save**.

Module review

1 Using 3 mm increments, at what control link length do you first notice penetration?

2 From the design study, what control link length results in penetration? How does this compare with your previous results?

3 If you specify a maximum desired penetration of 4 mm, what is the optimal length of the control link? How close is the maximum actual penetration from the maximum desired penetration?

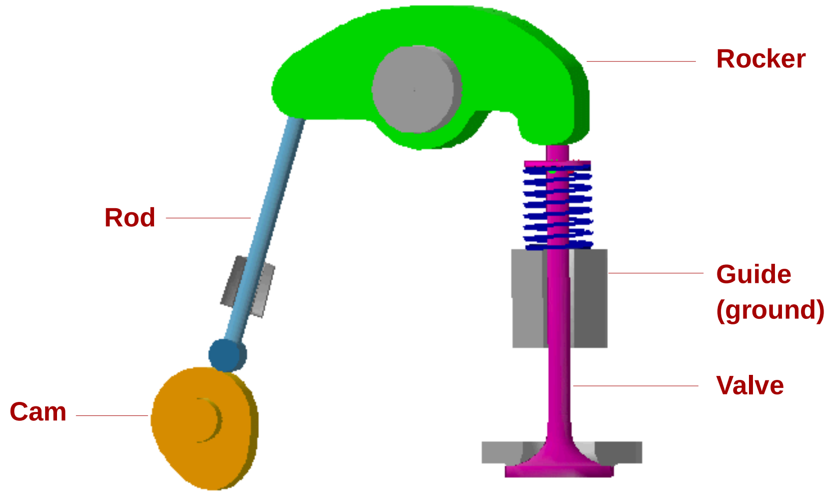
4 How many moveable parts does the model consist of?

5 How many joints does the model consist of?

6 What would happen if you deleted the conveyor belt?

2 ADAMS/VIEW INTERFACE OVERVIEW

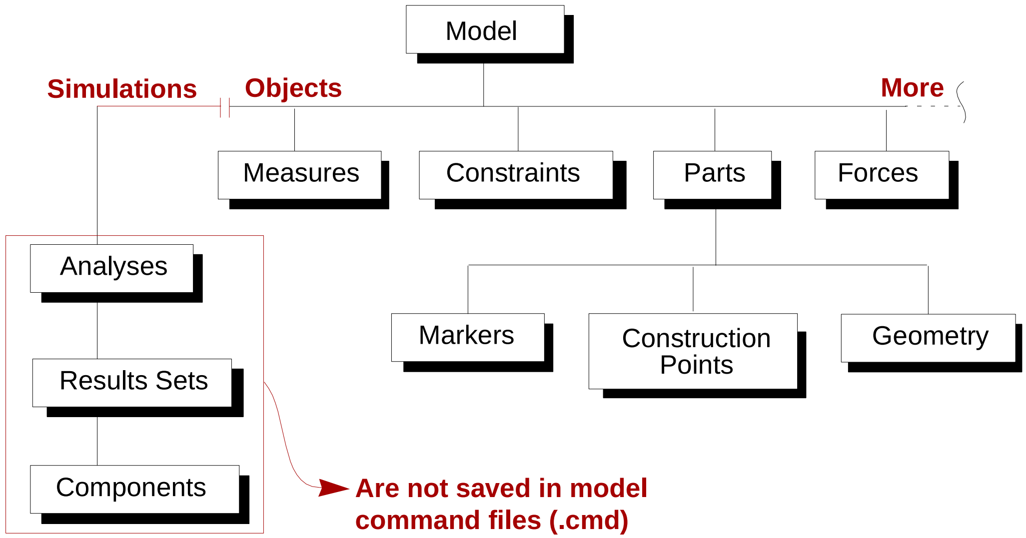
Use the ADAMS/View graphical-user interface (GUI) to manipulate, simulate, review, and refine the model shown next:



What's in this module:

ADAMS/View modeling hierarchy

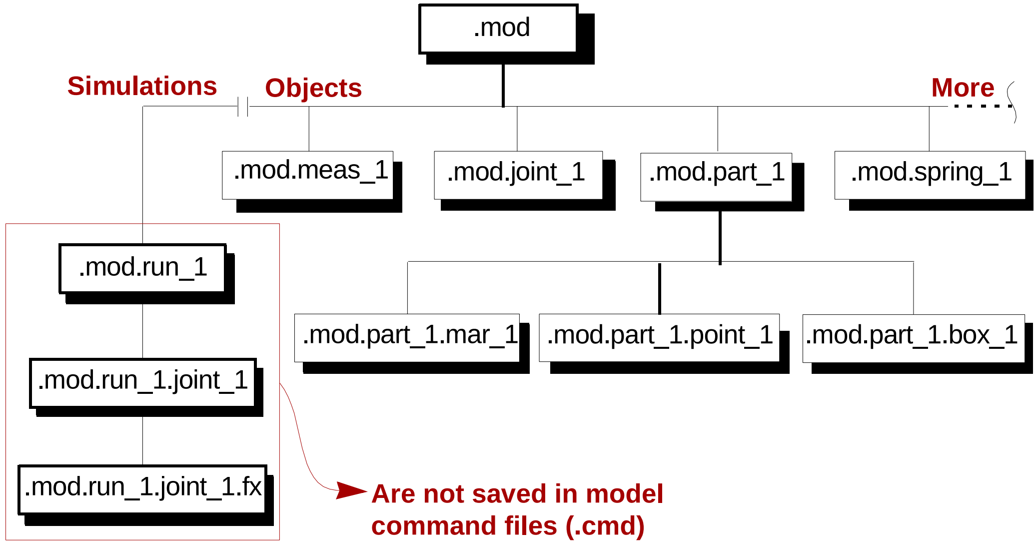
- ADAMS/View names objects based on this model hierarchy. For example, ADAMS/View names geometry as .model_name.part_name.geometry_name.
- To change the parent for an object, rename the object.



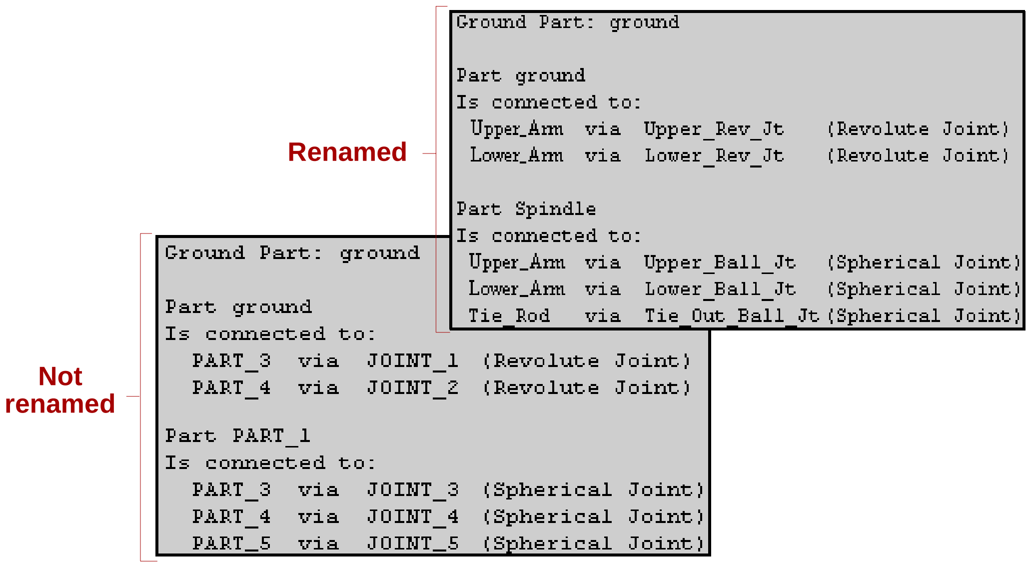
See also: Assembling Subsystem Models on page 182

Renaming Objects

ADAMS/View naming conventions



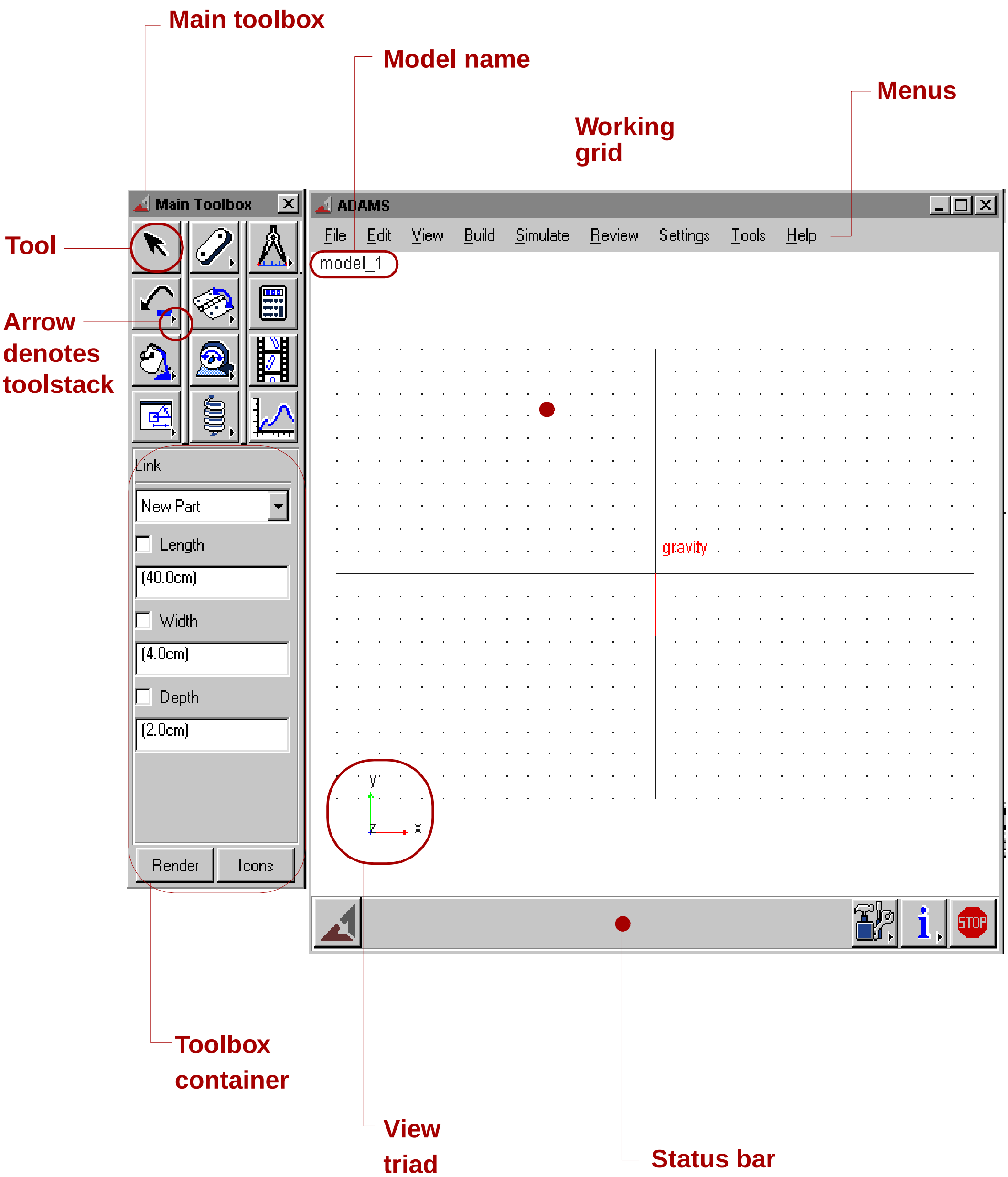
Renaming objects clarifies model topology as follows




```
Ground Part: ground

Part ground
Is connected to:
 Upper_Arm  via  Upper_Rev_Jt    (Revolute Joint)
 Lower_Arm  via  Lower_Rev_Jt    (Revolute Joint)

Part Spindle
Is connected to:
 Upper_Arm  via  Upper_Ball_Jt   (Spherical Joint)
 Lower_Arm  via  Lower_Ball_Jt   (Spherical Joint)
 Tie_Rod    via  Tie_Out_Ball_Jt (Spherical Joint)
```

```
Ground Part: ground

Part ground
Is connected to:
  PART_3  via  JOINT_1  (Revolute Joint)
  PART_4  via  JOINT_2  (Revolute Joint)

Part PART_1
Is connected to:
  PART_3  via  JOINT_3  (Spherical Joint)
  PART_4  via  JOINT_4  (Spherical Joint)
  PART_5  via  JOINT_5  (Spherical Joint)
```

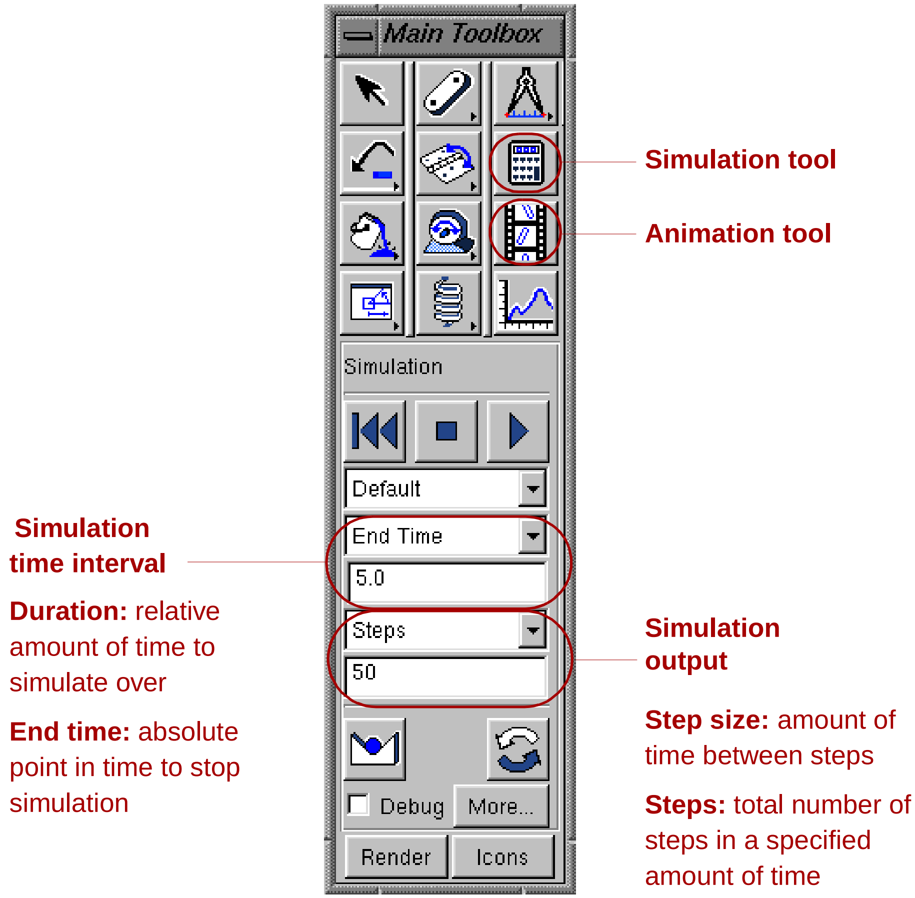
ADAMS/View main window



Simple Simulations

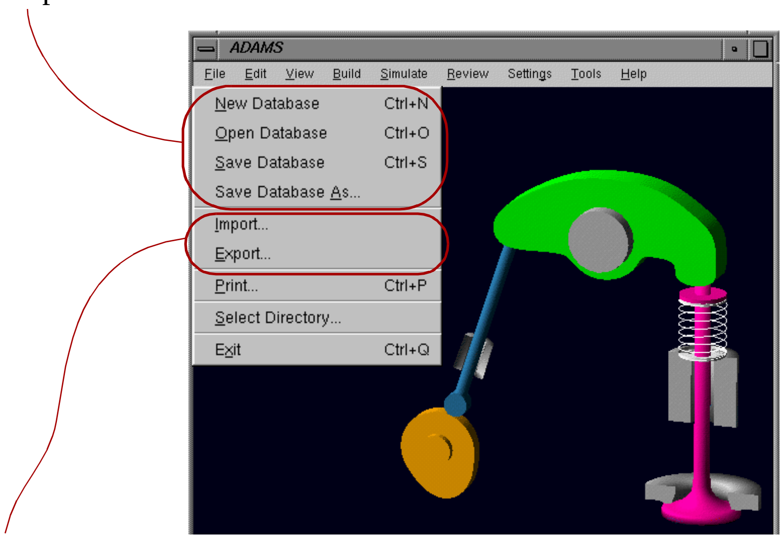
Simulation versus animation

- Simulations are solutions to equations of motion describing a mechanical system.
- Animations display a graphical playback of previously completed simulations.



Most common formats in which you can save ADAMS/View models

- ADAMS/View database files (.bin)
    - Include the entire modeling session including models, simulation results, plots, and so on.
    - Are typically very large.
    - Are platform independent in ADAMS 11.0 but all other versions are platform dependent.



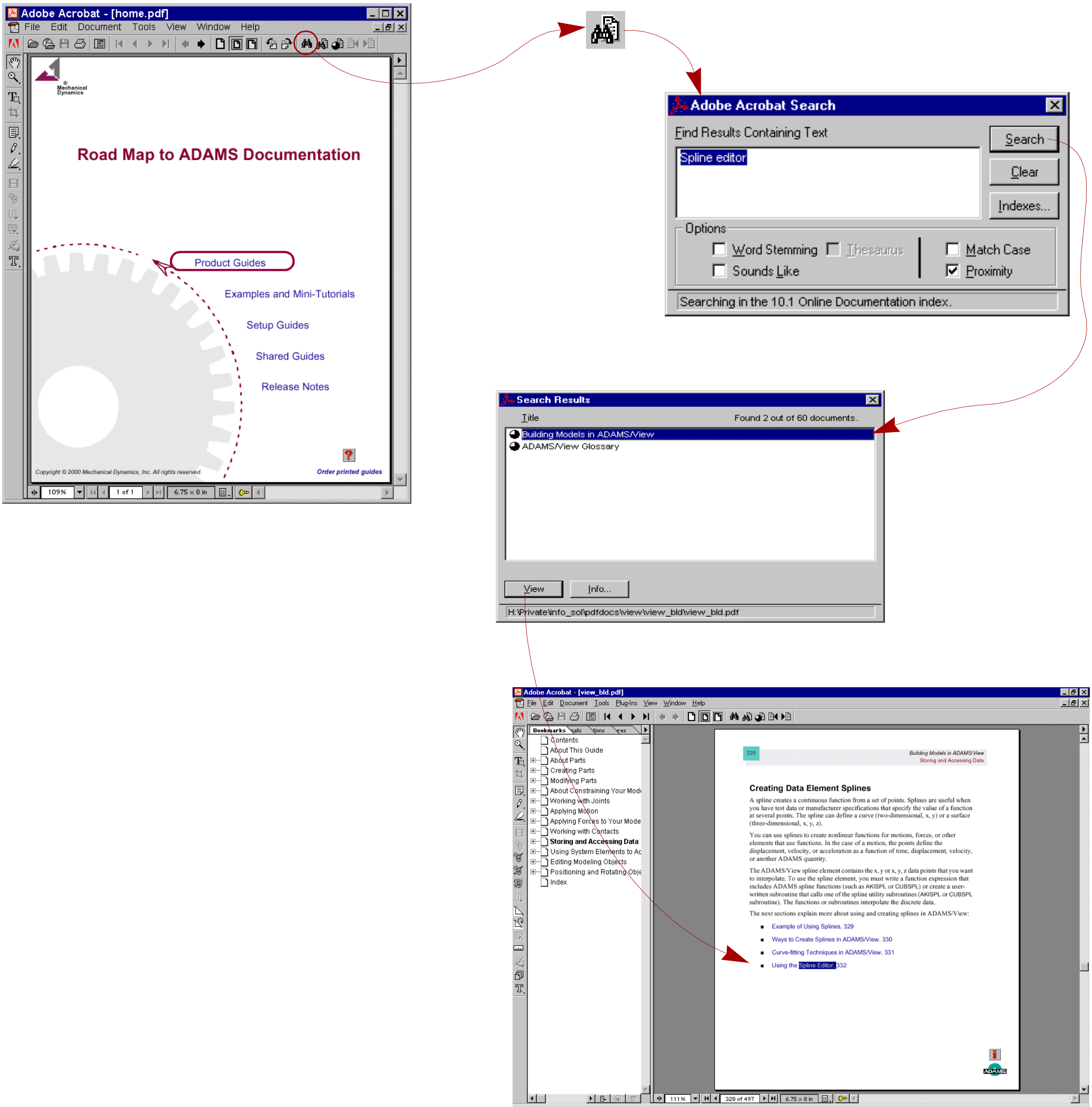
- ADAMS/View command files (.cmd)
    - Include only model elements and their attributes.
    - Are relatively small, editable text files.
    - Are platform independent.

Other formats in which you can import and export data

- ADAMS/Solver input files (.adm)
- Geometry files (STEP, IGES, DXF, DWG, Wavefront, Stereolithography)
- Test and spreadsheet data files
- Simulation results files (.msg, .req, .out, .gra, .res).

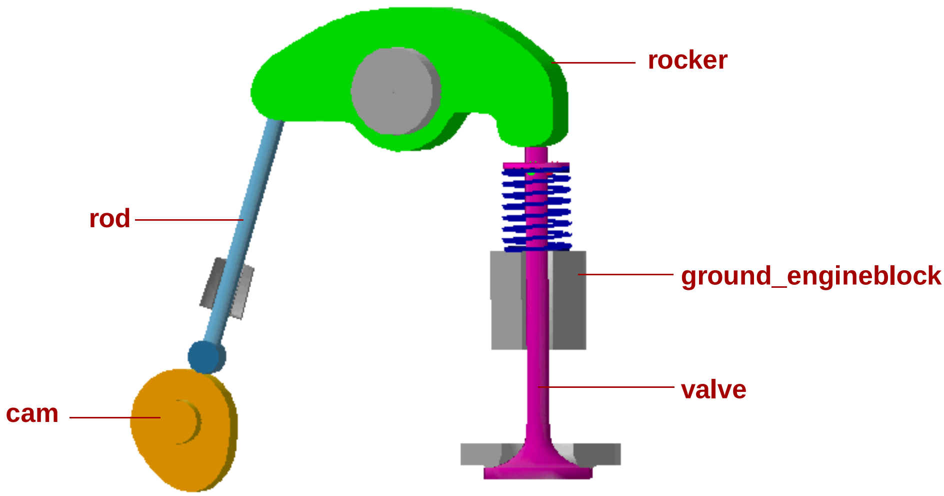
Referencing the online ADAMS/View guides

Doing global searches on any online ADAMS guide



Problem statement

Use ADAMS/View to manipulate, simulate, review, and refine the following model:



Model description

- The model represents a valvetrain mechanism.
- The cam is being rotated at a given velocity.
- The rod (follower) moves translationally based on its constraint to the cam.
- The rocker pivots about a pin attached to the engine block.
- The spring is always in compression to try and keep the rod in contact with the cam.
- The valve moves vertically as the rocker rotates.
- When the valve moves, it lets small amounts of air into the chamber below it (not modeled here).

Workshop 2—ADAMS/View Interface Overview...

Tips before you start

While working on this exercise, notice:

- The use of the right mouse button.
- The function of single-clicks and double-clicks.
- The messages on the Status bar.
- The animation options.

Start the workshop

Start ADAMS/View from the directory *exercise_dir*/mod_02_interface_overview and import the model command file valve.cmd. It contains commands to build a model named valve.

To start ADAMS/View in UNIX:

- From the ADAMS Toolbar , select the **ADAMS/View** tool .

To start ADAMS/View in Windows:

- On the **Start** menu, point to **Programs**, point to **ADAMS 11.0**, point to **AView**, and then select **ADAMS - View**.

To load the workshop files:

1. From the Welcome dialog box, select **Import a file**.
2. Click the file folder.

   The Find Directory dialog box appears.
3. Find and select the directory **mod_02_aview_interface** (*exercise_dir*/ mod_02_aview_interface).
4. Select **OK**.

   The File Import dialog box appears.
5. Set **File Type** to **ADAMS/View command file (*.cmd)**.
6. Right-click the **File to read** text box, and then select **Browse**.

   The Select File dialog box appears.

7. Find and select the file, **valve.cmd** and then select **Open**.
8. Select **OK**.

View the model

Now you'll learn how you can view models from different angles using the keyboard shortcuts for zooming, translating, and rotating.

To view the model from different angles:

1. To view a list of keyboard shortcuts, move the cursor away from the model, and then right-click in the ADAMS/View window.

   A menu appears, listing the keyboard shortcuts. To close the menu, left-click away from the menu.

2. In the space below, write the shortcut keys for performing the following view operations.

   Rotate: __________

   Translate: __________

   Zoom in and out: __________

   Zoom into a specific area: __________

   Fit: __________

   Front view: __________

3. Press the key representing the desired view operation, and follow the instructions in the Status bar.

Rename parts

Now you'll rename the parts to match the names given in the figure in the problem statement on page 30.

As you go through these instructions, notice that right-clicking always gives you a list of choices, while left-clicking selects an object.

To rename parts:

1. Move the cursor over a part and right-click. (For example, move the cursor over the rocker part.)
2. Point to **Part:PART_<x>**, and then select **Rename**.

   The Rename Object dialog box appears.

3 In the **New Name** text box, enter **.valve.<part name>**, and then select **OK**. (For example, for the rocker, you would enter .valve.rocker.) See the problem statement on page 30 for a listing of part names.

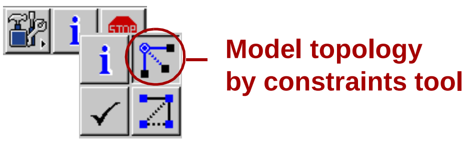
4 Continue renaming parts.

Inspect the model

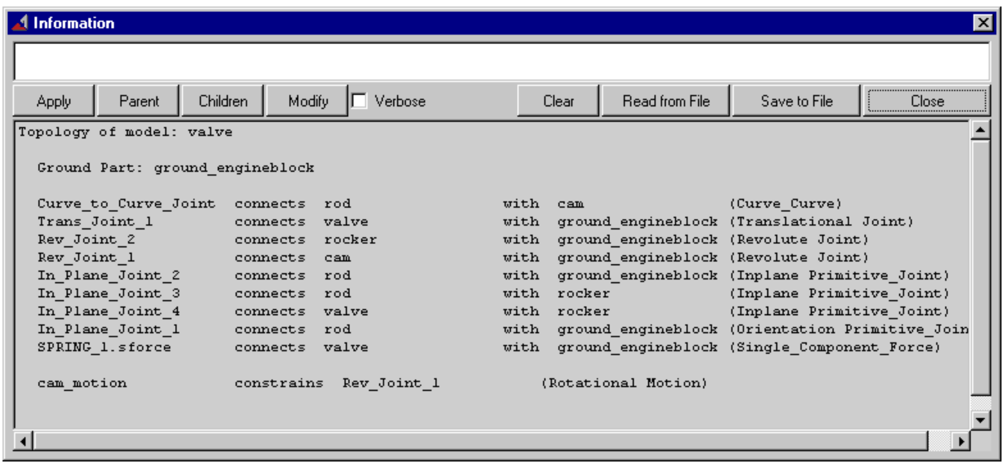
Now inspect the model to determine the number and type of constraints in the model and check if the model verified correctly. Use the values to answer Question 1 in Module review on page 38.

To determine the number and type of constraints:

1 Right-click the toolstack on the right side of the Status bar, and then select the **Model topology by constraints** tool.



The Information window appears as shown next:

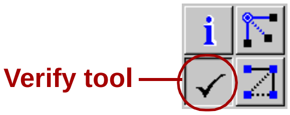


2 Note the number and type of constraints and use them to answer Question 1 in Module review on page 38.

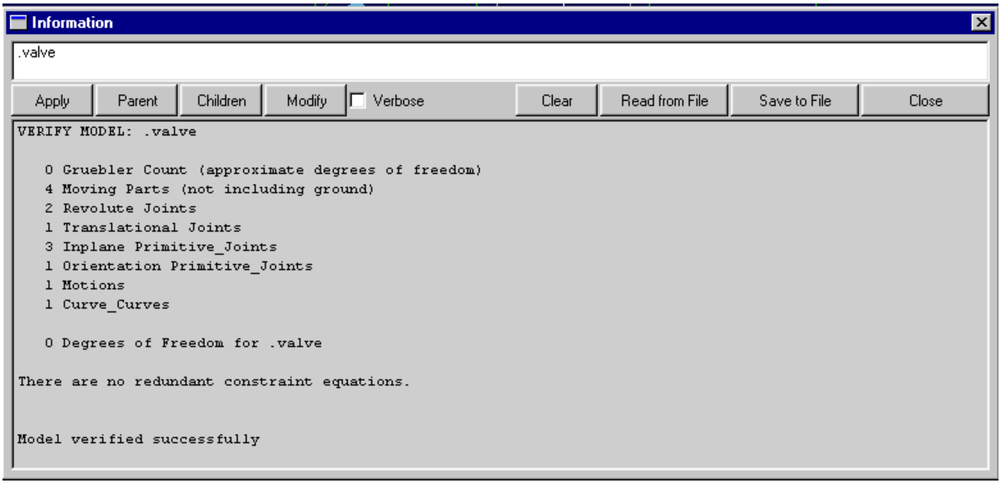
3 Select **Close**.

To check if the model verified successfully:

1 Right-click the toolstack on the right side of the Status bar and select the **Verify** tool.



The Information window appears as shown next:



Note that the text Model verified successfully appears in the Information window.

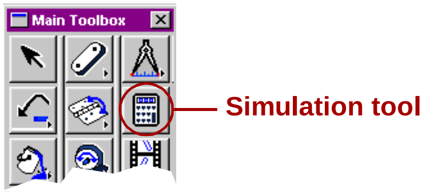
2 Select **Close**.

Simulate the model

In this section, you'll run a simulation for 2 seconds with 100 steps, and save the simulation results.

To run a simulation:

1. From the Main Toolbox, select the **Simulation** tool.



2. In the container that appears in the lower portion of the Main Toolbox:
   - Select **Default**.
   - Select **End Time**, and in the text box, below **End Time**, enter **2.0**.
   - In the text box below **Steps**, enter **100**.
3. Select the **Play** tool.
4. When the simulation is complete, select the **Reset** tool.

To save the simulation results:

1. From the **Simulate** menu, select **Interactive Controls**.

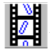
   The Simulation Control dialog box appears.
2. To save the last simulation results to the database under a new name, select the **Save Simulation** tool.

   The Save Run Results dialog box appears.
3. In the **Name** text box, enter a name for the simulation results, such as first_results.
4. Select **OK**.
5. Exit the Simulation Control dialog box.

Animate the results

In this section, you'll review the results of the simulation as an animation, which is a graphical playback of a simulation. You'll use the built-in ADAMS/View tools to run the animation. Optionally, you could use the ADAMS/PostProcessor tools to run animations.

Animate the model with icons turned off (default):

1. From the Main Toolbox, select the **Animation** tool .
2. Select the **Play** tool.
3. When the animation is complete, select the **Reset** tool.

Animate the model with icons turned on:

1. From the **Review** menu, select **Animation Controls**.

   The Animation Controls dialog box appears.
2. At the bottom of the Animation Controls dialog box, select **Icons**.
3. Select the **Play** tool.
4. When the animation is complete, select the **Reset** tool.
5. Close the Animation Controls dialog box.

Save your work

Now you'll save your work so the saved file contains only the model information.

To save your work:

1. From the **File** menu, select **Export**.
2. Set **File Type** to **ADAMS/View command file (*.cmd)**.
3. In the **File Name** text box, enter **valve1**.
4. In the **Model Name** text box, enter **valve**.
5. Select **OK**.
6. From the **File** menu, select **Exit**.
7. From the dialog box that appears, select **Exit, don't Save**.

Optional tasks

Have fun with the model:

This exercise introduces you to the ADAMS/View interface. Manipulate the model and experiment with it as much as you want.

Module review

1 How many constraints are there in this system? What type of constraints are they?

2 Is it possible to have more than one model in a database?

3 Is part geometry a direct child of a model? If not, what is part geometry a child of?

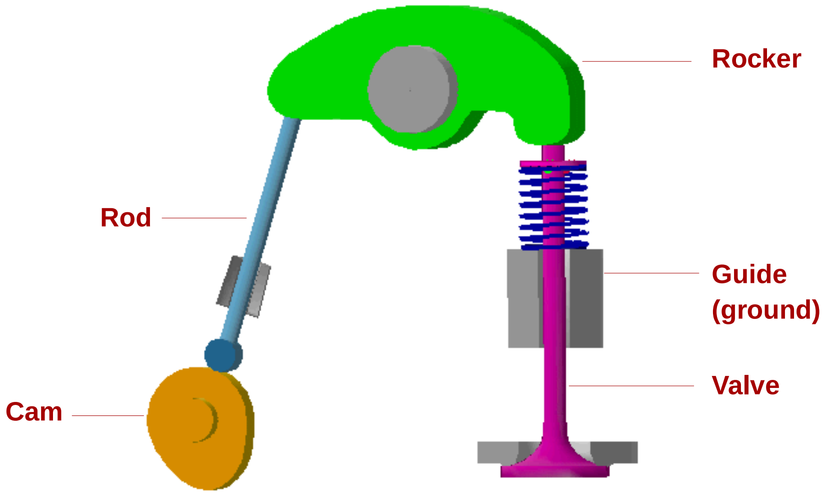
4 If you are in the middle of an operation and you are not sure what input ADAMS/View wants next, where should you look?

5 If you are working with our technical support staff and you want them to look at one of your files, what file format would you send them, a .cmd or .bin? Why?

6 What is the difference between the two search tools (the ones with binoculars) available in Adobe Acrobat Reader, which is the software we use to view the online documentation?

3 ADAMS/PostProcessor Interface Overview

Use the ADAMS/PostProcessor interface to simulate, review, and refine the model shown next:



What's in this module:

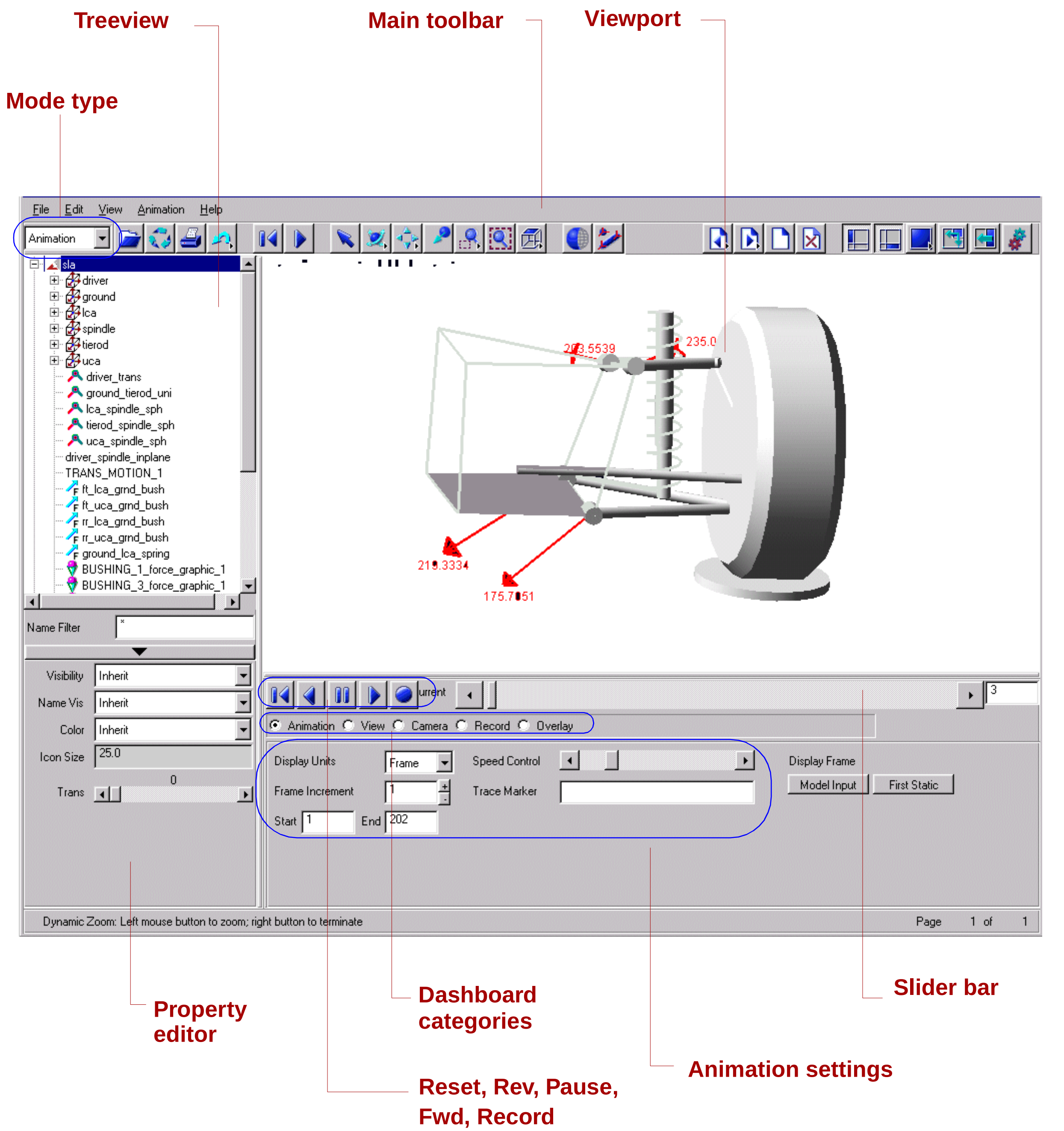
ADAMS/PostProcessor has two modes, depending on the active viewport:

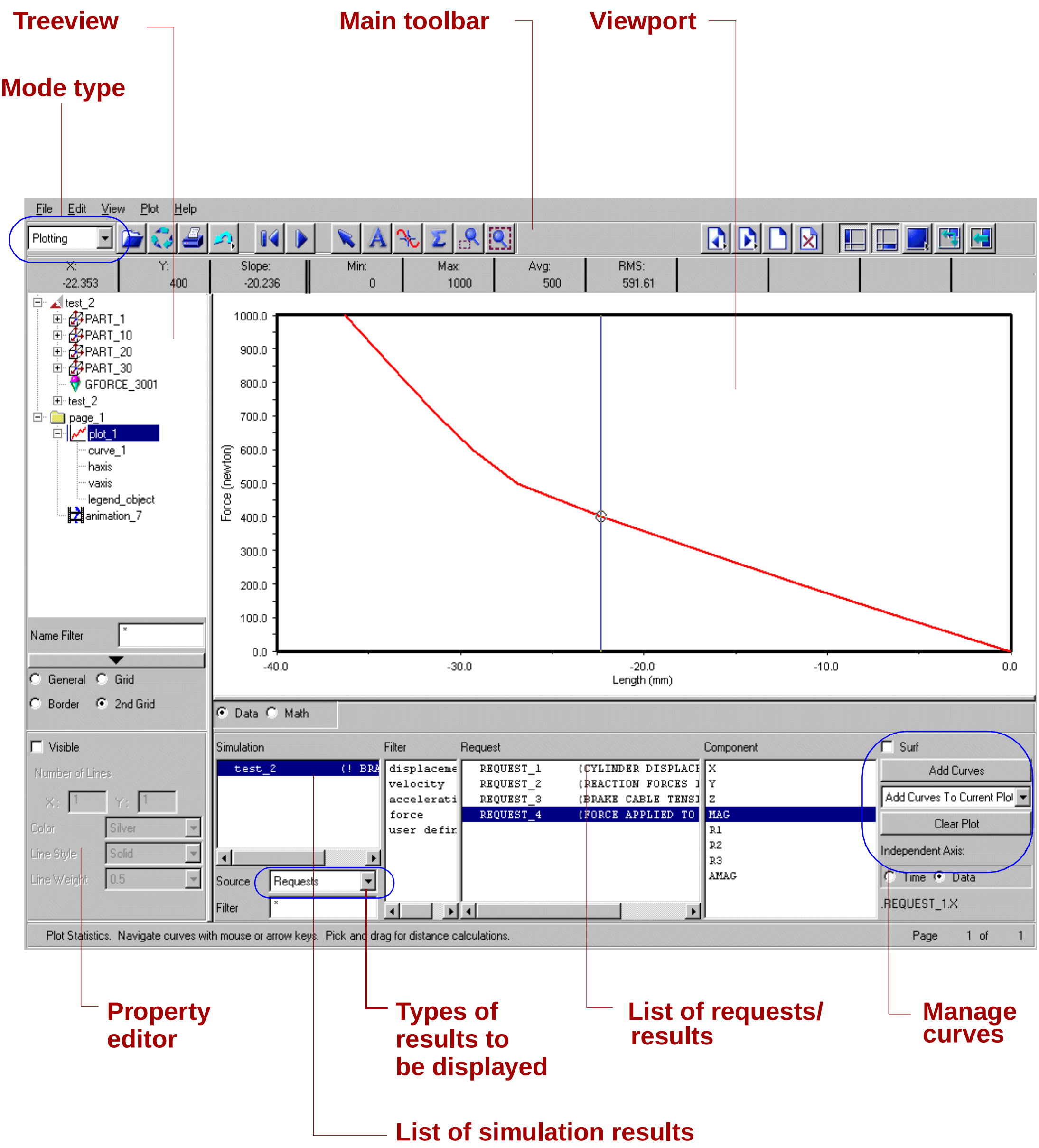
- Animation
- Plotting

Example:

The tools in the Main toolbar change if you load an animation or a plot into the viewport as shown next.

Animating

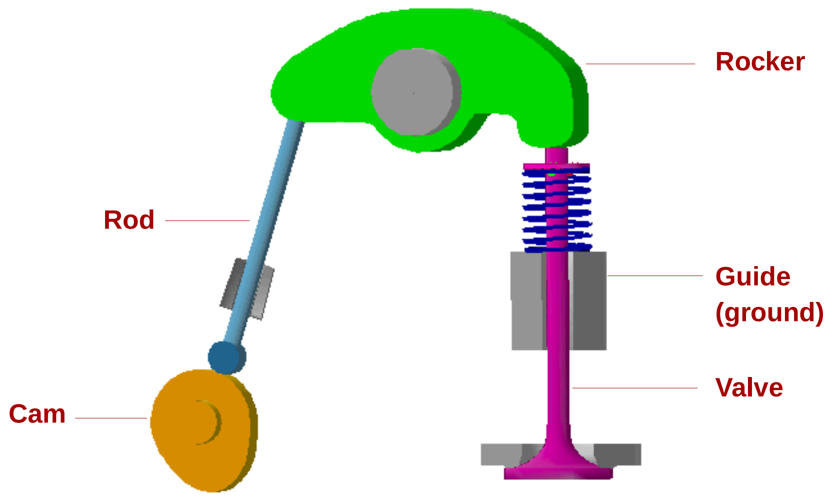

Treeview
Main toolbar
Viewport
Mode type
File Edit View Animation Help
Animation
sla
driver
ground
lca
spindle
tierod
uca
driver_trans
ground_tierod_uni
lca_spindle_sph
tierod_spindle_sph
uca_spindle_sph
driver_spindle_inplane
TRANS_MOTION_1
ft_lca_grnd_bush
ft_uca_grnd_bush
rr_lca_grnd_bush
rr_uca_grnd_bush
ground_lca_spring
BUSHING_1_force_graphic_1
BUSHING_3_force_graphic_1
Name Filter
Visibility Inherit
Name Vis Inherit
Color Inherit
Icon Size 25.0
Trans
Animation View Camera Record Overlay
Display Units Frame
Speed Control
Display Frame
Model Input
First Static
Frame Increment 1
Trace Marker
Start 1 End 202
Dynamic Zoom: Left mouse button to zoom; right button to terminate
Page 1 of 1
Property editor
Dashboard categories
Slider bar
Animation settings
Reset, Rev, Pause, Fwd, Record


Treeview
Main toolbar
Viewport
Mode type
Property editor
Types of results to be displayed
List of requests/ results
Manage curves
List of simulation results

Workshop 3—ADAMS/PostProcessor Overview

Problem statement

Use ADAMS/PostProcessor to manipulate, review, and refine the results of the suspension model you simulated in the previous module.



Model description

- The model represents a valvetrain mechanism.
- The cam is being rotated at a given velocity.
- The rod (follower) moves translationally based on its constraint to the cam.
- The rocker pivots about a pin attached to the engine block.
- The spring is always in compression to try and keep the rod in contact with the cam.
- The valve moves vertically as the rocker rotates.
- When the valve moves, it lets small amounts of air into the chamber below it (not modeled here).

Start the workshop

Start ADAMS/View from the directory *exercise_dir*/mod_03_ppt_interface and import the model command file valve1.cmd. It contains commands to build a model named valve.

To start the workshop:

1. Start ADAMS/View.
2. From the Welcome dialog box, select **Import a file**.
3. Click the file folder.

   The Find Directory dialog box appears.
4. Find and select the directory **mod_03_ppt_interface** (*exercise_dir*/mod_03_ppt_interface).
5. Select **OK**.

   The File Import dialog box appears.
6. Set **File Type** to **ADAMS/View command file (*.cmd)**.
7. Right-click the **File to read** text box, and then select **Browse**.

   The Select File dialog box appears.
8. Find and select the file, **valve1.cmd**, that you created in the previous workshop, and then select **Open**.
9. Select **OK**.

Simulate the model

Run a simulation for 2 seconds with 100 steps, and save the simulation results.

To run a simulation:

1. From the Main Toolbox, select the **Simulation** tool.
2. From the container in the Main toolbox:
   - Select **Default**.
   - Select **End Time**, and in the text box, below **End Time**, enter **2.0**.
   - In the **Steps** text box, enter **100**.
3. Select the **Play** tool.
4. When the simulation is complete, select the **Reset** tool.
5. Save the simulation results as you did in the section, Simulate the model on page 35 of Workshop 2—ADAMS/View Interface Overview.

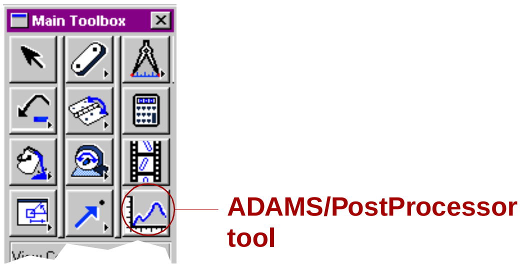
Plot the results

Now you'll plot the results using ADAMS/PostProcessor. You will plot:

- Displacement of the valve versus time. ADAMS/View tracks this data through a measure called valve_displacement.
- Force in the spring versus time. ADAMS/View tracks this data through a measure called force_in_spring.

To plot the results:

1 From the Main Toolbox, select the **ADAMS/PostProcessor** tool.



ADAMS/PostProcessor starts.

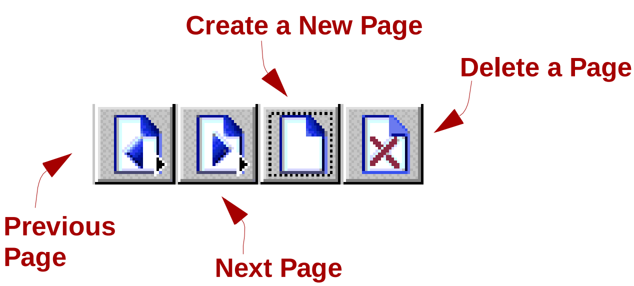
2 At the bottom of the window in the dashboard, from the Simulation list, select the name of the results set you saved in the previous section.

3 From the **Measures** list, select **Valve_Displacement**.

4 In the right corner of the dashboard, select **Add Curves**.

5 From the toolbar, select the **Create a New Page** tool.

The following figure shows the **Create a New Page** tool and other page tools.



In the treeview, shown on the left side of the ADAMS/PostProcessor window, you now have two pages.

6 From the **Measures** list, select **force_in_spring**.

7 Select **Add Curves**.

8 To return to ADAMS/View, in the upper right corner of the Main toolbar, select the **ADAMS/View** tool .

Manipulate model characteristics

You'll first find the spring stiffness coefficient, and then you'll modify the spring stiffness to 200 lbf/foot.

To find the spring stiffness coefficient:

1 Zoom in on the spring by typing a lowercase **w**, and then drawing a window around the spring.

2 Right-click the spring, point to **Spring:SPRING_1**, and then select **Info**.

The Information window appears.

3 Note the value of the stiffness coefficient.

4 Use the value to answer Question 2 in Module review on page 55.

5 Select **Close**.

To modify the spring stiffness to 200 lbf/foot:

1 Right-click the spring, point to **Spring:SPRING_1**, and then select **Modify**.

The Modify a Spring-Damper Force dialog box appears.

2 In the **stiffness coefficient** text box, enter **200 (lbf/foot)**.

Note: In the value you entered, the parentheses () are necessary because you enter compound fractional units.

3 Select **OK**.

4 Fit the model on the screen by typing a lowercase **f**.

Simulate the model

Run a simulation for 2 seconds with 100 steps.

To simulate the model:

1. In the Main Toolbox, select the **Simulation** tool.
2. In the container:
   - Select **Default**.
   - In the **End Time** text box, enter **2.0**.
   - In the **Steps** text box, enter **100**.
3. Select the **Play** tool.
4. When the simulation is complete, select the **Reset** tool.

To save the simulation results:

1. From the **Simulate** menu, select **Interactive Controls**.

   The Simulation Control dialog box appears.
2. To save the last simulation results to the database under a new name, select the **Save Simulation** tool .

   The Save Run Results dialog box appears.
3. In the **Name** text box, enter a new name.
4. Select **OK**.
5. Close the Simulation Control dialog box.

Overlay plots of force in spring for both simulations

Now you are going to overlay the results of both of your simulations to see the differences between the spring forces.

To overlay plots:

1 From the Main Toolbox, select the **ADAMS/PostProcessor** tool .

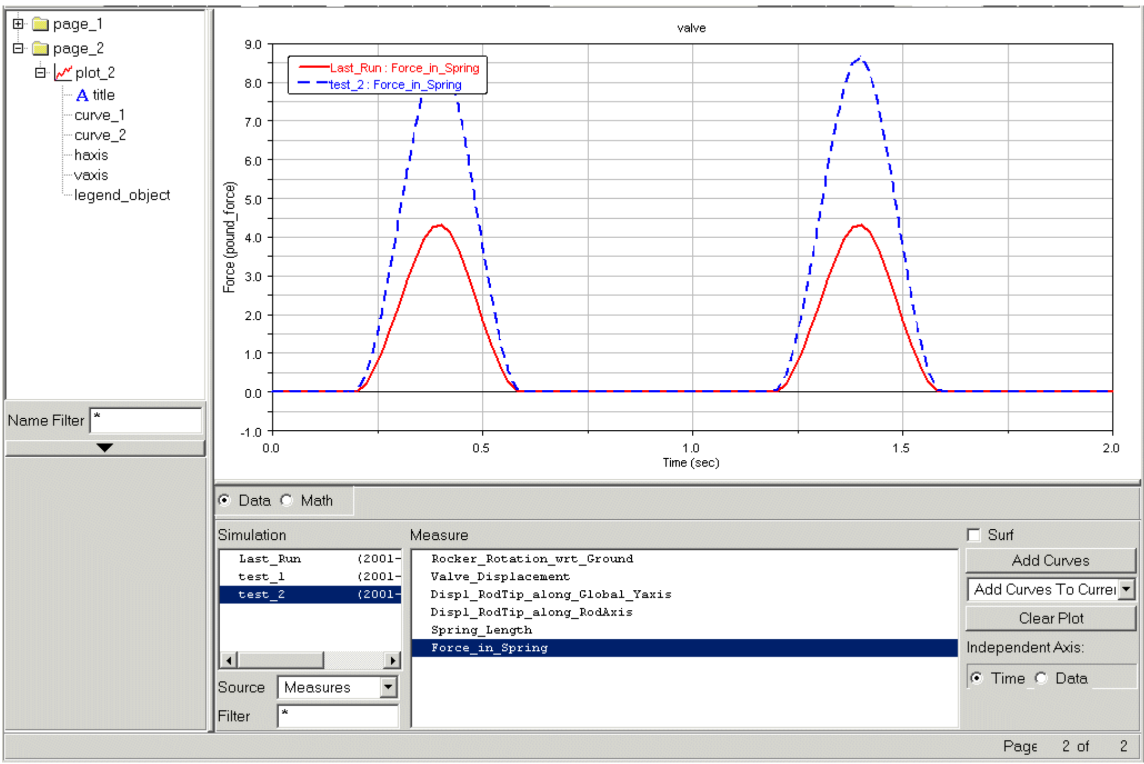
2 From the Simulation list, select the new simulation in your session.

3 From the Measure list, select **Force_in_Spring**.

4 Below the heading Independent Axis:, ensure that **Time** is selected.

5 Select **Add Curves**.

Notice the dashboard settings in the next figure.



Get plot statistics

Now you'll use the online documentation to find out how to get plot statistics and then find the plot statistics for the force_in_spring value.

To use the documentation to help you get plot statistics:

1. From the **Help** menu, select **ADAMS/PostProcessor Guide**.
2. Search for the phrase **plot statistics** and see where it leads you.

   If you are unable to find the phrase ask the instructor for help.
3. Use the Plot Statistics toolbar to find the maximum force_in_spring value in the second simulation.
4. Once you find the force_in-spring value, use it to answer Question 3 in Module review on page 55.

Modify the plot graphics

Now you'll modify the graphics of the plot to make the information in it more readable.

To give the plot a title:

1. In the treeview, expand **page_2** by clicking the **+** sign.
2. Select **plot_2**.
3. In the Property Editor below the treeview, enter the title **Spring Force vs. Time**.
4. Select **Enter**.

To label the vertical axis as Spring Force (lbf):

1. In the treeview, expand the plot by clicking the **+** sign.
2. Select **vaxis**.
3. In the Property Editor, select **Labels**.
4. Change the label to **Spring Force (lbf)**.

To modify the legend text and its placement:

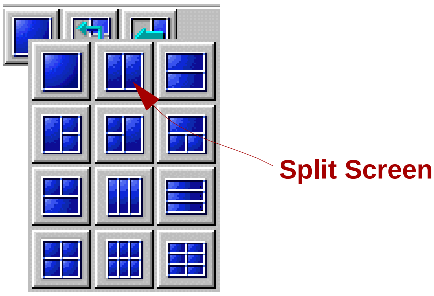
1. In the treeview, expand **plot_2**.
2. In the treeview, select **curve_1**.
3. In the Property Editor below, change the **Legend** text box to **k=100(lbf/foot)**.
4. Change the legend for **curve_2** to **k=200**.
5. In the treeview, select **legend_object**.
6. In the Property Editor, set **Placement** to **Top Right**.

Add an Animation

ADAMS/PostProcessor lets you display animations and plots at the same time. In this section, you'll add an animation next to your plot. You can also run the animation and watch the results appear in the plot.

To add an animation next to your plot:

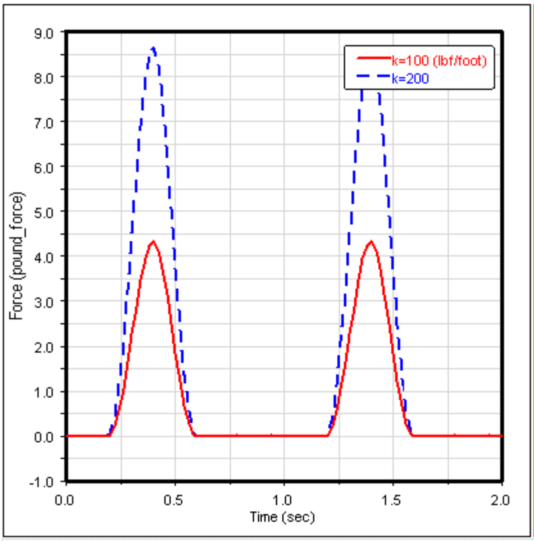
1. In the treeview, select **page_1**.
2. Split the screen by right-clicking on the **Page Layout** tool above the viewport in the toolbar, and selecting the **Split Screen** tool.

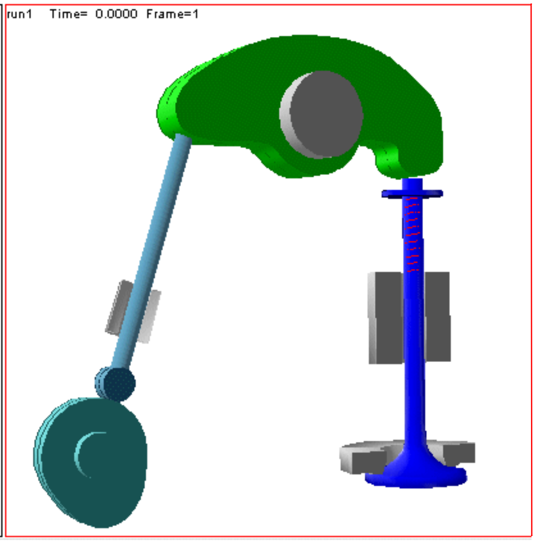


3. Set the new viewport to Animation by right-clicking in the viewport and choosing **Load Animation** from the pop-up menu.

4 From the Database Navigator, select one of the simulation results that you want to animate.

5 Select **OK**.

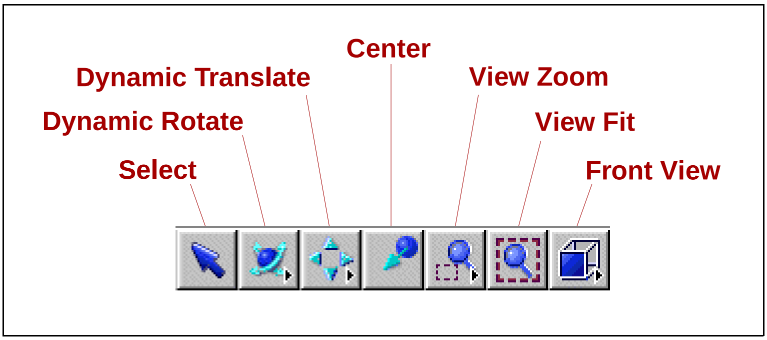




Viewing Results

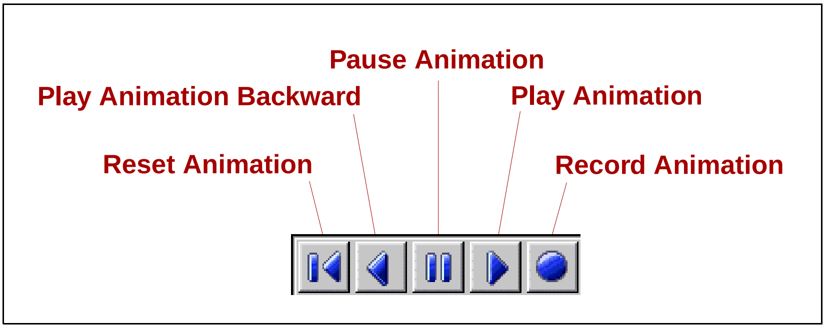
To view an animation of the results:

- Adjust your view of the model on your screen using the tools above the viewport. The figure below highlights some of the tools that are available. Try experimenting with the rotate, zoom, and translate tools.



To play an animation of the results:

- Play an animation of your model using the tools that are located above the viewport and in the dashboard. Experiment with the play and pause tools.



Modifying the graphics of your animation

To modify the graphics settings of your animation:

1. Select the **View** button in the dashboard.

   Your view options appear below the View button.

2. Experiment with the four check boxes that are available.

To change the color of the cam:

1. From the treeview, expand the model by clicking on the **+** sign.
2. Select **Cam**.
3. Below the treeview, in the property editor, set **Color** to **Coral**.

To enlarge the graphics that illustrate force:

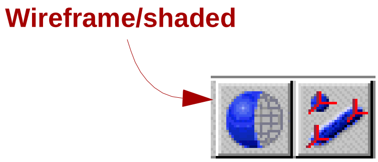
1. From the **Edit** menu, select **Preferences**.

   The PPT Preferences dialog box appears.

2. In the **Force Scale** text box, enter a value that is greater than 50, and then select **Close**.
3. Experiment with changing the scale of the force graphics.

To change the view from shaded to wireframe:

4 On the top toolbar, select **Wireframe/shaded**.



Save your ADAMS/PostProcessor session

To save your session:

1 Return to ADAMS/View.

2 Save your work and then exit ADAMS/View.

Module review

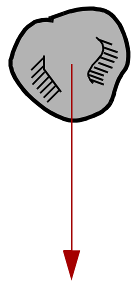
1. What is the mass of the valve? What is this mass currently based on?

2. What is the stiffness coefficient of the spring?

3. What was the maximum spring force when the spring coefficient was 200 lbf/foot?

4 FALLING STONE

Find the displacement, velocity, and acceleration of a stone after one second, when the stone, with zero initial velocity, falls under the influence of gravity.

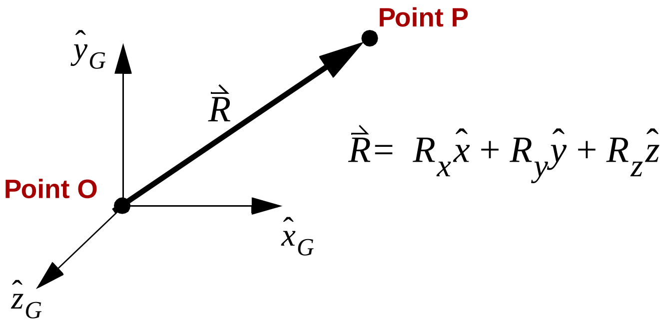


$$\vec{g} = 9810\frac{mm}{s^2}$$

What's in this module:

Definition of a coordinate system (CS)

- A coordinate system is essentially a measuring stick to define kinematic and dynamic quantities.



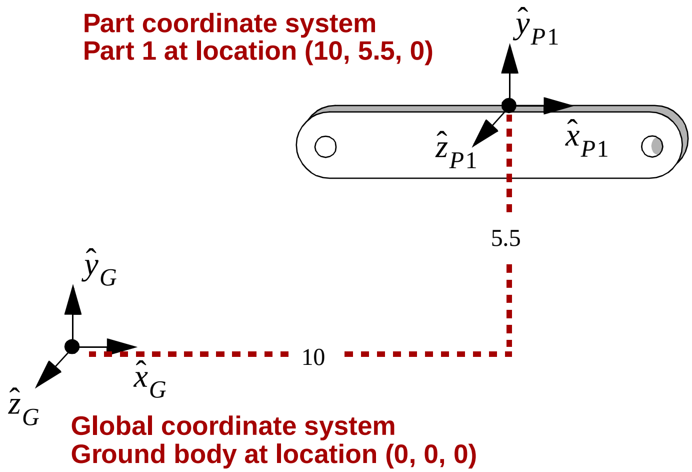
Types of coordinate systems

- Global coordinate system (GCS):
  - Rigidly attaches to the ground part.
  - Defines the absolute point (0,0,0) of your model and provides a set of axes that is referenced when creating local coordinate systems.
- Local coordinate systems (LCS):
  - Part coordinate systems (PCS)
  - Markers

Part Coordinate System

Definition of part coordinate systems (PCS)

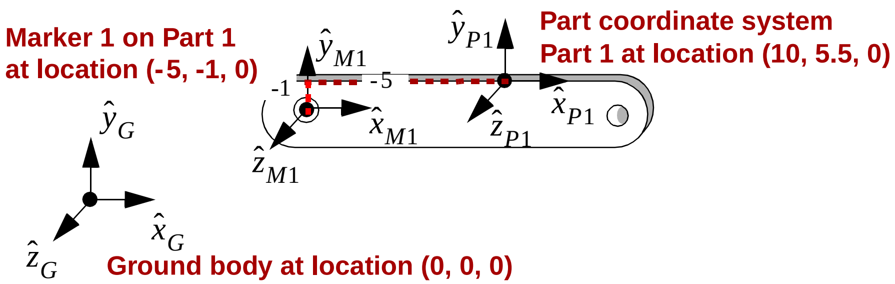
- They are created automatically for every part.
- Only one exists per part.
- Location and orientation is specified by providing its location and orientation with respect to the GCS.



- When created, each part's PCS has the same location and orientation as the GCS.

Definition of a marker

- It attaches to a part and moves with the part.
- Several can exist per part.
- Its location and orientation can be specified by providing its location and orientation with respect to GCS or PCS.



- It is used wherever a **unique location** needs to be defined. For example:
  - The location of a part’s center of mass.
  - The reference point for defining where graphical entities are anchored.
- It is used wherever a **unique direction** needs to be defined. For example:
  - The axes about which part mass moments of inertia are specified.
  - Directions for constraints.
  - Directions for force application.
- By default, all marker locations and orientations are expressed in GCS.

Differences Between Parts and Geometry

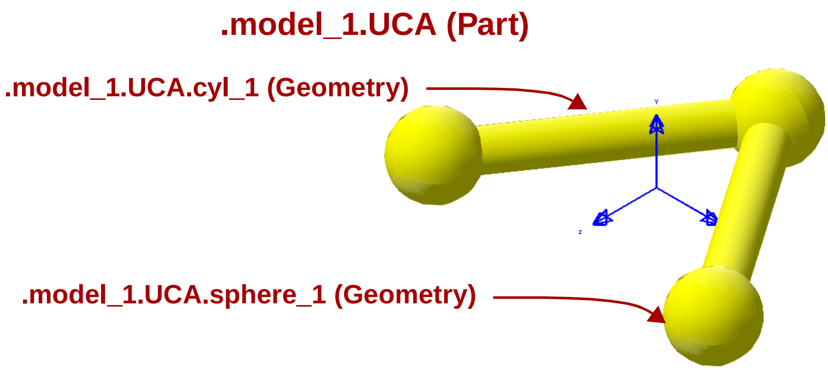
Parts

Define bodies (rigid or flexible) that can move relative to other bodies and have the following properties:

- Mass
- Inertia
- Initial location and orientation (PCS)
- Initial velocities

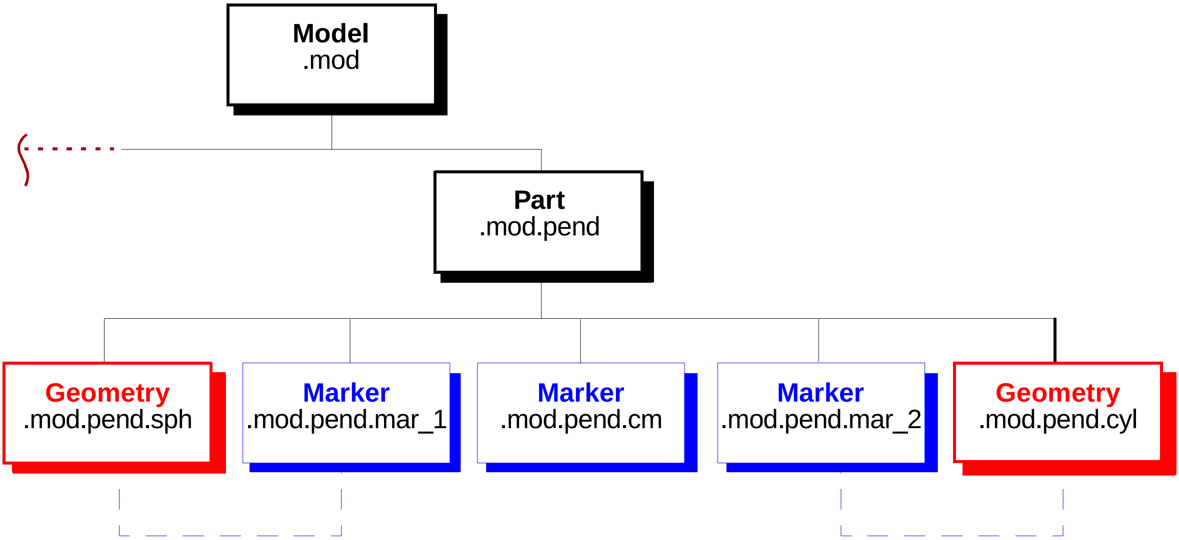
Geometry

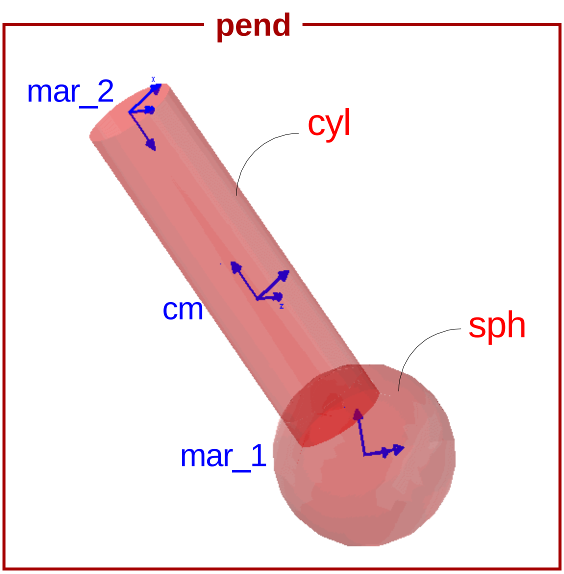
- Is used to add graphics to enhance the visualization of a part using properties such as:
  - Length
  - Radius
  - Width
- Is not necessary to perform simulations.



Dependencies in ADAMS

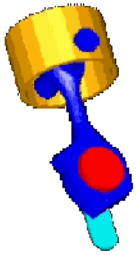
To understand the relationship between **parts**, **geometry**, and **markers** in ADAMS/View, it is necessary to understand the dependencies shown next:





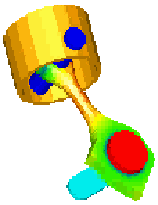
Types of Parts in ADAMS

Rigid bodies



- Are movable parts.
- Possess mass and inertia properties.
- Cannot deform.

Flexible bodies (beyond the scope of this course)



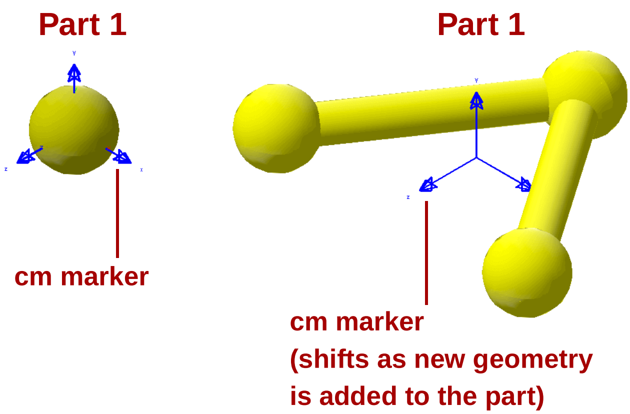
- Are movable parts.
- Possess mass and inertia properties.
- Can bend when forces are applied to them.

Ground part

- Must exist in every model and is automatically created when a model is created in ADAMS/View.
- Defines the GCS and the global origin and, therefore, remains stationary at all times.
- Acts as the inertial reference frame for calculating velocities and acceleration.

Mass and inertia properties

- ADAMS/View automatically calculates mass and inertial properties **only** for three-dimensional rigid bodies.
- ADAMS/View calculates the total mass and inertia of a part based on the part's density and the volume of its geometry.
- You can change these properties manually.
- ADAMS/View assigns mass and inertial properties to a marker that represents the part's center of mass (cm) and principal axes.
- You can change the position and orientation of the part's cm marker.



- The orientation of the cm marker also defines the orientation of inertial properties I_{xx}, I_{yy}, I_{zz}.

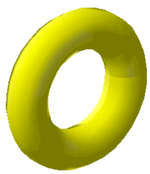
Measures

Measures in ADAMS

- Represent data that you would like to quantify during a simulation, such as:
  - Displacement, velocity, or acceleration of a point on a part
  - Forces in a joint
  - Angle between two bodies
  - Other data resulting from a user-defined function
- Capture values of measured data at different points in time over the course of the simulation.

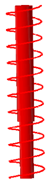
Definition of object measures

Measure pre-defined measurable characteristics of parts, forces, and constraints in a model



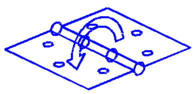
Part measure characteristics:

- CM position
- CM velocity
- Kinetic energy
- Others



Spring measure characteristics:

- Deformation
- Force

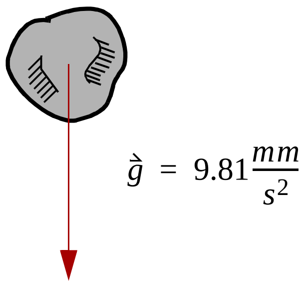


Joint measure characteristics:

- Relative velocity
- Force
- Torque
- Others

Problem statement

Find the displacement, velocity, and acceleration of a stone after one second, when the stone with zero initial velocity, falls under the influence of gravity.



Start the workshop

First, start ADAMS/View and create a model in the directory *exercise_dir/* mod_04_falling_stone. Executing ADAMS/View in that directory ensures that all saved data gets stored there.

To start the workshop:

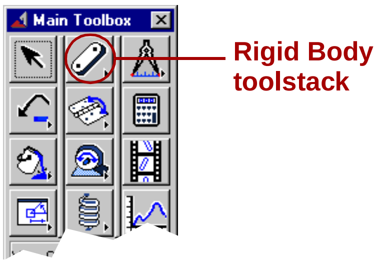
1. Start ADAMS/View.
2. In the Welcome dialog box:
   - Under the heading, **How would you like to proceed**, select **Create a new model**.
   - Set the directory to *exercise_dir/*mod_04_falling_stone.
   - Name the model **projectile**.
   - Set gravity to **Earth Normal** (-Global Y).
   - Set units to **MMKS - mm, Kg, N, s, deg**.
3. Select **OK**.

Build the stone

Use the **Sphere** tool  to create a stone part with a 50 mm radius and its center at the global origin. You'll also rename the part and set its mass to 1 kg.

To build the stone:

1. To view the coordinates as you create the sphere so you know its size, from the **View** menu, select **Coordinate Window.**
2. From the Main Toolbox, right-click the **Rigid Body** toolstack, and then select the **Sphere** tool .



3. Follow the Status bar instructions and pick the center of the sphere at the global origin, then drag the cursor until you create a sphere with a 50 mm radius.

To rename it:

1. Right-click the sphere, point to **Part:PART_2**, and then select **Rename**.
2. In the **New Name** text box, enter **.projectile.Stone**, and then select **OK**.

To set the mass to 1 kg:

1. Right-click the sphere, point to **Part:Stone**, and then select **Modify.**
2. In the **Mass & Inertia defined by** text box, select **User Input**.
3. If an alert box opens, select **Close**.
4. In the **Mass** text box, enter **1.0**.
5. Select **OK.**

Create measures for the falling stone

To calculate the vertical displacement, velocity, and acceleration of the stone's cm marker in the $\hat{y}_g$,direction, you'll create three object (part) measures. You'll set Y as the component to measure.

To calculate the displacement of the stone in the $\hat{y}_g$ direction:

1. Right-click the stone, point to **Part:Stone**, and then select **Measure.**
2. In the **Measure Name** text box, enter:

   displacement
3. Set **Characteristic** to **CM position**.
4. Set **Component** to **Y**.
5. Set **From/At** to **.projectile.Stone.cm**.
6. Select **Create strip chart.**
7. Select **OK.**

   A measure strip chart appears. It is empty because you need to run a simulation before ADAMS has the necessary information for the chart.

To calculate the velocity of the stone in the $\hat{y}_g$ direction:

1. Right-click the stone, and select **Measure**.
2. In the **Measure Name** text box, enter:

   velocity
3. Set **Characteristic** to **CM velocity**.
4. Set **Component** to **Y**.
5. Set **From/At** to **.projectile.Stone.cm**.
6. Select **Create strip chart.**
7. Select **OK.**

To calculate the acceleration of the stone in the $\hat{y}_g$ direction:

- Follow the instructions above but set **Characteristic** to **CM acceleration**.

Verify the model

Now you'll verify the model. When you verify the model, ADAMS/View checks for error conditions such as misaligned joints, unconstrained parts, or massless parts in dynamic systems and alerts you to other possible problems in the model.

To verify the model:

1. In the right corner of the Status bar, right-click the **Information** tool, and then select the **Verify** tool.
2. In the Information window, check that the model has verified successfully.
3. Close the Information window.

Set up and run a simulation

Now you'll zoom out the display so that the falling stone is clearly visible while it simulates. You'll then simulate it for 1 second with 50 steps.

To zoom out:

1. Select the **Select** tool to display the view control options in the toolbox.
2. Select the **Zoom** tool, and then click and drag the mouse to zoom out until the entire working grid is visible.
3. Select the **Translate** tool, and then drag the working grid to the top of the screen.

To run a simulation for 1 second with 50 steps:

1. In the Main Toolbox, select the **Simulation** tool.
2. In the **End Time** text box, enter **1.0** and in the **Steps** text box, enter **50.**
3. Select the **Play** tool.

   As the stone falls, ADAMS/View plots the corresponding data on the displacement, velocity, and acceleration graphs.

4 When the simulation ends, reset the model to the input, or design configuration by selecting the **Reset** tool.

5 Animate the simulation to replay the simulation without simulating again.

Find the values of displacement, velocity, and acceleration

Now you'll use ADAMS/PostProcessor to find the stone's displacement, velocity, and acceleration after 1 second.

To run ADAMS/PostProcessor:

- Right-click the blank area inside the strip chart **.projectile.displacement**, point to **Plot:scht1**, and then select **Transfer to Full Plot**.

  ADAMS/PostProcessor appears, replacing the ADAMS/View window.

To find the value of the stone's displacement:

1 In ADAMS/PostProcessor, from the Main toolbar, select the **Plot Tracking** tool .

2 Because you want to know the final conditions after 1 second, move the cursor over the end point of the plot.

3 In the area below the menu bar, the value of X is displayed as 1. Note the value of Y; this is your answer.

4 Compare this value of Y to the results given in the closed-form solution on page 73.

5 Use the value to answer Question 1 in Module review on page 74.

To find the value of the stone's velocity after 1 second:

1 Select **Surf**.

This lets you view a selected measure without using the Add Curves button.

2 Set **Source** to **Measures**.

3 From the **Measures** list, select **velocity**.

4 Because you want to know the final conditions after 1 second, move the cursor over the end point of the plot.

5 In the area below the menu bar, the value of X appears. It is 1. Note the value of Y; this is your answer.

6 Compare this value of Y to the results given in the Closed-form solution on page 73.

7 Use the value to answer Question 2 in Module review on page 74.

To find the value of stone's acceleration after 1 second:

1 From the **Measures** list, select **acceleration**.

2 To display the acceleration plot, select **Surf**.

3 Because you want to know the final conditions after 1 second, move the cursor over the end point of the plot.

4 In the area below the menu bar, the value of X will be displayed as 1. Note the value of Y; this is your answer.

5 Compare this value of Y to the results given in the closed-form solution on page 73.

6 Use the value to answer Question 3 in Module review on page 74.

7 To return to ADAM/View and close all three plots, select the **ADAMS/View** tool.

Save your work

Now save your work such that the file contains only the model information. You will use this model in the next module.

Tip: Save the model as a command file.

To save your work:

1. From the **File** menu, select **Export**, and then select **OK**.
2. Exit ADAMS/View.

Optional tasks

Save your work before performing these tasks. **Do not** save your work after performing these tasks because you will use this model in the next module. If you must save the model after performing these tasks, give the model a different name.

To inspect the behavior of the stone after changing its mass:

1. Change the mass of the stone to 2 kg.
2. Simulate the model.
3. Compare the results of this simulation with the results of the simulation where the mass of the stone was 1 kg.
4. Does changing the mass affect the displacement, velocity, or acceleration?
5. Measure the kinetic energy of the stone. Do these results make sense?

   K.E. = (1/2)m*v^2
6. From the **File** menu, select **Exit**.

ADAMS results

- Displacement after 1 sec = -4903.3 mm
- Velocity after 1 sec = -9806.6 mm/sec
- Acceleration after 1 sec = -9806.6 mm/sec^2

Closed-form solution

Analytical solution:

$s = ½ (at^2) = 4903.325$ mm

$v = at = 9806.65$ mm/sec

$a= g = 9806.65$ mm/sec^2

Where:

s = Distance (mm)

a = Acceleration (mm/sec^2)

t = Time (sec)

v = Velocity (mm/sec)

Module review

1 What is the displacement of the stone after one second?

2 What is the velocity of the stone after one second?

3 What is the acceleration of the stone after one second?

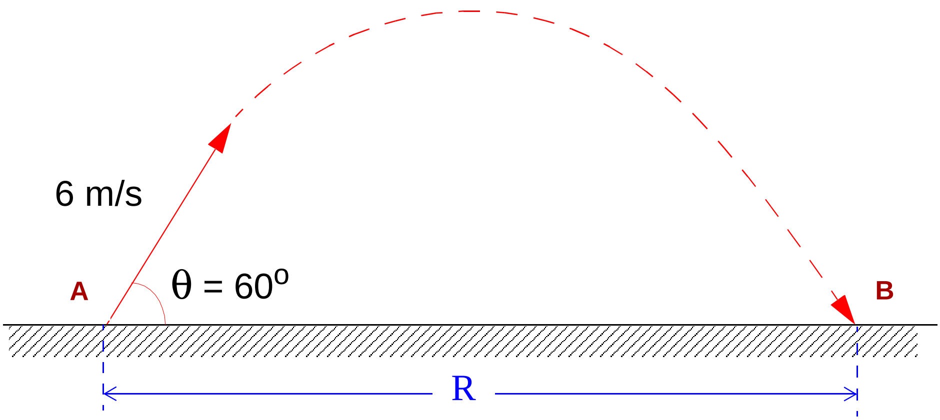
4 What are the most basic building blocks in ADAMS, which are used in parts, constraints, forces, and measures?

5 Why is the ground part automatically created?

6 Can ADAMS/View generate mass properties for two-dimensional geometry? Why?

5 PROJECTILE MOTION

Compute the range, R, when a stone is launched as a projectile with an initial speed of 6 m/s at an angle of 60^o, as shown next.



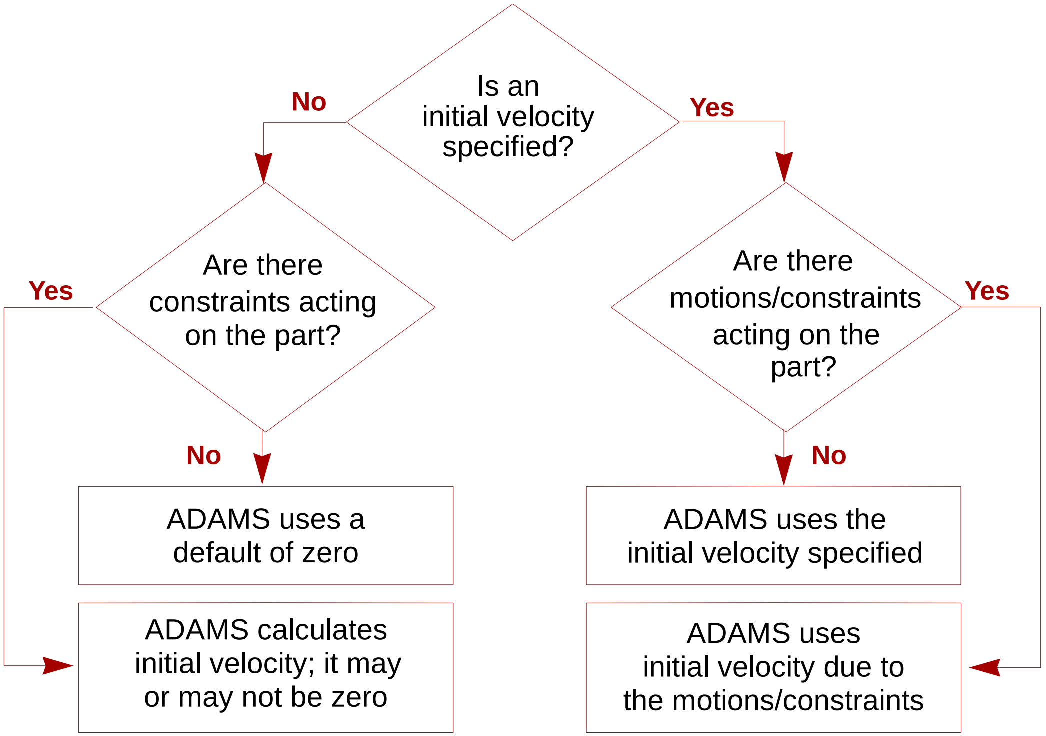
What's in this module:

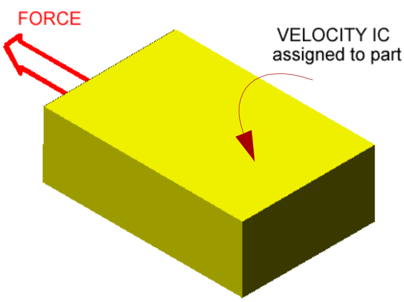
Initial location and orientation

- The design configuration of all the parts (their part coordinate systems) in a model defines their initial locations and orientations.
- You can fix a part's location and orientation to be used during the assembly simulation (covered later).

Initial velocities

- In ADAMS, a part initially moves (at t = 0) as follows:





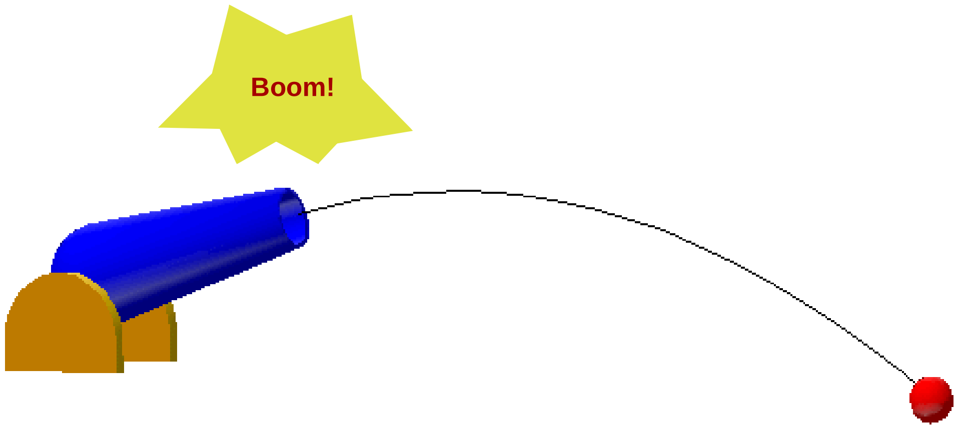
Point Trace

Definition of a point trace

- Tracks the location of a marker during an animation.
- Can be used to visualize the clearance between two bodies during a simulation.

Example of a point trace

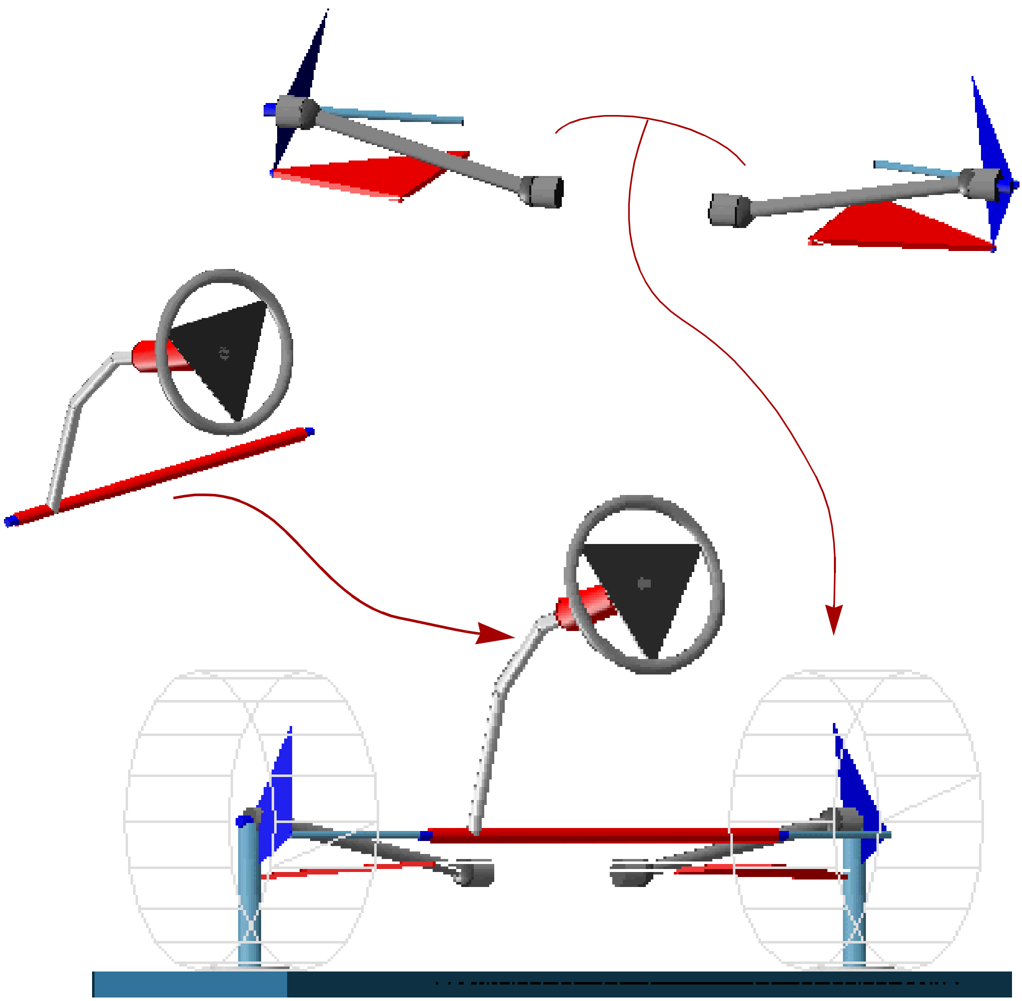
- Trajectory of a ball.



The crawl-walk-run approach

- Do not build the entire mechanism at once.
- As you add a new component, make sure that it works correctly.
- Check your model at regular intervals.

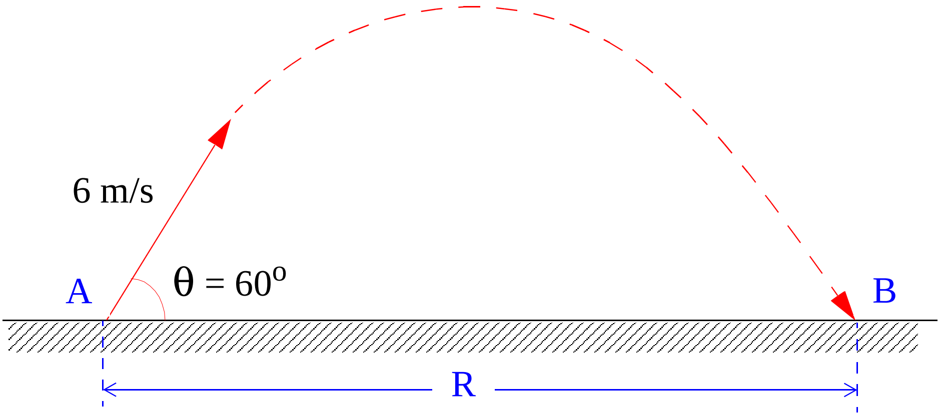
Avoid the need for complex debugging by following the crawl-walk-run approach!



Workshop 5—Projectile Motion

Problem statement

Compute the range, R, when a stone is launched as a projectile with an initial speed of 6 m/s at an angle of 60^o, as shown next.



Model description

In this workshop, you use the model you built in Workshop 4—Falling Stone on page 66.

Start the workshop

To start the workshop, import the model that you created in the previous module. Note that the model file is not in the current working directory. It is in the directory *exercise_dir*/mod_04_falling_stone/completed.

If you need a fresh copy of the model, import the command file projectile.cmd from your current working directory.

To start the workshop:

1. In the Welcome dialog box, under the heading, **How would you like to proceed**, select **Import a file**.
2. Set the directory to ***exercise_dir*/mod_05_projectile**. Executing ADAMS/View in this directory ensures that all saved data gets stored here.
3. Select **OK**.
4. Find and select **projectile.cmd**.
5. Select **OK**.

Build the plane

In this section, you'll build a plane using the Box tool. The plane will have the following dimensions:

- Length: 3500 mm
- Height: 100 mm
- On ground

Before building the plane, you'll set up the display by resetting the working grid to 4000 mm x 3000 mm with spacing of 50 mm, and zooming out.

To set the display:

1. From the **Settings** menu, select **Working Grid**.
2. In the **Size: X** text box, enter **4000**.
3. In the **Size: Y** text box, enter **3000**.
4. In the **Spacing** text boxes, enter **50**, and then select **OK**.
5. Zoom out by typing a lowercase **z**, and then clicking and dragging the mouse to zoom out and view the entire working grid.

To build the plane:

1 Turn on the coordinate window (from the **View** menu, select **Coordinate Window**).

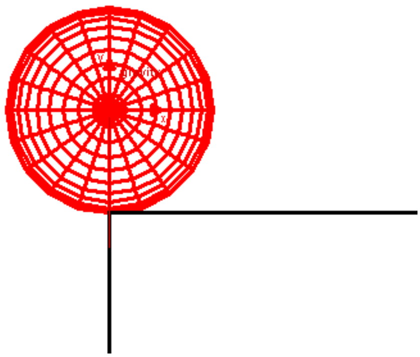
2 From the Main Toolbox, right-click the **Rigid Body** toolstack, and then select the **Box** tool .

3 In the toolbox container:

- Select **On Ground**.
- Select **Length**, and then in the **Length** text box, enter **3500 mm**.
- Select the **Height** and **Depth** options, and then in the **Height** and **Depth** text boxes, enter **100 mm**.

4 Use the mouse to select the corner of the box at **0, -150, 0**.

The stone should appear to be balanced at the upper left corner of the plane in a front view.



Set initial conditions

Now set initial velocity conditions for the stone as follows:

- V_{x_o} = 6000*cos(60^o) = 3000 mm/sec
- V_{y_o} = 6000*sin(60^o) = 5196 mm/sec

To set initial conditions:

1. Reset the Main Toolbox by selecting the **Select** tool .
2. Right-click the stone, point to **part:Stone**, and then select **Modify**.
3. From the Modify Rigid Body dialog box, select **Velocity ICs**.
4. In the Simulation Settings dialog box, under **Initial velocity along:** select **X axis**, and in the **X axis** text box, enter **(6*cos(60d)(m/sec))**.
5. In the Simulation Settings dialog box, under **Initial velocity along:** select **Y axis**, and in the **Y axis** text box, enter **(6*sin(60d)(m/sec))**.
6. Select **Apply**, and then close the window.
7. From the Modify Rigid Bodies dialog box, select **OK**.

Create measures for projectile motion

Next, create an object (part) measure to calculate the horizontal displacement, $\hat{x}_g$, of the stone's center of mass (cm) marker when it is projected.

To create a measure:

1. Right-click the stone, point to **part:Stone**, and then select **Measure**.
2. In the **Name** text box, enter:

   R_displacement
3. Set **Characteristic** to **CM position**.
4. Set **Component** to **X**.
5. Set **From/At** to **.projectile.Stone.cm**.
6. Select **Create strip chart**.
7. Select **OK**.

Run a simulation

Run a simulation for 1.5 seconds, using a sampling rate of .02 seconds.

To run a simulation:

1. From the Main Toolbox, select the **Simulation** tool.
2. In the **End Time** text box, enter **1.5**.
3. In the **Step Size** text box enter **0.02**.
4. Select the **Play** tool.

   ADAMS/View runs the simulation and plots the corresponding data in a strip chart.
5. When the simulation ends, select the **Reset** tool.

Find the range, R

Using animation tools, determine the time at which the stone encounters the plane. Use the time value to answer Question 1 in Module review on page 87.

To find the range:

1. From the Main Toolbox, select the **Animation** tool .
2. Select the **Play** tool.
3. When the stone makes contact with the plane, select the **Stop** tool .
4. Use the **Step Forward** 

 and **Step Backward** 

 tools to obtain the exact point at which the stone makes contact with the plane.
5. Note the time at which the stone makes contact with the plane in the plot. (ADAMS displays the time in the upper left corner of the ADAMS window.)
6. Select the **Reset** tool.

Create a point trace

Create a point trace to view the trajectory of the projectile during an animation.

To create a point trace:

1. On the **Review** menu, select **Animation Controls**.

   The Animation Controls dialog box appears.
2. Select **Icons**.
3. Set **No Trace** to **Trace Marker**.
4. Right-click the empty text box that appears, select **Marker**, and then select **Browse**.
5. From the Database Navigator, select **Stone.cm**.
6. Note that the marker name is entered into the text box, and then select the **Play** tool.
7. Close the Animation Control dialog box.

Find horizontal displacement

In ADAMS/PostProcessor, use the Plot Tracking tool to return the horizontal displacement corresponding to the time step determined earlier. Use the value to answer Question 2 in Module review on page 87.

To find the horizontal displacement:

1. Right-click a blank area inside the strip chart, and select **Transfer to Full Plot**.

   ADAMS/PostProcessor appears, replacing ADAMS/View.
2. Select the **Plot Tracking** tool .
3. Because you want to know the displacement when the stone makes contact with the plane, move the cursor over the plot until the value of X is equal to the time at which contact was made.
4. Note the value of displacement, Y. This is your answer for Question 2 in Module review on page 87.
5. Compare this value of Y to the results given in the closed-form solution on page 86.
6. To return to ADAMS/View, select the **ADAMS/View** tool.

Save your work

Save your work such that the file contains not only the model information, but also the results and plots.

To save your work:

1. From the **File** menu, select **Save Database As**.
2. In the **File Name** text box, enter **projectile**, and then select **OK**.

   An ADAMS binary file is created containing not only the model information but also the results and plots.
3. From the **File** menu, select **Exit**.

Optional tasks

Save your work before performing these tasks. **Do not** save your work after performing these tasks. If you must save the model after performing these tasks, give the model a different name.

To follow the stone during an animation:

1. Zoom in on the stone.
2. From the **Review** menu, select **Animation Controls.**

   Now change the reference frame while animating.
3. On the Animation Controls dialog box, change **Fixed Base** to **Base Part**. Select the part to which you want to fix the camera.
4. Go to the online ADAMS/View guides and look up the Animation Controls dialog box to read about the other functionality available.

ADAMS results

R = 3180 mm (Can vary slightly depending on several factors, most likely the sampling rate.)

Closed-form solution

Analytical solution:

The analytical solution for R, the range covered by the projectile, is as follows:

$x_o = 0 \quad x_f = R$

$y_o = 0 \quad y_f = 0$

$V_{x_o} = 6000 \times \cos 60° = 3000 \text{ mm/sec}$ t = time

$V_{y_o} = 6000 \times \sin 60° = 5196 \text{ mm/sec}$

$$y_f = y_o + V_{y_o} t - \frac{1}{2} g t^2$$

$$0 = 0 + 5196t - 0.5 \times 9806 \times t^2$$

$$0 = (5196 - 4905t)t$$

$$t = 1.06 \text{ sec}$$

$$x_f = x_o + V_{x_o} t$$

$$R = 0 + 3000 \times 1.06$$

$$R = 3180 \text{ mm}$$

Module review

1 At what time does the stone encounter the plane?

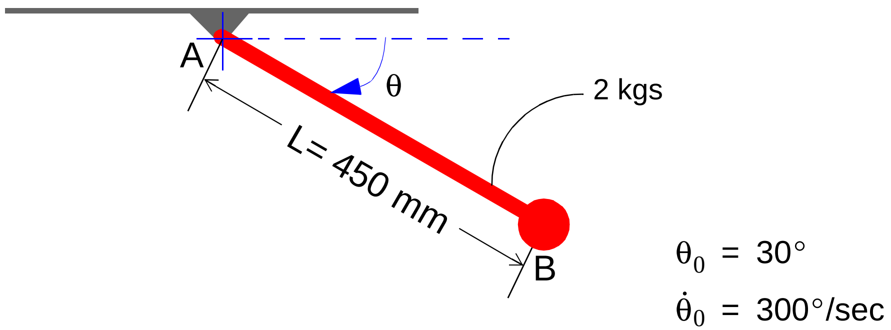
2 What is the range, R, as defined in the problem statement?

3 If a part's initial velocity conflicts with a system constraint, which will take precedence during a simulation?

4 What modifications would be necessary to convert the stone into a pendulum?

6 ONE DOF PENDULUM

Find the initial force supported by a pin at A for a bar that swings in a vertical plane, given the initial angular displacement (θ_0) and initial angular velocity ($\dot{\theta}_0$). Also, find the pendulum frequency.



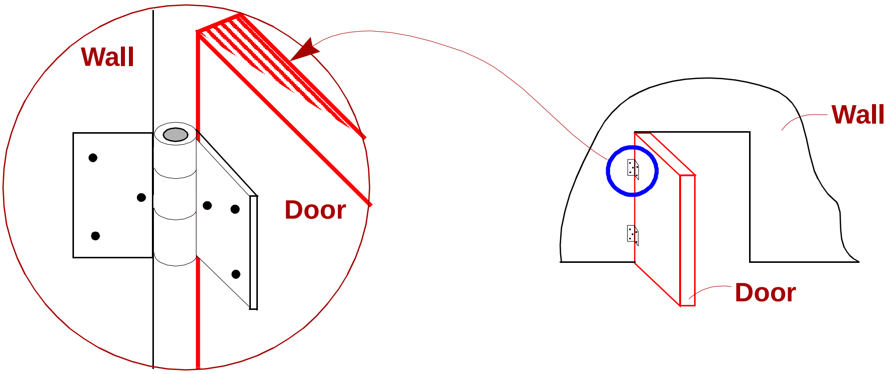
What's in this module:

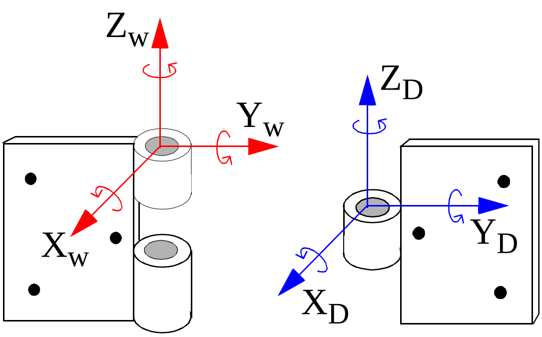
Constraints

Definition of a constraint

- Restricts relative movement between parts.
- Represents idealized connections.
- Removes rotational and/or translational DOF from a system.

Example





Translational constraints of the hinge

$$X_D - X_W = 0$$

$$Y_D - Y_W = 0$$

$$Z_D - Z_W = 0$$

Rotational constraints of the hinge

$$\Phi_D - \Phi_W = 0 \text{ (about x-axis)}$$

$$\theta_D - \theta_W = 0 \text{ (about y-axis)}$$

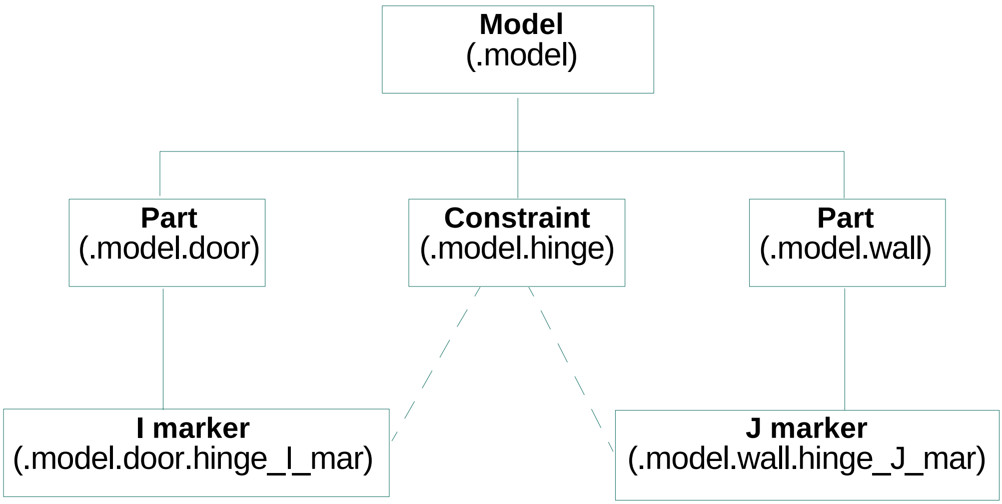
Therefore, ϕ_D and ϕ_W are free

Use of Markers in Constraints

Constraint equations in ADAMS

- Constraints are represented as algebraic equations in ADAMS/Solver.
- These equations describe the relationship between two markers.
- Joint parameters, referred to as I and J markers, define the location, orientation, and the connecting parts:
  - First marker, **I**, is fixed to the first part.
  - Second marker, **J**, is fixed to the second part.

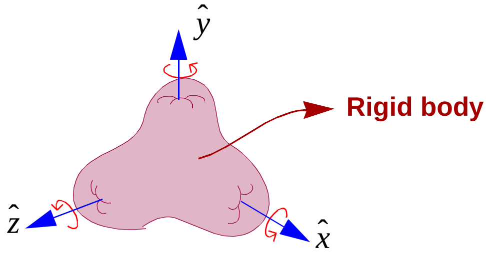
Anatomy of a constraint in ADAMS



Degrees of Freedom (DOF)

Constraints and DOF

- Each DOF in mechanical system simulation (MSS) corresponds to at least one equation of motion.
- A freely floating rigid body in three-dimensional space is said to have six DOF.
- A constraint removes one or more DOF from a system, depending on its type.



Determining the number of system DOF

- ADAMS will provide an **estimated** number of system DOF by using the Gruebler's Count:

$$\text{System DOF} = (\text{number of movable parts} \cdot 6 \ \text{DOF/ part})$$

$$- \sum_{i\,=\,\text{type}} [\#\ \text{Constraints} \cdot \#\ \text{DOF (Constraint)}]$$

- ADAMS also provides the **actual** number of system DOF, as it checks to see if:
  - Appropriate parts are connected by each constraint.
  - Correct directions are specified for each constraint.
  - Correct type of DOF (translational versus rotational) are removed by each constraint.
  - There are any redundant constraints in the system.

See also: DOF removed by a revolute joint on page 320

Joint Initial Conditions (ICs)

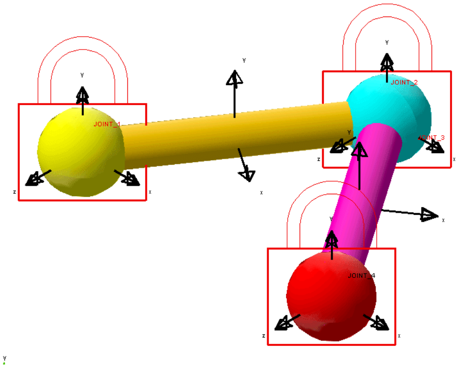
Characteristics of joint initial conditions

- You can specify displacement and velocity initial conditions for revolute, translational, and cylindrical joints.
- ADAMS uses the specified initial conditions of the joint while performing a simulation, regardless of any other forces acting on the joint.
- If you do not specify joint ICs, ADAMS calculates the conditions of the connecting parts while performing a simulation depending on the other forces acting on the joint.

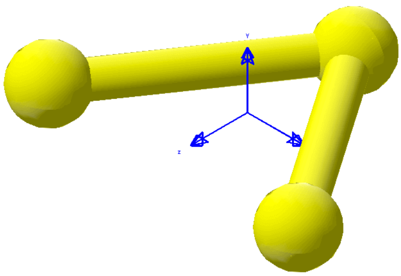
Question: What would happen if the joint initial conditions in a system were different from the part initial conditions?

Methods of attaching multiple geometry to a part

- **Using fixed joint** to constrain geometric objects.



- **Adding new geometry** to an existing part.



Note: ADADAMS/Solver handles simulations better if you merge geometry on a rigid part as opposed to constraining multiple parts.

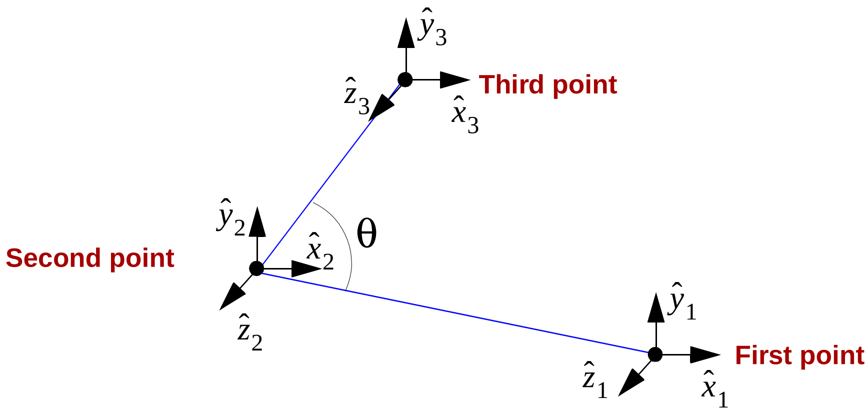
Question: When you merge geometry is the overlapping volume accounted for?

Angle Measures

Definition of angle measures

They are used to measure the included angle, θ:

- Between two vectors
- Defined by three markers
- Defined throughout a simulation



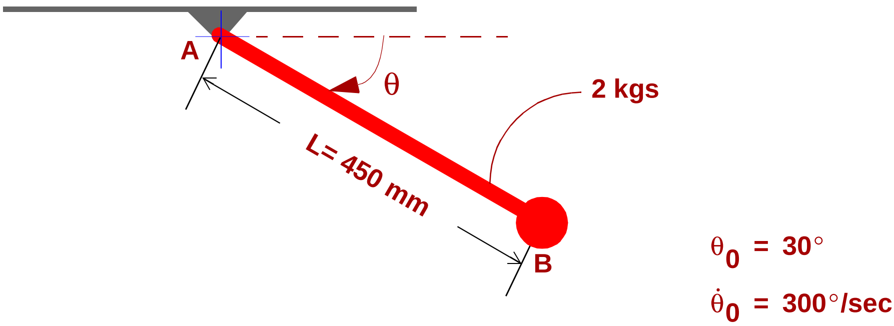
Notes:

- The units used for angle measures are in current ADAMS/View angle units (degrees or radians).
- The sign convention (+/-) is defined such that the first nonzero value is positive.

Workshop 6—One DOF Pendulum

Problem statement

Find the initial force supported by the pin at A for a bar that swings in a vertical plane, given the initial angular displacement (θ_0) and initial angular velocity ($\dot{\theta}_0$). Also, find the pendulum frequency.



Start the workshop

First, you'll start ADAMS/View from the directory *exercise_dir*/mod_06_pendulum and then you'll create a new model. Executing ADAMS/View in this directory ensures that all saved data gets stored here.

To start ADAMS/View and create model:

- Start ADAMS/View:
  - Create a new model named **pendulum**, with gravity set to **earth normal** (-global y), and units set to **MMKS - mm, Kg, N, s, deg**.
  - Set the directory to ***exercise_dir*/mod_06_pendulum**.

Build the pendulum link

Now, build the link section of the pendulum using the following parameters:

- Width: 20 mm
- Depth: 27.5mm
- Endpoints: (0, 0, 0) and (450, 0, 0)

To build the link:

1. Turn on the coordinate window.
2. From the Main Toolbox, right-click the **Rigid Body** toolstack, and then select the **Link** tool .
3. In the container:
   - Select **New Part**.
   - Select **Length**, and in the **Length** text box, enter **450 mm**, and then press **Enter**.
   - Select **Width**, and in the **Width** text box, enter **20 mm**, and then press **Enter**.
   - Select **Depth**, and in the **Depth** text box, enter **27.5 mm**, and then press **Enter**.
4. Using the mouse, select **0, 0, 0** and **450,0,0** as the endpoint locations.

   Tip: Use the Location Event (right-click away from the model) to help select the endpoints. When you right-click, the Location Event appears in the lower left corner of the ADAMS window. Enter the coordinates for the link in the upper text box and press **Enter**.

Build the sphere section

Next, build the sphere section of the pendulum using the following parameters:

- Add to Part
- Radius: 25 mm
- Centerpoint: 450, 0, 0

To build the sphere section:

1 From the Main Toolbox, right-click the **Rigid Body** toolstack, and then select the **Sphere** tool .

2 In the container:

- Select **Add to part**.
- Select **Radius**, and in the **Radius** text box, enter **25mm**, and then press **Enter**.

3 Using the mouse, select **PART_2** as the part to add to.

4 Using the mouse, select **450,0,0** as the location.

Rename the pendulum

Now you'll rename the pendulum from PART_2 to Pendulum.

To rename the pendulum:

1 Right-click the **link**, point to **Part:PART_2**, and then select **Rename**.

The Rename Object dialog box appears.

2 In the **New Name** text box, enter **.pendulum.pendulum**, and then select **OK**.

Set the mass of the pendulum

Now, set the mass of the pendulum to 2 kg, set all three inertias (Ixx, Iyy, Izz) to 0, and change the location of the center of mass.

To set the mass of the pendulum:

1 Right-click the **pendulum**, point to **Part: pendulum**, and then select **Modify**.

The Modify Rigid Body dialog box appears.

2 In the **Mass & Inertia defined by** text box, select **User Input**.

An alert box may appear. If it does, close it.

3 In the **Mass** text box, enter **2.0**, and then select **Apply**.

4 In the **Inertia** text boxes (I_{xx}, I_{yy}, I_{zz}), enter **0**.

5 Select **Apply**.

The Mass & Inertia defined by dialog box is still open.

6 Right-click the **Center of Mass Marker** text box, point to **pendulum.pendulum.cm**, and then select **Modify**.

7 In the **Location** text box, enter **450, 0, 0**.

8 Select **OK** in both dialog boxes.

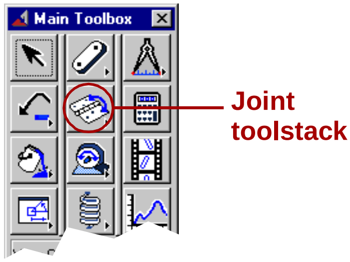
Your model should look like this:

Build the pivot

In this section, you'll build the pivot by creating a revolute joint between ground and the pendulum at location A, as shown in the figure on the page 96, and rename it Pivot.

To build the pivot:

1. From the Main Toolbox, right-click the **Joint** toolstack, and then select the **Revolute joint** tool .



2. In the container, select **2 Bod-1 loc** and **Normal to Grid**.
3. Select the **pendulum** as the **first body**.
4. Select the **ground** as the **second body**.
5. Select **0, 0, 0** as the **location**.

To rename the joint:

1. Right-click the **revolute** joint, point to **Joint:JOINT_1**, and then select **Rename**.
2. In the **New Name** text box, enter **.pendulum.pivot**, and then select **OK**.

Create measures

Create two object (joint) measures to track the force supported by the pin, resolved in the $\hat{x}_g$ and $\hat{y}_g$ directions.

To create object measures:

1. Right-click the **pivot** joint, point to **Joint:pivot**, and then select **Measure**.

   The Joint Measure dialog box appears.

2. In the dialog box:

   - In the **Measure name** text box, enter:

     pivot_force_x

   - Set **Characteristic** to **Force,** and select **X** as the **Component**.
   - Be sure **.pendulum.MARKER_4** and **Create Strip Chart** are selected.
   - Select **Apply**.

   A strip chart displays the force during simulation and animation.

3. In the dialog box:

   - In the **Measure name** text box, enter:

     pivot_force_y

   - Set **Characteristic** to **Force**, select **Y** as the **Component**,
   - Be sure **.pendulum.MARKER_4** and **Create Strip Chart** are selected.
   - Select **OK**.

   A strip chart displays the force during simulation and animation.

Create a reference marker

Create a marker on ground to use as a reference location for the angle measure you will create in the next section. Instead of right-clicking on the marker to change its name, you'll use the **Edit** menu.

To create a reference marker:

1. On the Main Toolbox, right-click the **Rigid Body** toolstack, and then select the **Marker** tool .
2. In the container, be sure that **Add to ground** and **Global XY** are selected.
3. Using the mouse, select **450,0,0** as the location.
4. With the marker still selected, from the **Edit** menu, select **Rename**.

   The Rename Object dialog box appears.
5. In the **New Name** text box, enter **.pendulum.ground.angle_ref**, and then select **OK**.

Create angle measure

Now, create the angle measure to track the angular displacement of the pendulum, θ.

To create an angle measure:

1. From the **Build** menu, point to **Measure**, point to **Angle**, and then select **New**.
2. In the **Measure Name** text box, enter **pend_angle**.
3. Right-click the **First Marker** text box, point to **Marker**, and then select **Pick**.
4. On the screen, pick a marker that is on the pendulum and at its end (for example, select the cm marker).

   Tip: Right-click the end of the pendulum to select the cm marker.
5. Right-click the **Middle Marker** text box, point to **Marker**, and then select **Pick**.
6. Pick a marker that is at the pivot location.
7. Right-click the **Last Marker** text box, point to **Marker**, and then select **Pick**.
8. Pick a marker that is on the ground and at the end of the pendulum (this will be the marker that was created in the previous section).

Specify initial conditions

In this section, you'll specify the following joint initial conditions:

- Displacement initial condition of $\theta_0 = 30^o$
- Initial velocity condition of $\dot{\theta}_0 = 300^o$/sec

To specify the initial conditions:

1 Right-click the **pivot** joint, point to **Joint:pivot**, and then select **Modify**.

2 Select **Initial Conditions**.

3 In the Joint Initial Conditions dialog box:

- Select **Rot. Displ** and, in the **Rot Displ.** text box, enter **-30**.
- Select **Rot. Velo.** and, in the **Rot Velo.** text box, enter **-300**.

4 Select **OK** in both dialog boxes.

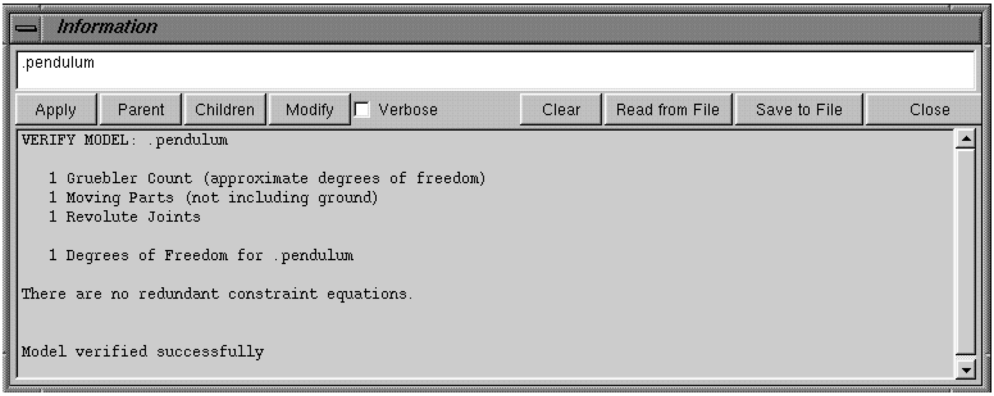
Verify your model

Before simulating your model, verify it.

To verify your model:

1 Select the **Verify** tool on the right side of the status bar.

The Information window appears as shown next:



You also receive a warning that the initial conditions for the joint position does not match the design configuration. This is what we expect.

2 Close the Information window.

Simulate your model

Run a simulation for 2 seconds.

To simulate your model:

1 From the Main Toolbox, select the **Simulation** tool.

2 In the container:

- Select **Default.**
- In the **End Time** text box, enter **2.0**.
- In the **Steps** text box, enter **100**.

3 Select the **Play** tool.

Determine global components

Now, determine the global components (x, y) of the initial force supported by the pivot. Use the value to answer Question 2 in Module review on page 109.

To determine global components:

1. Right-click the blank area inside the **pend_angle** strip chart, and then select **Transfer to Full Plot**.

   ADAMS/PostProcessor appears, replacing ADAMS/View.

2. Select the **Plot Tracking** tool .
3. Move the cursor over the plot at **t =0**.
4. In the area below the Main toolbar, note the value of Y.
5. In the dashboard, select **Clear Plot**.
6. From the **Measure** list, select **Pivot_force_X**.
7. Select **Surf**.
8. Move the cursor over the plot at **t =0**.
9. In the area below the Main toolbar, note the value of Y.
10. From the **Measure** list, select **Pivot_force_Y**.
11. Move the cursor over the plot at **t =0**.
12. In the area below the Main toolbar, note the value of Y.

Determine the frequency of the pendulum

Estimate the frequency by determining the period (seconds) and then inverting that value to obtain Hertz. This is the answer to Question 3 in Module review on page 109.

To determine frequency:

1. From the **Measure** list, select **pend_angle**.
2. Estimate the period of the curve.
3. Invert the period to find Hertz.
4. To return to ADAMS/View, select the **ADAMS/View** tool.

Save your work

To save your work:

1. Save your modeling session such that the saved file contains not only the model information but also the results and plots.
2. Exit ADAMS/View.

Optional tasks

Save your work before performing these tasks. **Do not** save your work after performing these tasks. If you must save the model after performing these tasks, give the model a different name.

To find the natural frequency of the pendulum automatically by performing an FFT on the plot of theta versus time:

1. Set up for the FFT by simulating the model based on current findings:

   - End time = 1.65 (approximate time of one period)
   - Steps = 127

   When preparing for an FFT operation, we recommend that the number of points be an even power of two (for example, 128, 256, 512, and so on). By solving the equation and asking for 127 steps, you will get 128 data points; 127 + 1 for the initial conditions.

   You should get the same frequency as you did by calculating it manually.

   The peak value of the resultant curve is the natural frequency.

2. In ADAMS/PostProcessor, from the **Plot** menu, select **FFT**.

   The FFT dialog box appears.

3. Accept the default values in the FFT dialog box, and then select **Apply**.

   You should get the same frequency as you did by calculating it manually.

   The peak value of the resultant curve is the natural frequency.

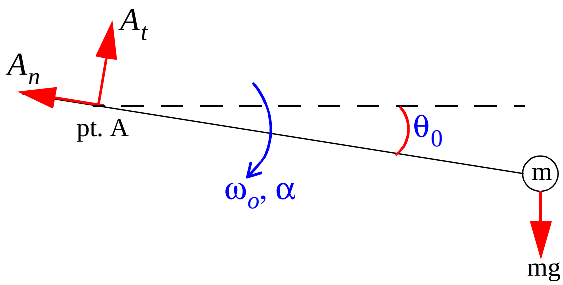
4. Return to ADAMS/View.

5. Exit ADAMS/View.

ADAMS results

- Horizontal force supported by the pivot at A = -29.86 N.
- Vertical force supported by the pivot at A = 17.24 N.

Closed-form solution



$$I_A = I_{zz} + mr^2 \qquad \omega = 300°/\text{sec}$$

$$I_A = 0 + mL^2 \qquad \omega = 5.24\text{rad/sec}$$

The analytical solution for the force supported by the pivot at A when $\theta_0 = 30^o$ and $\dot{\omega}_0$ = 300 degrees/sec:

$$\Sigma M_A = I_A\alpha \quad -mg(L\cos 30) = (mL^2)\alpha$$

$$g\cos 30 = L\alpha$$

$$\alpha = -\frac{g}{L}\cos 30$$

$$\alpha = -18.88\text{rad/sec}^2$$

$$\Sigma F_t = mr\alpha \; mg\cos 30 - A_t = mL\alpha$$

$$A_t = m(g\cos 30 - L\alpha)$$

$$A_t = 0N$$

$$\Sigma F_n = mr\omega^2 \; A_n - mg\sin 30 = mL\omega^2$$

$$A_n = m(g\sin 30 + L\omega^2)$$

$$A_n = 34.53N$$

Horizontal force supported by pivot at: $A = -A_n \cos 30$

$F_x = -29.90N$

Vertical force supported by pivot at: $A = A_n \sin 30$

$F_y = 17.27N$

Module review

1 What are the global components of the initial force supported by the pivot?

2 What is the frequency of the pendulum using the initial conditions in the problem statement?

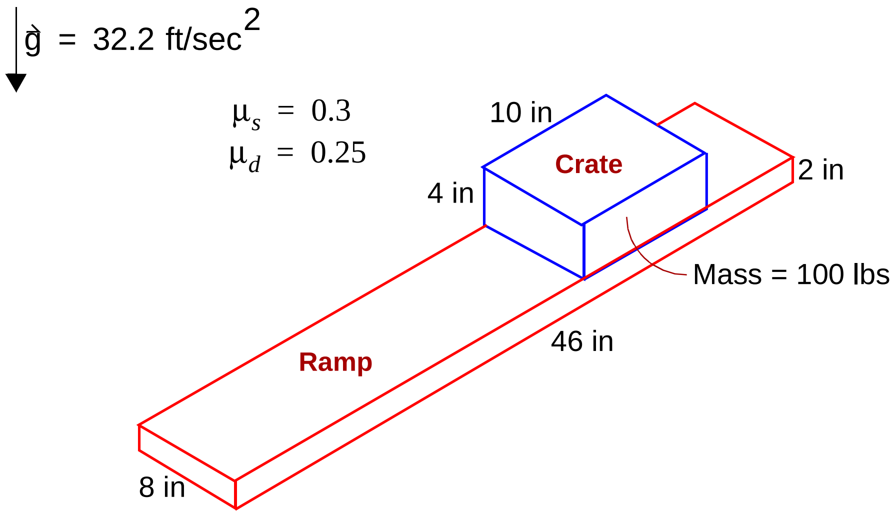
3 If the initial velocity of a part can be set through a connecting joint *and* the part itself, which will ADAMS/View use if they are both set?

4 If a model (human_hip) had two parts (femur and hip_bone) constrained by a joint, I and J markers would be created by ADAMS. If both markers were named MAR_1, what would the complete name of the I and J markers be?

5 Can the I and J markers for a joint belong to the same part? Why?

7 INCLINED PLANE

Find the minimum inclination that will ensure that a crate slides off an inclined plane, using the properties shown next:



What's in this module:

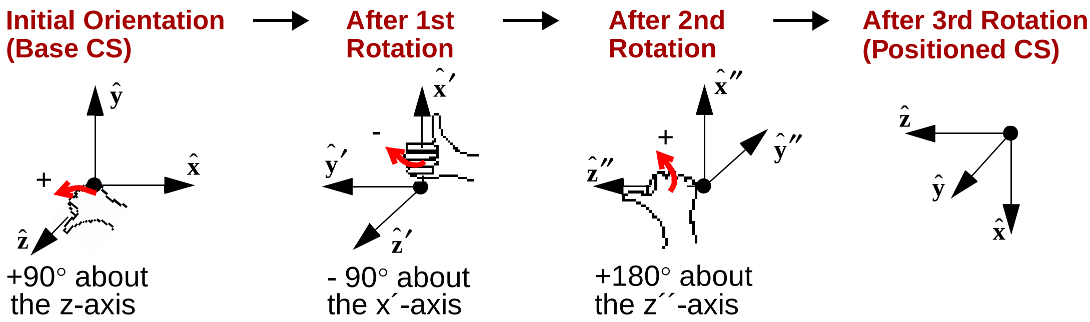
Euler Angles (Rotation Sequence)

Definition of Euler angles

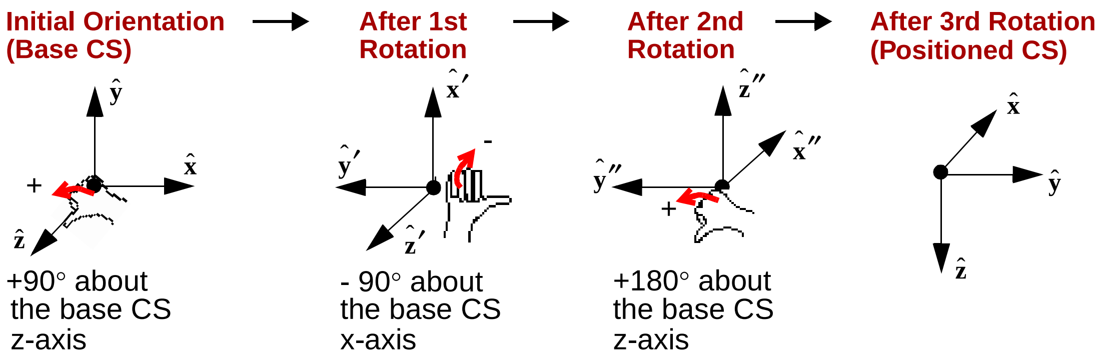
- ADAMS/View uses three angles to perform three rotations about the axes of a coordinate system.
- These rotations can be **space-fixed** or **body-fixed** and are represented as Body [3 1 3], Space [1 2 3], and so on, where:
    - 1 = x axis
    - 2 = y axis
    - 3 = z axis

  For rotation about these axes, use the right-hand rule
- Default in ADAMS is Body [3 1 3].

Example of body [3 1 3]: [90°, -90°, 180°]

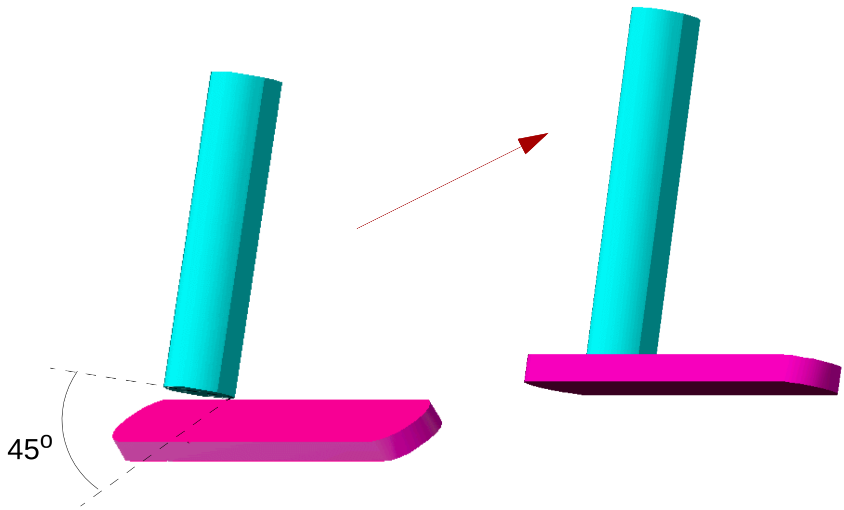


Example of space [3 1 3]: [90°, -90°, 180°]:



To rotate objects about an axis in ADAMS/View, specify:

- The objects to rotate.
- The axis about which the objects are rotated.
- The angle through which the objects are rotated.



Note: Be careful with the sign of the angle. ADAMS/View uses the right-hand rule. You can rotate several objects at once about the same axis.

Modeling Friction

Joint friction can be applied to:

- Translational joints
- Revolute joints
- Cylindrical joints
- Hooke/Universal joints
- Spherical joints

Friction forces (F_f)

- Are independent of the contact area between two bodies.
- Act in a direction opposite to that of the relative velocity between the two bodies.
- Are proportional to the normal force (N) between the two bodies by a constant (μ).

$$F_f = \mu N$$

Modeling Friction...

Phases that define friction forces

- Stiction
- Transition
- Dynamic

Idealized case

- **Stiction:** $|V_{rel}| = 0$

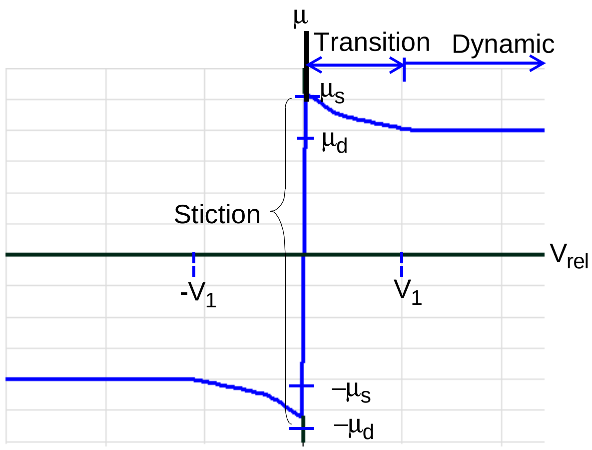
  $0 < \mu < \mu_s$

- **Transition:** $0 < |V_{rel}| = V_1$

  $\mu_d < \mu < \mu_s$

- **Dynamic:** $V_1 < |V_{rel}|$

  $\mu = \mu_d$



ADAMS case

- **Stiction:** $|V_{rel}| < \Delta V_s$

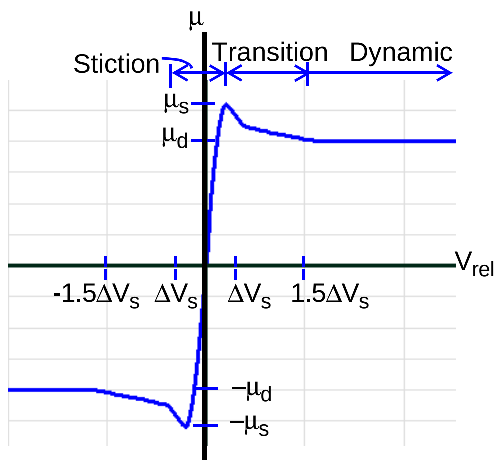
  $0 < \mu < \mu_s$

- **Transition:** $\Delta V_s < |V_{rel}| < 1.5\Delta V_s$

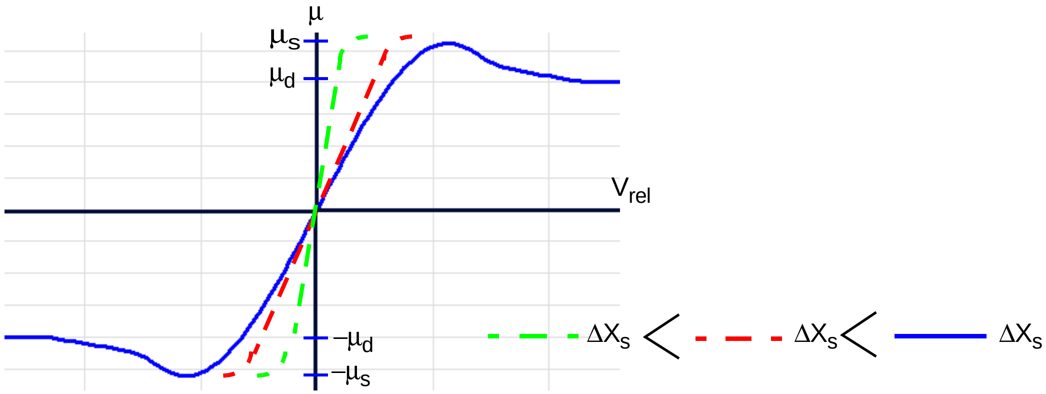
  $\mu_d < \mu < \mu_s$

- **Dynamic:** $\Delta V_s < |V_{rel}|$

  $\mu = \mu_d$



Effect of maximum deformation on friction



Input forces to friction

- Always include preload and reaction force.
- Bending and torsional moment are possible (however, advanced uses of joint friction are beyond the scope of this course).

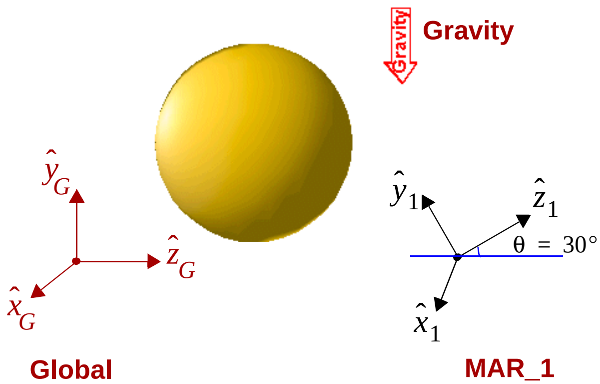
Measures in LCS

Measures can be represented in:

- Global coordinate system (GCS) (default)
- A marker's local coordinate system (LCS)

Example

- When a ball falls due to gravity:



- Acceleration due to gravity in the GCS using $\hat{x}_g, \hat{y}_g, \hat{z}_g$ symbols to represent the global x, y, and z components is:

$$\vec{g} = (0\hat{x}_g - 9.81\hat{y}_g + 0\hat{z}_g)\frac{m}{s^2}$$

- Acceleration due to gravity in MAR_1's coordinate system is:

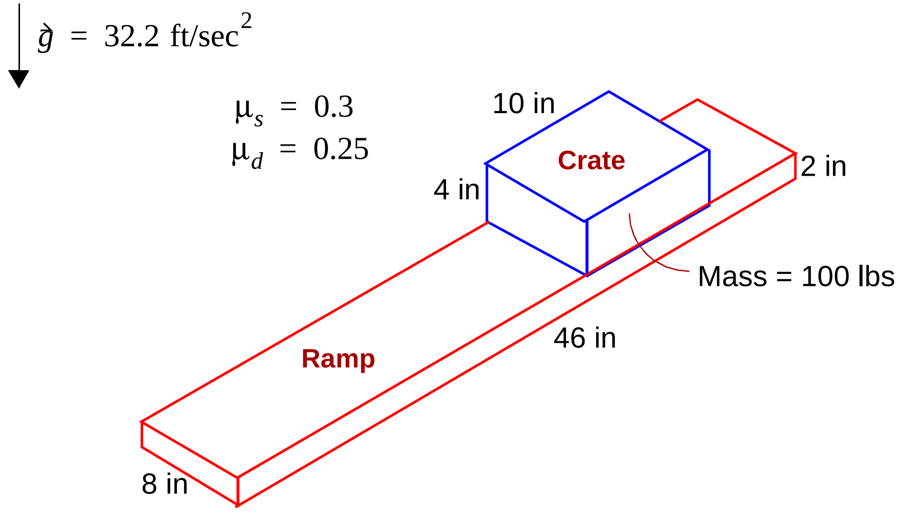
$$\vec{g} = (0\hat{x}_1 - (9.81\sin 30°)\hat{y}_1 - (9.81\cos 30°)\hat{z}_1)\frac{m}{s^2}$$

$$\vec{g} = (0\hat{x}_1 + -4.91\hat{y}_1 - 8.50\hat{z}_1)\frac{m}{s^2}$$

Workshop 7—Inclined Plane

Problem statement

Find the minimum inclination that will ensure that a crate slides off an inclined plane, using the properties shown next:



Start the workshop

First, you'll start ADAMS/View from the directory exercise_dir/mod_07_inclined_plane and then create a new model.

To start the workshop:

- Start ADAMS/View:
  - Set the directory to ***exercise_dir*/mod_07_inclined_plane**.
  - Create a new model named **inclined_plane**, with gravity set to **Earth Normal** (-global y), and units set to **IPS - inch, lbm, lbf, s, deg**.

Adjust the working grid

Now adjust the spacing and orientation of the working grid.

To adjust the spacing of working grid:

1. From the **Settings** menu, select **Working Grid**.
2. Set **Spacing** to **1"** in the **x** and **y** direction.

To adjust the orientation of the working grid:

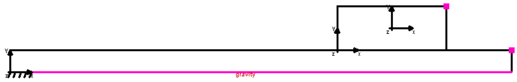
- Make sure that the working grid is oriented along the global XY direction (default setting when you open ADAMS/View). The Set Orientation text box allows you to choose Global XY, YZ, XZ, or custom orientation.

Build the parts

When creating parts, use an inclination angle of 0^o. You will rotate the parts to the desired inclination angle later in the exercise. Be sure to set the ramp geometry to be on ground.

To build the parts:

1. Build the ramp geometry using the following parameters:
   - **Length: 46"**
   - **Height: 2"**
   - **Depth: 8"**
   - **On ground**
2. Build the crate geometry using the following parameters:
   - **Length: 10"**
   - **Height: 4"**
   - **Depth: 8"**
   - **New part**



To modify the parts:

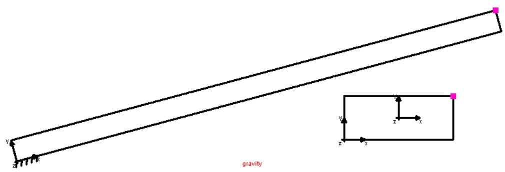
1. Rename the crate part and the ramp geometry as shown in the figure on page 118.
2. Set the **Mass** of the crate to **100 lbm**:
   - Right-click the crate, point to **Part:crate**, and then select **Modify**.
   - Set **Mass & Inertia defined by:** to **User Input**.
   - In the **Mass** text box, enter **100 lbm**.
   - Select **OK**.

Set the model's inclination angle

Now you are going to rotate the model 15^{o}. Because the ramp is on ground and you cannot rotate ground, to rotate the ramp, you are going to change the orientation of the ramp's corner marker to 15, 0, 0. The orientation of this marker sets the orientation for the ramp. You'll use the **Rotate** tool to rotate the crate since it is not on ground. You'll rotate the crate about the same axis that you rotated the ramp about.

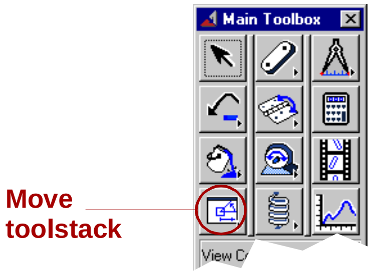
To rotate the ramp to $\theta = 15^{o}$:

1. Right-click the marker, point to the marker name, and then select **Modify**.
2. In the **Orientation** text box, change 0,0,0 to **15,0,0**.



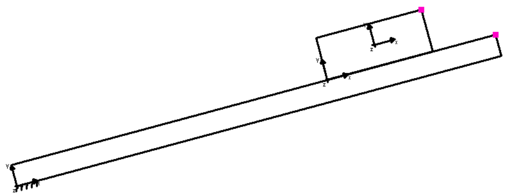
To rotate the crate 15°:

1 In the Main Toolbox, from the **Move** tool stack, select the **Align & Rotate** tool .



2 In the container, in the **Angle** text box, enter **15** as the increment by which to rotate the crate.

3 Select the crate as the object to rotate and the ramp’s base marker as the axis about which to rotate.



Constrain the model

Now you'll create a translational joint between the ramp and the crate.

To constrain the model:

1. From the Main Toolbox, right-click the **Joint** toolstack, and then select the **Translational** tool .
2. Set the location of the translational joint at the crate geometry's base marker.
3. Set the vector so it points up the ramp.

Take measurements

To create a measure:

- Create an object (part) measure for the crate's acceleration along the ramp as you did in To create object measures: on page 101.

Verify the mechanism (crawl-walk-run)

To verify the mechanism:

1. Simulate the model for **1 second** and **50 steps**.
2. Find the value of the crate's (constant) acceleration.

   To verify this value, see Without friction in the Closed-form solution on page 127.

   If the values do not match, check the units in the closed-form solution and in the model.

Refine the model

In this section, you'll add joint friction to the translational joint using the μ_s, μ_d values from the problem statement on page 118. You'll then simulate the model to see if the crate slides off the ramp.

> ***Tip:*** Be sure that the only friction forces to consider are those resulting from reaction forces.

To add friction and simulate:

1 Display the crate's modify dialog box (right-click the crate, point to **Joint:JOINT_1**, and then select **Modify**).

2 In the lower right corner of the Modify dialog box, select the **Friction** tool  .

3 Simulate the model and note if the crate slides off the ramp.

Rotate the ramp and crate to theta = 20^{o}

To rotate the ramp and crate, you'll create a group consisting of the crate part, joints, and geometry making up the ramp. You'll then select that group and rotate it.

To create a group:

1. From the **Build** menu, select **Group**.
2. Make a group containing:
   - The crate part.
   - The joint.
   - All of the geometry (including markers) on the ramp, but not the ramp part itself, because, remember, you cannot rotate ground.

To select the group:

1. From the **Tools** menu, select **Database Navigator**.
2. Set the Database Navigator's top option menu to **Select List**.
3. From the buttons near the bottom right side of the window, select **Clear** to clear the current select list.
4. Using the tree list on the left, find and select your group.
5. From the buttons near the bottom right side of the window, select **Add** to add your group to the select list.
6. Select **Apply**.

   All the elements that make up your group should now be highlighted.
7. Leave the select list open for future use, but move it out of the way.

To rotate the selected group:

1. In the Main Toolbox, from the **Move** tool stack, select the **Align & Rotate** tool .
2. Check the **Selected** box, because you want to rotate the items that you already have selected (highlighted).

   When you rotate with the **Align & Rotate** tool, it rotates in increments. So, if you have already rotated to 15 degrees and you want to arrive at 20 degrees, you should enter a 5 as the angle.
3. Again, use the ramp's base marker as the axis about which to rotate the whole group.

   Note: In this case, you can ignore any warnings associated with the friction element.

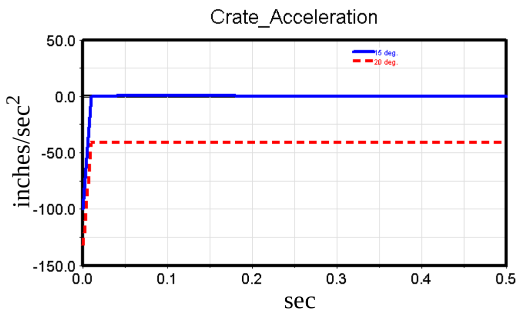
Find the inclination angles between which the crate starts to slide

To find the inclination angle:

1. Simulate the model and note if the crate slides off the ramp.

   For an end time of 0.5 seconds, verify that the crate acceleration versus time strip chart matches the adjoining figure.

   The initial spike is due to the acceleration (due to gravity) present at t=0.



2. Through trial and error, find the approximate angle (within 0.5º) at which the crate starts to slide off the ramp. Use it to answer Question 1 in Module review on page 129.

Save your work

Save your model and exit ADAMS/View.

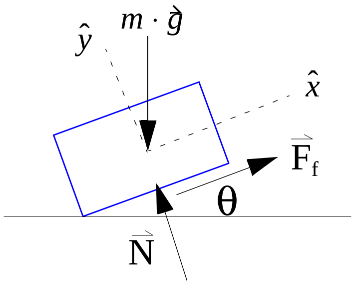
Optional tasks

Save your work before performing these tasks. **Do not** save your work after performing these tasks. If you must save the model after performing these tasks, give the model a different name.

To view an animation and its corresponding plot simultaneously in ADAMS/PostProcessor:

1. Open ADAMS/PostProcessor.
2. Create two views by right-clicking the **Window layout** tool  on the toolbar and selecting the **2 Views, side by side** tool .
3. Select the left view.
4. Plot crate acceleration versus time.
5. Select the right view, and then change the **Plotting** menu to **Animation**.
6. Right-click the right view, and then select **Load Animation**.
7. Run the animation.

Closed-form solution



Without friction:

$$\Sigma F_x = ma_x : -mg \cdot \sin\theta = ma_x$$

$$a_x = -g\sin\theta$$

$$\text{For } \theta = 15°, a_x = -32.2\sin(15°)$$

$$a_x = -8.33 \text{ ft/sec}^2$$

With friction:

$$\Sigma F_y = 0 : -mg \cdot \cos\theta + N = 0$$

$$N = mg \cdot \cos\theta$$

Maximum angle (θ_{max}) at which crate will not slide:

$$\Sigma F_x = 0 : F_f - mg \cdot \sin\theta_{max} = 0$$

$$\mu_s \cdot N - mg \cdot \sin\theta_{max} = 0$$

$$\mu_s \cdot mg \cdot \cos\theta_{max} - mg \cdot \sin\theta_{max} = 0$$

$$\mu_s - \tan\theta_{max} = 0$$

$$\theta_{max} = \text{atan}(\mu_s) = \text{atan}(0.30) = 16.7°$$

Once the crate starts sliding,

$$\Sigma F_x = ma_x : F_f - mg \cdot \sin\theta = ma_x$$

$$\mu_k \cdot N - mg \cdot \sin\theta = ma_x$$

$$\mu_k \cdot mg \cdot \cos\theta - mg \cdot \sin\theta = ma_x$$

$$\mu_k \cdot \cos\theta - \sin\theta = \frac{a_x}{g}$$

$$a_x = (\mu_k \cos\theta - \sin\theta) \cdot g$$

For $\theta = 20^o$, $a_x = (0.25 \cdot \cos 20° - \sin 20°) \cdot 32.2 \text{ ft/sec}^2$

$$a_x = -3.45 \text{ ft/sec}^2$$

ADAMS results

- At angle $\theta = 15^o$, a = $6.63e^{-5} \approx 0$
- At angle $\theta = 20^o$, the crate accelerates down the inclined plane at a = -41.35 in/sec^2 (-3.45 ft/sec^2)
- Based on the angular increments of 0.5^o, $(16.5^o < \theta_{max} < 17.0^o)$

Module review

1 What is the approximate angle (within 0.5^{o}) at which the crate starts to slide off the ramp?

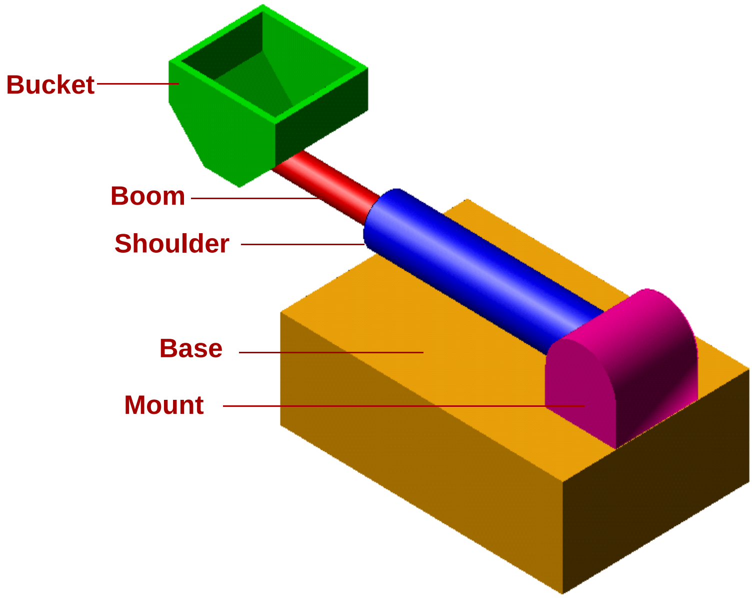
2 Can you apply friction to all joints in an automated way?

3 What are I and J markers?

4 If a joint with friction enabled crosses its stiction threshold velocity (ΔV_s), how does the maximum stiction displacement (ΔX_s) affect the system?

8 LIFT MECHANISM I

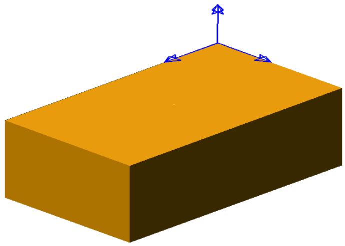
Use ADAMS/View to create each moving part of the lift mechanism shown next:



What's in this module:

Properties of geometry

- It must belong to a part and moves with the part.
- It is used to add graphics to enhance the visualization of a part.
- It is not necessary for performing simulations.
- Locations and orientations are defined indirectly by parts using anchor markers.



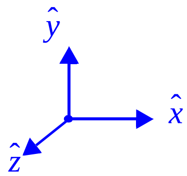
Note: If you move an anchor marker, all associated geometry moves with it. Conversely, anchor markers move when you move the associated geometry.

Types of geometry in ADAMS/View

- Construction geometry
  - Includes objects that have no mass (spline, arc, and so on).
  - Is used to define other geometry.
- Solid geometry
  - Includes objects with mass (box, link, and so on).
  - Can be based on construction geometry.
  - Is used to automatically calculate mass properties for the parent part.

Construction Geometry Properties

Marker geometry



Has:

- Anchor marker, which is itself
- Parent: part
- Orientation and location

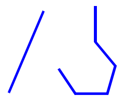
Point geometry

Has:

- No anchor marker
- Parent: part
- Location

Construction Geometry Properties ...

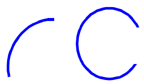
Polyline geometry



Has:

- No anchor marker
- Parent: part
- One line or multiple lines
- Open or closed
- Length, vertex points, and angle

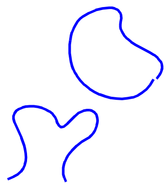
Arc geometry



Has:

- Anchor marker
- Parent: part
- Start and end angle, radius

Spline geometry

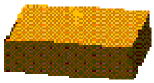


Has:

- Anchor marker
- Parent: part
- Segment count, open/closed, points

Solid Geometry

Block geometry



Has:

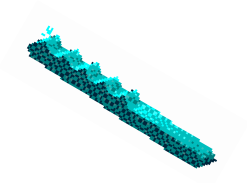
- Anchor marker, which is the corner marker
- Parent: part
- Length (x), height (y), depth (z) with respect to corner marker

Torus geometry

Has:

- Anchor marker, which is the center marker
- Parent: part
- Radius of ring (xy plane), radius of circular cross section ($\perp$ to xy plane)

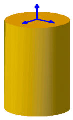
Extrusion geometry



Has:

- Anchor marker, which is the reference marker
- Parent: part
- Open/closed profile, depth, forward/backwards

Cylinder geometry

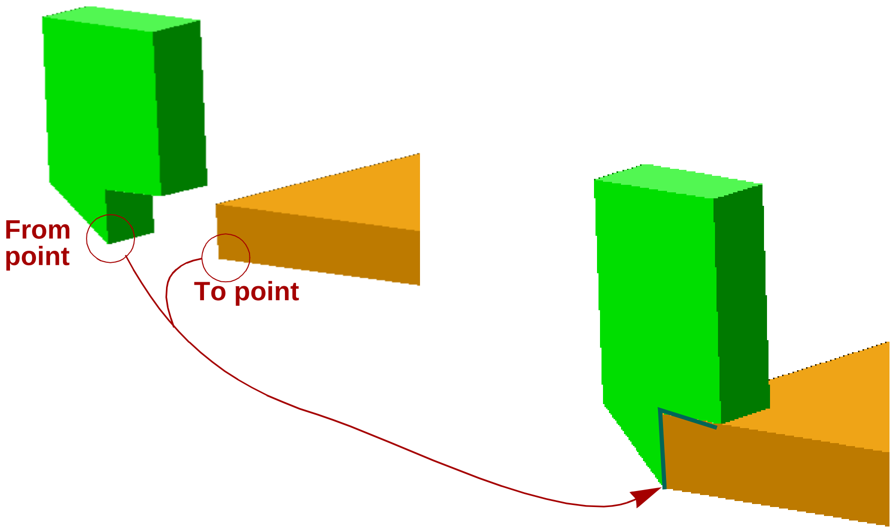


Has:

- Anchor marker, which is the center marker (placed at first end)
- Parent: part
- Length (z), radius

To move objects in ADAMS/View, specify:

- The object being moved (or copied).
- And:
    - Either, a point on the object, and the location to which the selected point will be moved.
    - Or, a vector and a distance along the vector.

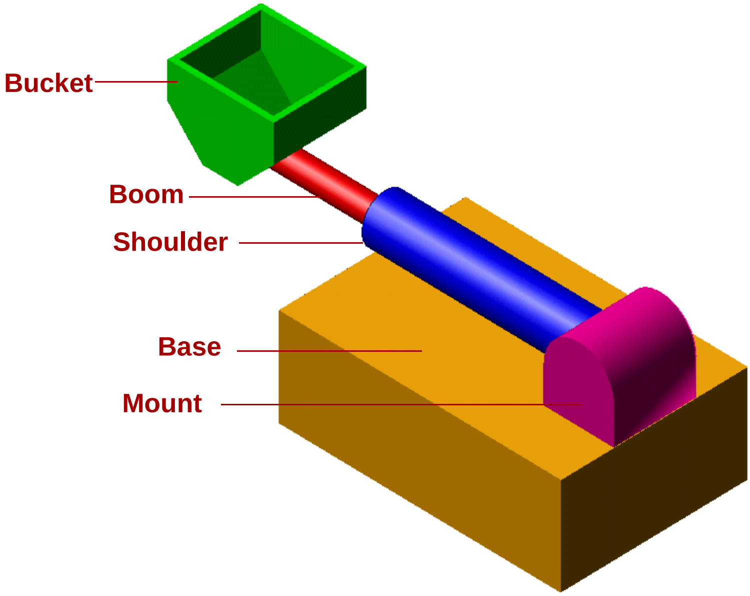


The moved object maintains its orientation.

Workshop 8—Lift Mechanism I

Problem statement

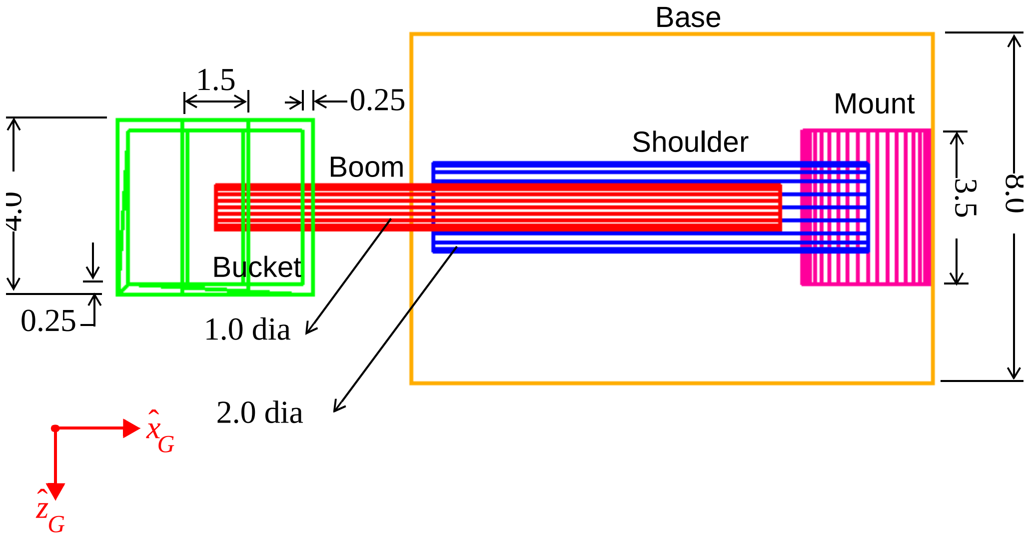
Use ADAMS/View to create each moving part of the lift mechanism shown next:



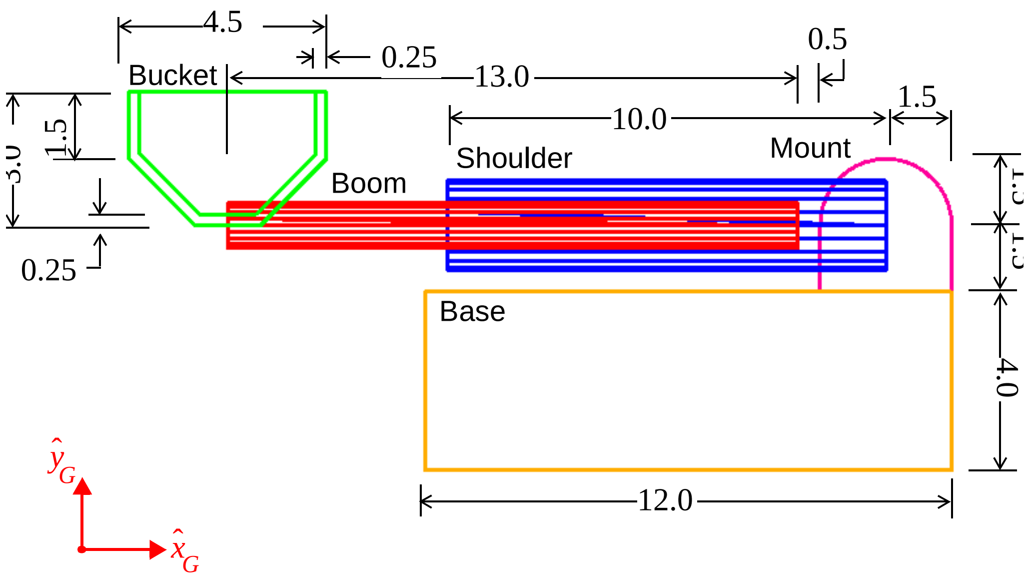
Background mechanism information

- The following diagrams provide the dimensions for building the lift mechanism.
- All units are in **meters**.

Top View of Lift Mechanism



Front View of Lift Mechanism



Tips before you start

- Check the three-dimensional view of the model at regular intervals to verify that the parts are being placed in the right location.
- Rename parts as soon as you build them.

Start the workshop

First, you'll start ADAMS/View from the directory *exercise_dir*/mod_08_lift_mech_1 and create a new model.

To start the workshop:

- Start ADAMS/View:
  - Set the directory to ***exercise_dir*/mod_08_lift_mech_1**.
  - Create a new model named **lift_mech**, with gravity set to **Earth Normal (-Global Y)**, and units set to **MKS - m, Kg, N, s, deg**.

Set up the working environment

Now you'll set up the ADAMS environment to make it easier to build the model.

To set up the working environment:

1. Adjust the grid based on the measurements given in the images on page 139.
2. Adjust icon sizes so you can see them because this model is in meters and, by default, the screen icons are set for models in millimeters. To adjust screen icons, from the **Settings** menu, select **Icons**, and then set a size for the icons.

Build all parts except for the bucket:

In this section, you'll create all the parts except the bucket. For information on how to build the parts, refer to the diagrams in Background mechanism information on page 139.

To build the parts:

1. Build the **base** part.

   Tip: Note the orientation of the block with respect to the xy plane.

2. Build the **mount** part by creating a block and then applying fillets using the **Fillet** tool.

3. Inspect your model. Note that the mount must be centered on the base. If necessary, use the vector option of the **Point-to-Point** tool on the **Move** toolstack to slide the mount along the base.

4. Build the **shoulder** part.

   Tip: Before building the shoulder, set the working grid to cut through the center of the block.

5. Build the **boom** part.

   Tip: Use the Location Event to start the cylinder two units over from the mount center-of-mass (cm) marker.

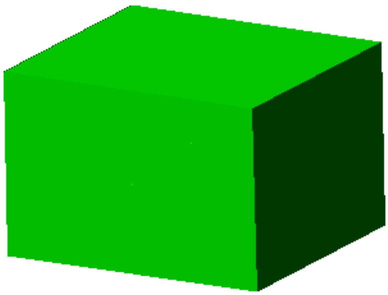
Build the bucket

Now build the bucket.

To build the bucket:

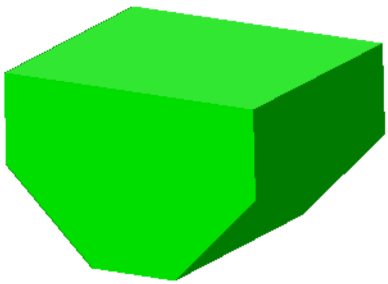
1 Build a block with the largest dimensions of the bucket:

- **Length: 4.5 m**
- **Height: 3.0 m**
- **Depth: 4.0 m**



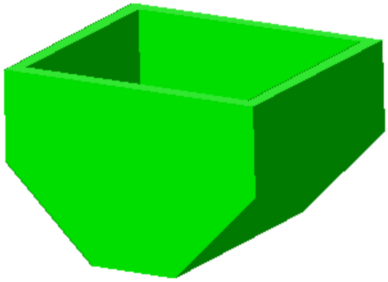
2 Chamfer the front and back, bottom corners of the block using the **Chamfer** tool :

- **Width: 1.5 m**



3 Hollow out the solid bucket using the **Hollow** tool:

- **Thickness: 0.25 m**
- Pierce the top face of the block



Check model topology by parts

To check model topology:

- Check model topology by parts to ensure that there are no floating parts that are not accounted for.

  There should be six parts, including ground.

Save your work

To save your work:

- Save your work such that the saved file contains only the model topology and not the results.

Optional tasks

To refine the geometry of the lift mechanism:

1 Using the **Torus** tool, add tires to the lift mechanism.

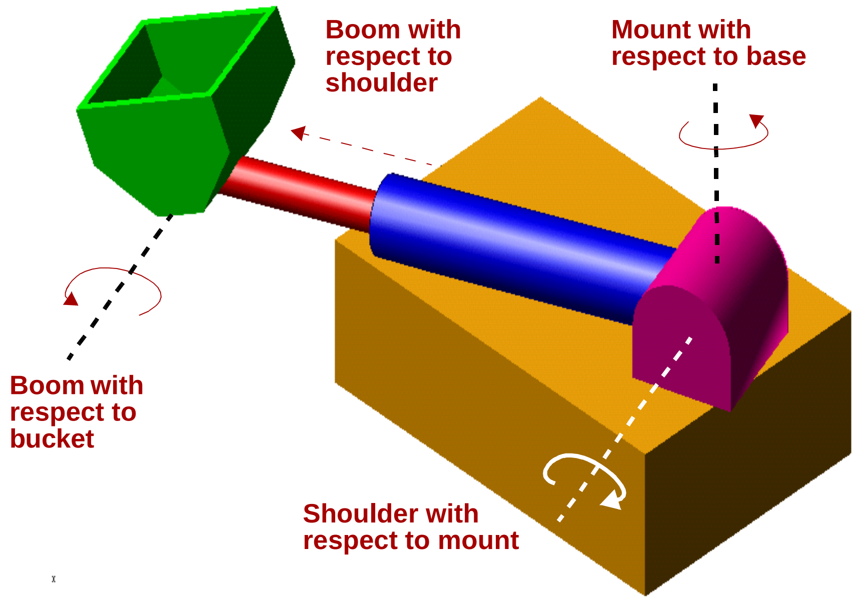
2 Using the **Fillet** tool, round the edges of the base.

Module review

1 What is the basic difference between construction geometry and solid geometry?

9 LIFT MECHANISM II

Constrain the lift mechanism model by adding joints and motions as shown next:



What's in this module:

Applying Motion

ADAMS/View provides two types of motions

- Joint motion
- Point motion

Joint motion

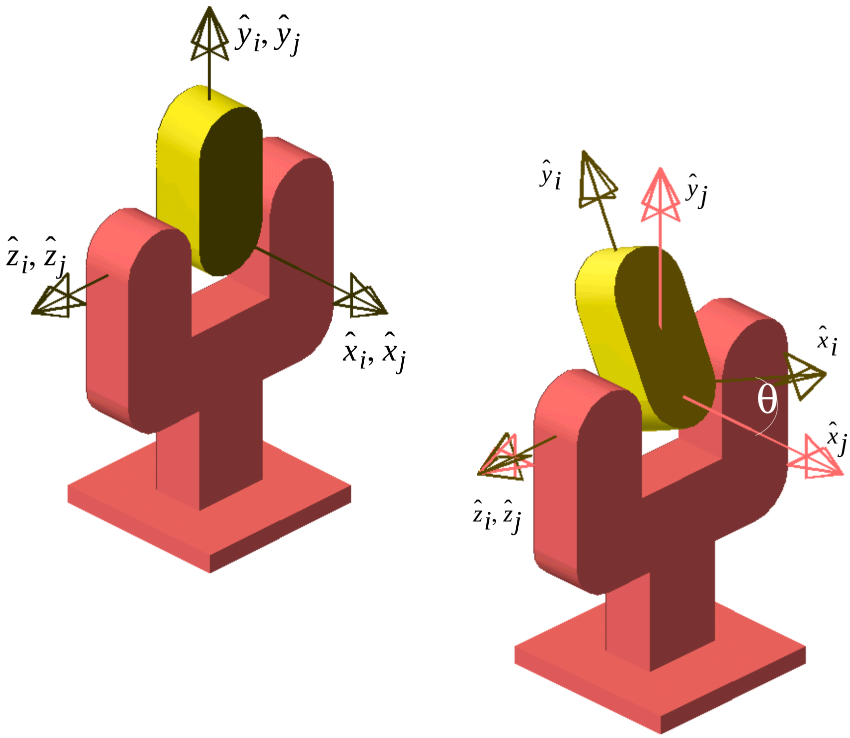
- There are two types:
  - Translational: applied to **translational** or **cylindrical** joints (removes 1 DOF).
  - Rotational: applied to **revolute** or **cylindrical** joints (removes 1 DOF).
- You define the joint to which motion is applied.
- ADAMS automatically uses the joint's I and J markers, bodies, and single DOF.
- You define function for magnitude.

Questions: How does a motion remove DOF?

Does this mean that a motion is considered a constraint?

Marker usage in joint motions

- The I and J markers (and, therefore, the parts to which they belong) referenced in the joint move with respect to each other as follows:



- The I and J markers overlap when motion $\theta_t = 0$.
- During simulation, the z-axes of both markers are aligned.
- You can define motion magnitude as a:
  - Displacement
  - Velocity
  - Acceleration function of time

Definition of functions in ADAMS

- You use functions to define magnitudes of input vectors used in:
  - Motion drivers
  - Applied forces
- Functions can depend on time or other system states, such as displacement, velocity, and reaction forces.
- Every function evaluates to a single value at each particular point in time.
- Motion drivers can only be a function of time:

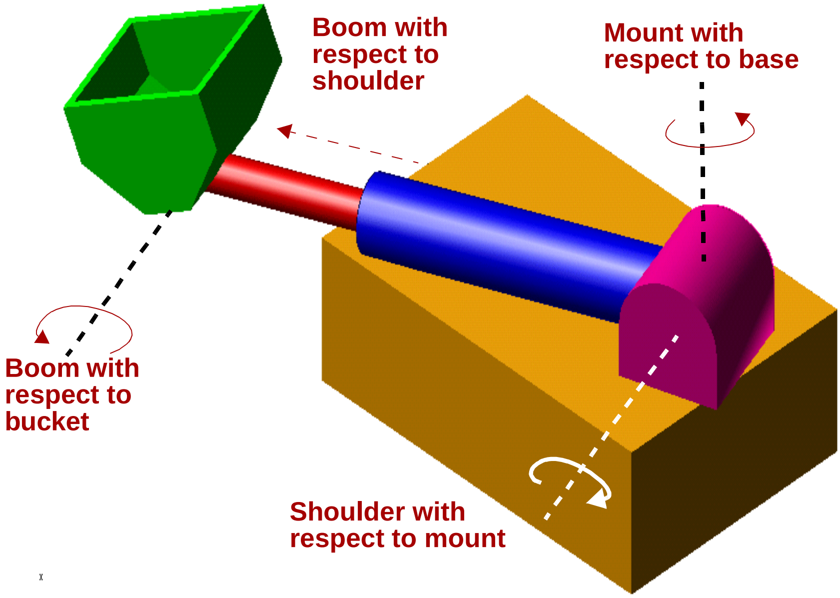
  M = f(time)
- Functions defining motion driver magnitudes can be:
  - Displacement (time)
  - Velocity (time)
  - Acceleration (time)

Note: You can use the Function Builder to create and verify functions in ADAMS/View. To access the Function Builder, right-click any text box that expects a function.

Workshop 9—Lift Mechanism II

Problem statement

Constrain the lift mechanism model by adding joints and motions as shown next:



Model description

In this workshop, you use the model you built in the Lift Mechanism I module.

Tips before you start

- Use the above figure to find out what type of constraints you need.
- Simulate the model at regular intervals to check the constraints.
- Rename joints as soon as you create them.
- Adjust icon sizes whenever necessary (see Set up the working environment, on page 140 of Workshop 8—Lift Mechanism I.)

Workshop 9—Lift Mechanism II...

Start the workshop

Note that the file for this exercise is not in the current working directory.

To start the workshop:

- Start ADAMS:
  - Set the directory to ***exercise_dir*/mod_09_lift_mech_2**.
  - From the directory ***exercise_dir*/mod_08_lift_mech_1**, import the model that you created in the previous module.

  If you need a fresh copy of the model, import the command file lift_mech_I_completed.cmd from the directory *exercise_dir*/mod_08_lift_mech_1/completed.

Constrain the parts

In this section, you'll constrain the parts that you created in the previous workshop. The figure in the Problem statement, on page 149, shows how you should constrain the parts.

To constrain the parts:

1. Use a fixed joint  to fix the **base** to **ground**.
2. Constrain **mount** to **base**.

   Tip: Use the options **2 Bodies - 1 Loc** and **Pick Feature** and constrain the mount to the base at the mount's cm marker.
3. Constrain **shoulder** to **mount**.

   Tip: Right-click to select the cylinder's anchor marker.
4. Constrain the **boom** to the **shoulder**.
5. Constrain the **bucket** to the **boom**.

   Tip: Use the end point of the cylinder.

Verify the model (crawl-walk-run)

Before continuing, check your work by checking model topology and by performing a simulation.

To verify your model:

1 Check model topology by constraints to ensure that all the parts are constrained as expected.

2 Perform a simulation.

Are the visual results of the simulation (the animation), what you expected?

Add joint motions to your model

To add joint motions:

1 Add a rotational joint motion  to the mount-to-base joint such that:

D(t) = 360d*time

Tip: Build the joint motion using the default expression in the Main toolbox container and then modify the expression using the Rotational Joint Motion Modify dialog box.

2 Add a motion to the shoulder-to-mount joint such that:

D(t) = STEP(time, 0, 0, 0.10, 30d)

Note: We will discuss the specifics of the STEP function in the next module, Lift Mechanism III, on page 153.

3 Add a motion to the boom-to-shoulder joint such that:

D(t) = STEP(time,0.8,0,1, 5)

4 Add a motion to the bucket-to-boom joint such that:

D(t) = 45d*(1-cos(360d*time))

Run a simulation

To run a simulation:

Run a simulation such that the mount achieves one full rotation.

Note: If any of the motions are opposite of what you expect from the Problem statement, on page 149, add a negative sign in front of the expression in the motion's modify dialog box.

Save your work

To save your work:

- Save the model such that the saved file contains only the model topology and not the results.

Optional tasks

If you did not already do so as explained in the Optional tasks, on page 143, for Lift Mechanism 1:

- Add tires to your model using the **Torus** tool.
- Constrain the tires to the base using revolute joints.

Module review

1 What are the markers that a joint refers to called?

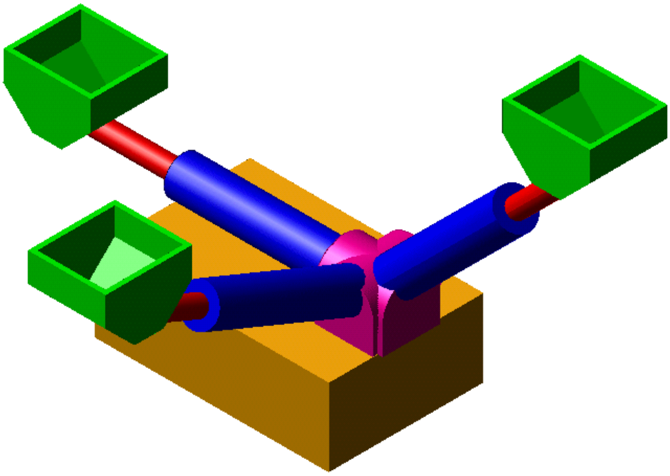
2 When motion is applied to a joint, what dictates its direction (positive versus negative)?

3 Are motions considered a constraint? Why?

4 Is it possible to determine the torque required to achieve a prescribed motion imposed on a revolute joint? How?

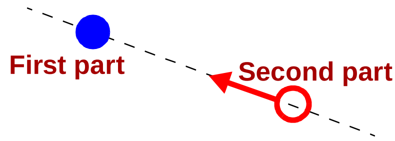
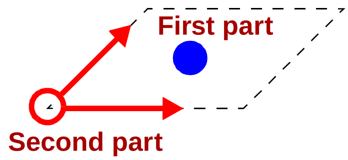
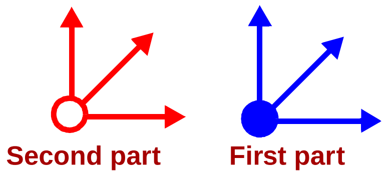
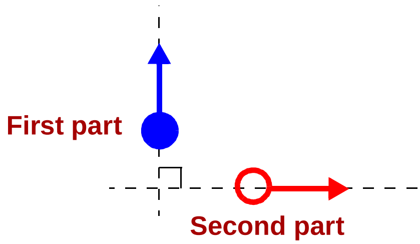
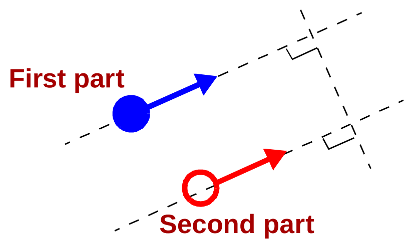
10 LIFT MECHANISM III

Constrain the bucket such that the base of the bucket always maintains its horizontal orientation (therefore, keeping the bucket-passenger safe) as shown next:



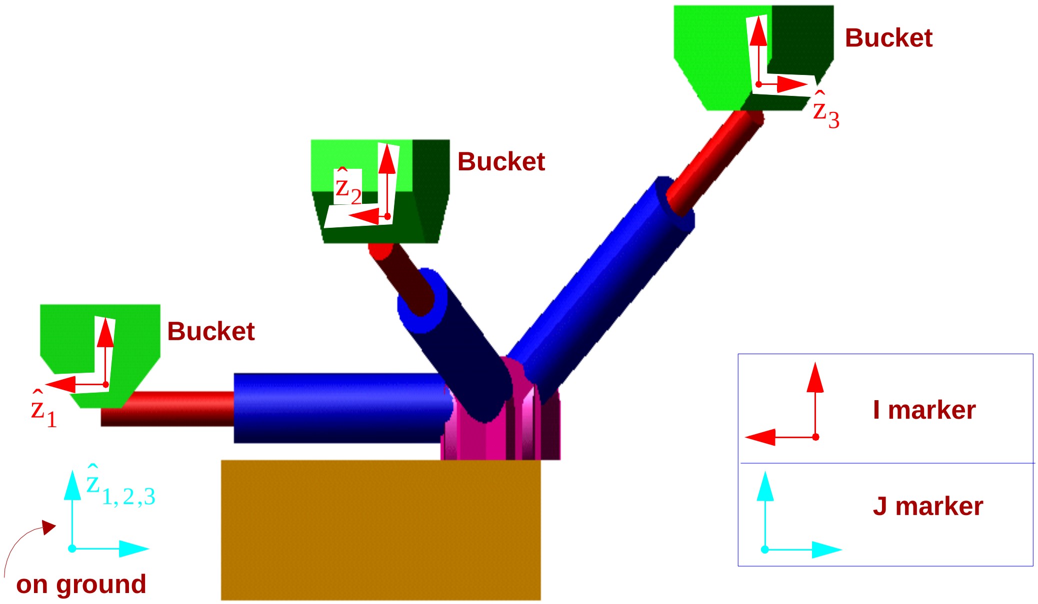
What's in this module:

Types of Joint Primitives

Description:	DOF removed:	Illustration:
Inline - One point can only move along a straight line	Two translational	
Inplane - One point can only move in a particular plane	One translational	
Orientation - One coordinate system cannot rotate with respect to another	Three rotational	
Perpendicular - One coordinate system can rotate about two axes	One rotational	
Parallel axis - One coordinate system can rotate about one axis	Two rotational	

See also: DOF removed by joint primitives on page 320

Example of I and J markers in a perpendicular joint primitive



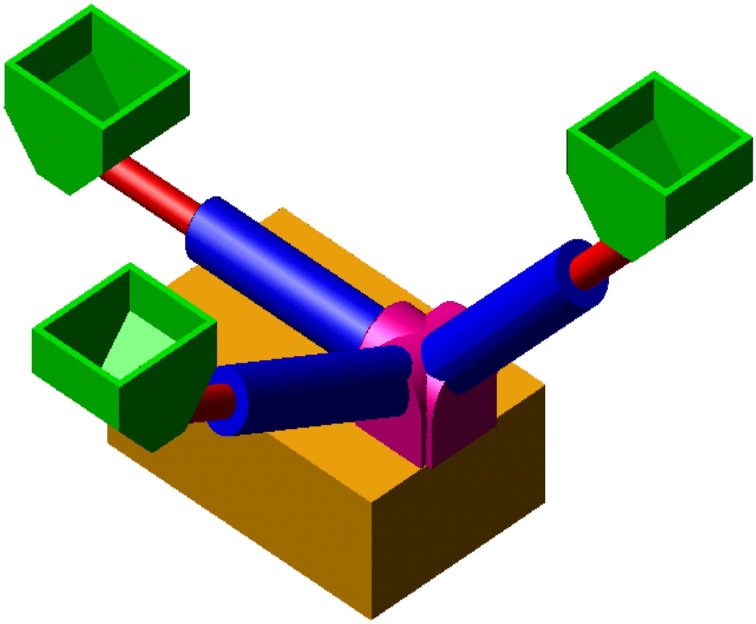
- I marker:
  - Parent part: Bucket
  - Its xy-plane is coplanar to the ground plane.
- J marker:
  - Parent part: ground
  - Its z-axis is perpendicular to the z-axis of the I marker.
- When constrained, the z-axes of the I and J markers are **always perpendicular** during simulation.
- Use the construction method 2 Bodies - 2 Locations.

Question: Would the lift mechanism behave any differently if the J marker's parent part was Base?

Workshop 10—Lift Mechanism III

Problem statement

Constrain the bucket such that the base of the bucket always maintains its horizontal orientation (thus keeping the bucket passenger safe) as shown next:



Model description

In this workshop, you use the model you saved in the Lift Mechanism II module.

Workshop 10—Lift Mechanism III...

Start the workshop

Note that the file for this exercise is not in the current working directory.

To start the workshop:

1. Start ADAMS/View from the directory ***exercise_dir*/mod_10_lift_mech_3**.
2. From the directory ***exercise_dir*/mod_09_lift_mech_2**, import the model that you created in the previous module.

   If you need a fresh copy of the model, import the command file lift_mech_2_completed.cmd from the directory *exercise_dir*/mod_09_lift_mech_2/completed.

Constrain the bucket

Now you will constrain the bucket to ground using the appropriate joint primitive.

To constrain the bucket:

1. Delete the motion on the bucket-to-boom joint.
2. Use the appropriate joint primitive to constrain the bucket to ground. You access the joint primitives from the Joints palette (from the **Build** menu, select **Joints**).

   Tip: Use the construction method **2 Bodies - 2 Locations** and refer to Example of I and J markers in a perpendicular joint primitive, on page 155 for assistance.

Verify the orientation of the I and J markers

Now you will verify the orientation of the I and J markers in the joint primitive.

To verify the orientation:

1. In the right corner of the Status bar, select the **Information** tool .
2. Note the names of the I and J marker and select **Close**.
3. Check that the z-axis of the marker on the *bucket* is pointing in the (positive or negative) global x direction.
4. Check that the z-axis of the marker on *ground* is pointing in the (positive or negative) global y direction.

Verify the model and then run a simulation

In this section, you will perform a simulation with icons on.

To verify the model and run a simulation:

1. Verify the model. It should have zero degrees of freedom.
2. From the **Settings** menu, point to **Solver**, and then select **Execution Display**.
3. In the Simulation Settings dialog box, select **Icons visible**.
4. Simulate the model.

Save your work

To save your work:

1. Save the model such that the saved file contains only the model topology and not the results.
2. Exit ADAMS/View.

Optional tasks

To complete extra tasks from previous module:

If you did not already do so as explained in the Optional tasks, on page 143, for Lift Mechanism 1:

- Add tires to your model using the **Torus** tool.
- Constrain the tires to the base using revolute joints.

To make the bucket transparent:

1. From the **View** menu, select **Render Mode**, and then select **Shaded**.
2. Right-click the bucket, point to **Part: bucket**, and then select **Appearance**.
3. To set the transparency of the part, use the slider bar:
   0%: fully visible, 100%: completely invisible.

To check graphical topology of the model:

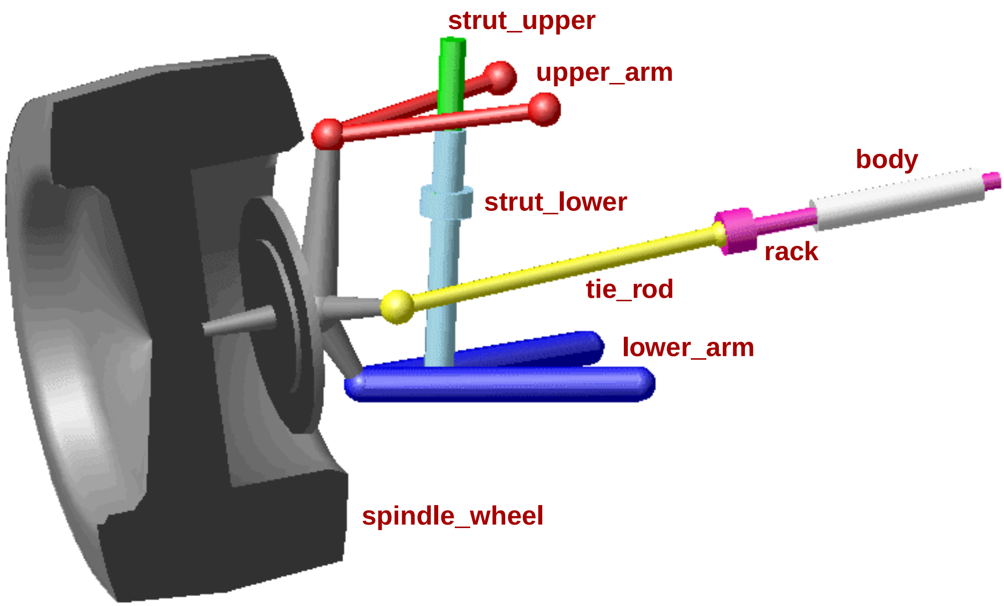
1. From the **Tools** menu, select **Database Navigator**.
2. Set the menu at the top of the Database Navigator to **Graphical Topology**.
3. Browse to the bucket and notice how many constraints act on the bucket.

Module review

1. When you use the construction method of 2 Bodies - 2 Locations, how does the order in which you select parts affect the order in which you select the locations and orientations?

11 SUSPENSION SYSTEM I

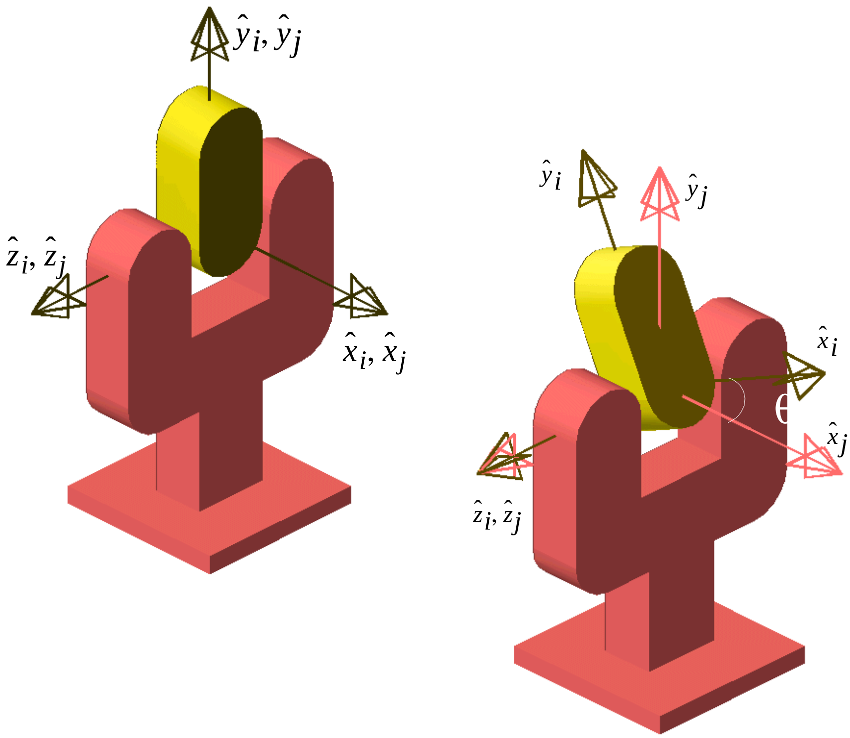
Set up the suspension such that it moves 80 mm in jounce and rebound.



What's in this module:

Point motions

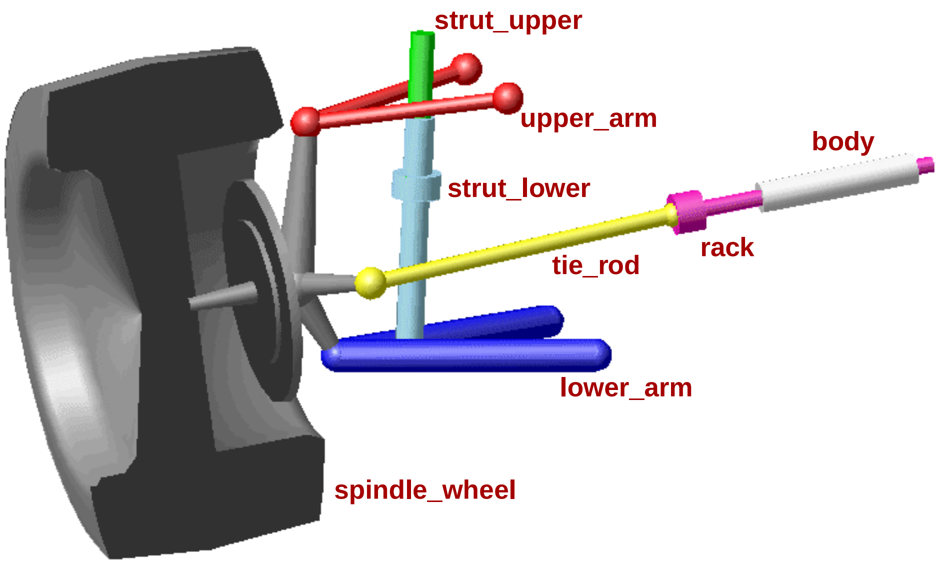
- There are two types:
  - Single-point motion (removes 1 DOF)
  - General-point motion (removes 1 to 6 DOF)
- You define:
  - I and J markers to which motion is applied (via two bodies, location and orientation).
  - Constraint nature of the motion (between 1 and 6 DOF).
  - Functions for magnitudes of motion.



Workshop 11—Suspension System I

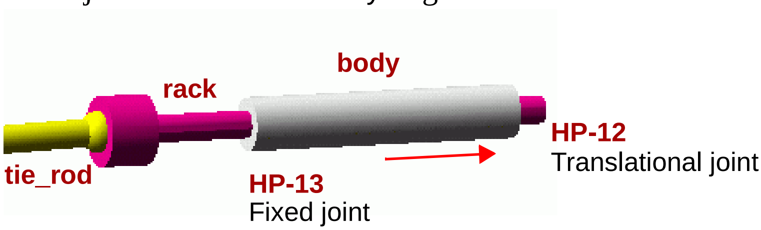
Problem statement

Inspect the toe angle that the wheel exhibits throughout its vertical travel of 80 mm in jounce and rebound.

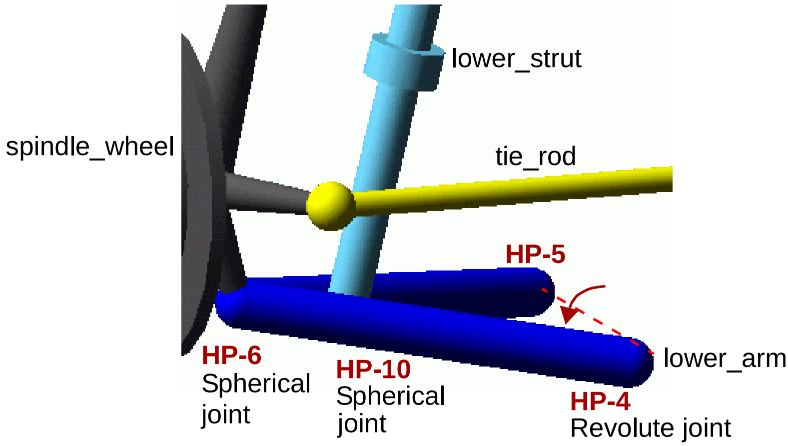


Model description

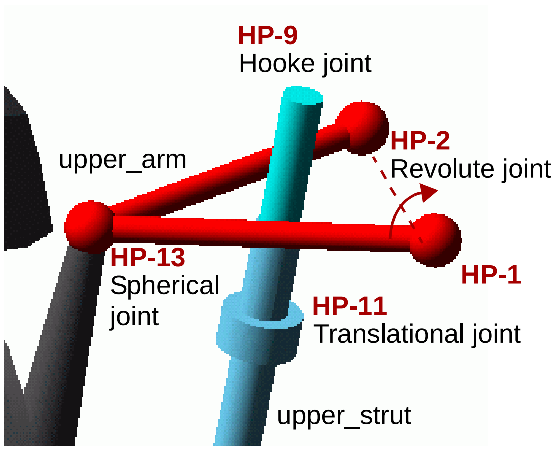
- The given model is a geometric representation of a short-long arm (SLA) suspension subsystem.
- The rack and body are constrained as shown in the following figure:
  - A translational joint connects the rack to the body.
  - A fixed joint connects the body to ground.



- The lower_arm and lower_strut are constrained as shown next:
  - A spherical joint connects the lower_strut to the lower_control_arm.
  - A revolute joint connects the lower_arm to the body.



- The upper_arm and upper_strut are constrained as shown next:
  - A revolute joint connects the upper_arm to the body.
  - A hooke joint connects the upper_strut to the body.

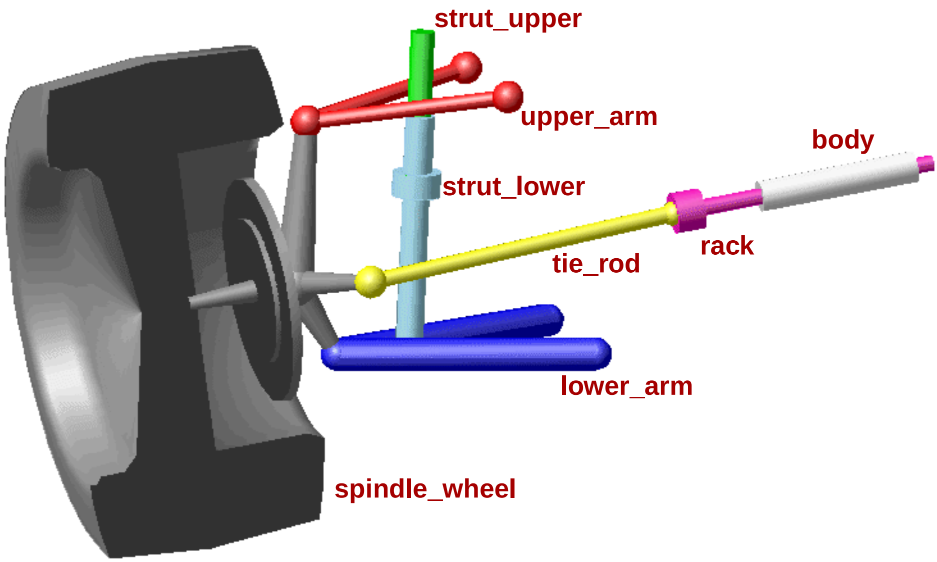


Start the workshop

To start the workshop:

1. Start ADAMS/View from the directory ***exercise_dir*/mod_11_suspension_1**.
2. Import the model command file **suspension_parts_start.cmd**.

   This file contains commands to build a model named suspension and the following parts with geometric representation:



Inspect the model

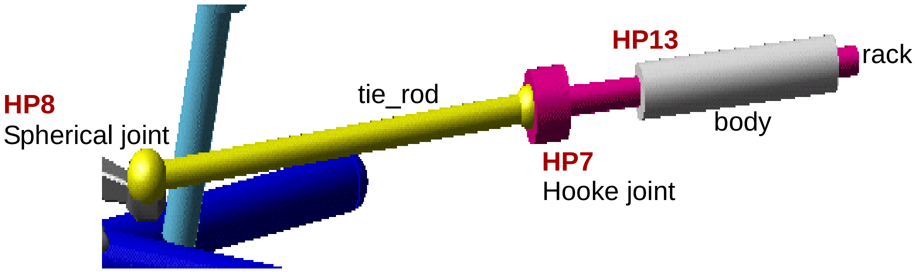
In this section, you'll investigate the model to note its movement and topology, especially that of the part tie_rod.

To inspect the model:

1. Simulate the model, noting the movement of tie_rod.
2. From the **Tools** menu, select **Database Navigator**.
3. In the Database Navigator, select **Graphical Topology**.
4. Double-click **suspension**, and then select **tie_rod**.

Constrain the suspension subsystem model

Constrain the tie rod as shown next:



When constraining the **tie_rod**, use the ADAMS/View hard points provided with the model.

To constrain the model:

1. Create a spherical joint.
2. Create a hooke joint, using the direction vectors of **HP7** to **HP8**, and **HP7** to **HP13**.
3. Simulate the model.

Apply motions

To apply motions:

1 At the marker **.spindle_wheel.center**, apply a point motion, in the $\hat{y}_G$ direction, to the **spindle_wheel** center using the following function:

Displacement(time) = 80*sin(360d*time).

2 Modify the translational joint between the **rack** and the **body** to be a **fixed joint**, so that the rack is unable to translate during a simulation.

Verify and simulate the model

Now, to see the model's full range of motion, simulate it.

To verify and simulate the model:

1 Verify the model.

2 Simulate the model for one second.

Save your work

1 Save your model as **suspension_parts.cmd**.

2 Exit ADAMS/View.

Optional tasks

To modify hardpoint locations:

1 From the **Tools** menu, select **Table Editor**.

2 From the options along the bottom of the Table Editor, select **Points**.

3 Change the **Loc Y** value of **HP3** from **351.05** to **400**.

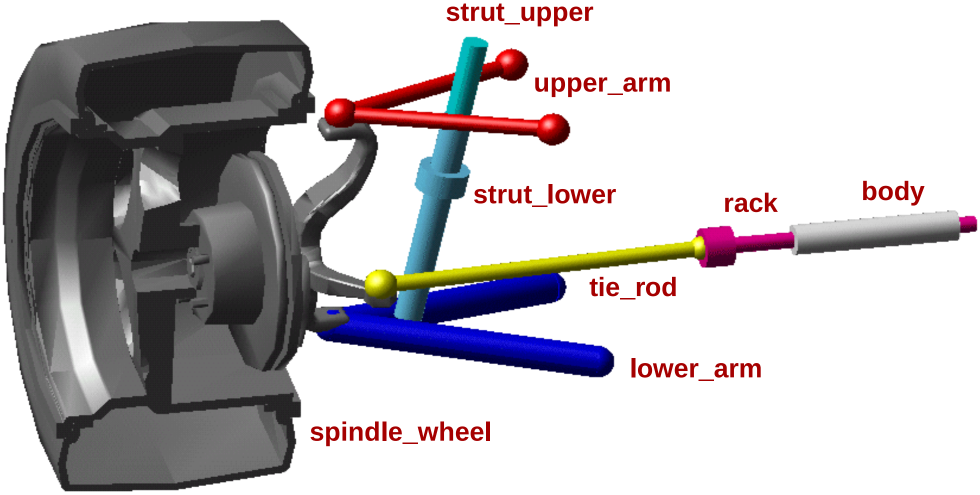
As you make this change, note how the upper arm's connection to the spindle changes.

Module review

1. What is the difference between a point motion and a joint motion?

12 SUSPENSION SYSTEM II

Inspect the toe angle that the wheel exhibits throughout its vertical travel of 80 mm in jounce and rebound.



What's in this module:

Point-to-point measures

- Measure kinematic characteristics of one point relative to another point, such as the relative velocity or acceleration.
- To define them, you specify:
    - Characteristic (displacement, velocity, or acceleration)
    - To-point marker location (I marker)
    - From-point marker location (J marker, default is global origin)
    - Represent coordinates in marker coordinate system (R marker, default is GCS)
    - Component to return (x, y, z, or magnitude)
- ADAMS/View uses displacement, velocity, or acceleration functions.

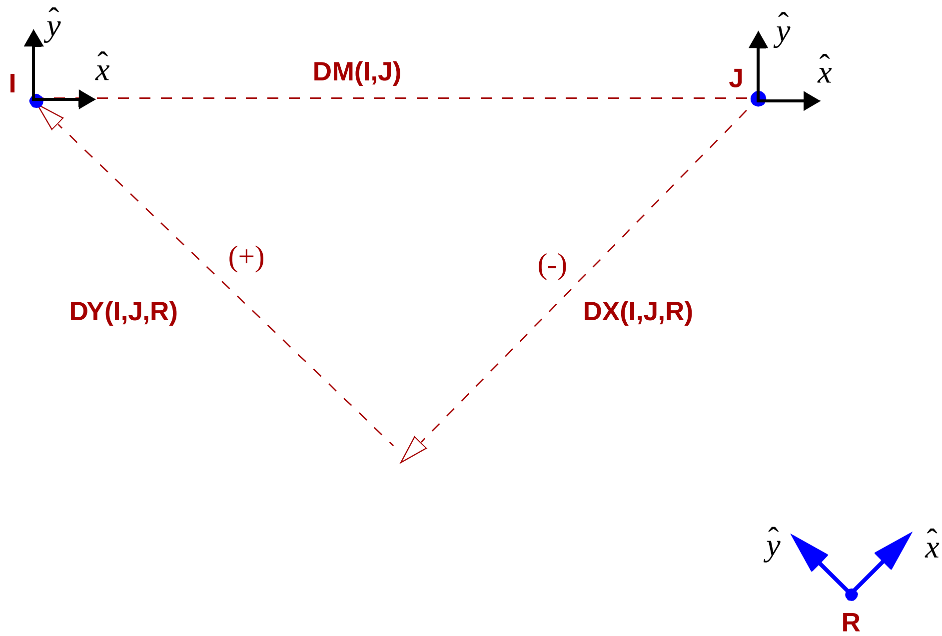
Function measures

- Let you evaluate arbitrary, user-defined expressions of interest during solution run-time, such as:
    - Flow rate
    - Aerodynamic pressure
    - Stress
- You can create them in the Function Builder.
- Unlike other measures, function measures let you specify plotting attributes.

Displacement functions

- For translational displacement, return scalar portions of vector components (measurements are taken to I from J, resolved in R's CS), as shown below.
- For rotational displacement, return angles associated with a particular rotation sequence.

Example



Importing CAD-Based Geometry

Conceptual Design Method

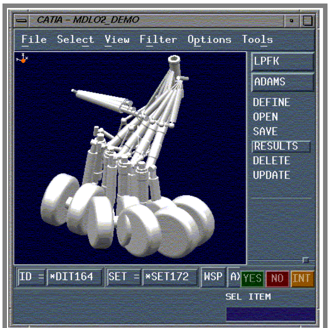
CAD Assembly

Design Validation Method

Dynamic Motion! → CAD Assembly ← **Dynamic Motion!**

Import .res (Conceptual Design Method)

Import .res (Design Validation Method)

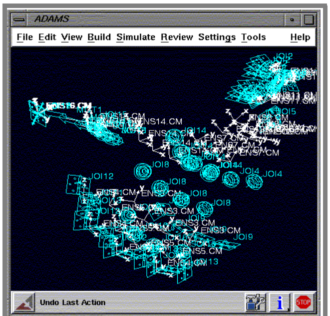


- Rigid Bodies
- Mass Properties
- Detailed Geometry
- Joints
- Springs
- Applied Forces
- Kinematic Motion

Export

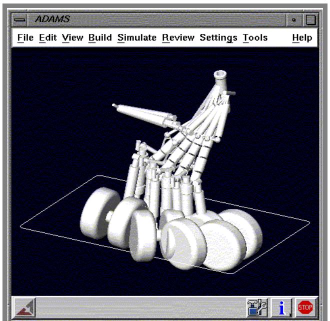
.adm ← → .cmd + .adm + .shl, .slp

ADAMS/Solver or ADAMS/View without Geometry



- Rigid Bodies
- Mass Properties
- Joints
- Springs
- Applied Forces
- Advanced Modeling
- Dynamic Motion

ADAMS/View with Geometry

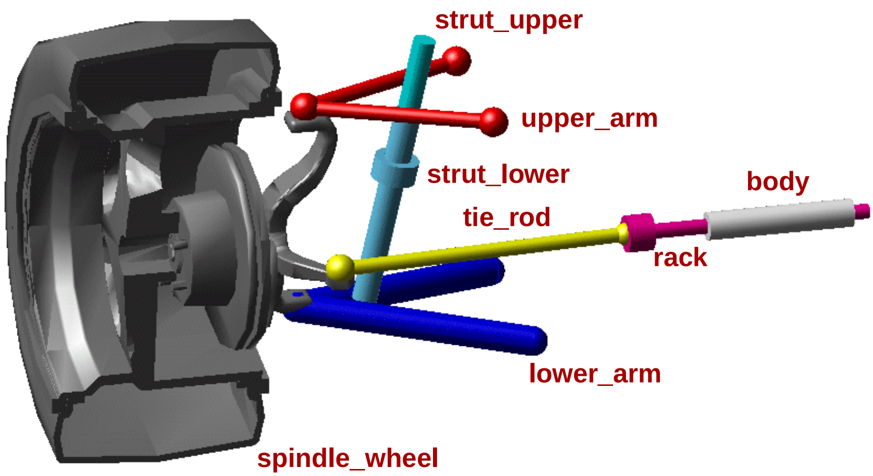


- Rigid Bodies
- Mass Properties
- Detailed Geometry
- Joints
- Springs
- Applied Forces
- Advanced Modeling
- Dynamic Motion

Workshop 12—Suspension System II

Problem statement

Inspect the toe angle that the wheel exhibits throughout its vertical travel of 80 mm in jounce and rebound.



Model description

In this workshop, you use the model you built in Workshop 11—Suspension System I, on page 163.

Start the workshop

Note that the file for this workshop is not in the current working directory.

To start the workshop:

1. Start ADAMS/View from the directory ***exercise_dir*/mod_12_suspension_2**.
2. From the directory ***exercise_dir*/mod_11_suspension_1**, import the model that you created in the previous workshop.

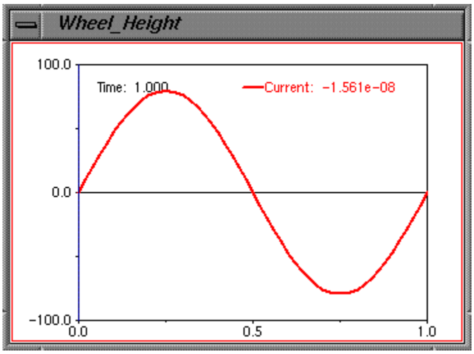
   If you need a fresh copy of the model, import the command file suspension_1_completed.cmd from the directory *exercise_dir/*mod_11_suspension_1/completed.
3. Simulate the model to verify the motion.

Create measures

To create measures:

1. Create a point-to-point measure, named **.suspension.Wheel_Height**, for the relative wheel displacement in the $\hat{y}_G$ direction using the markers **spindle_wheel.center** and **ground.WH_ref.**

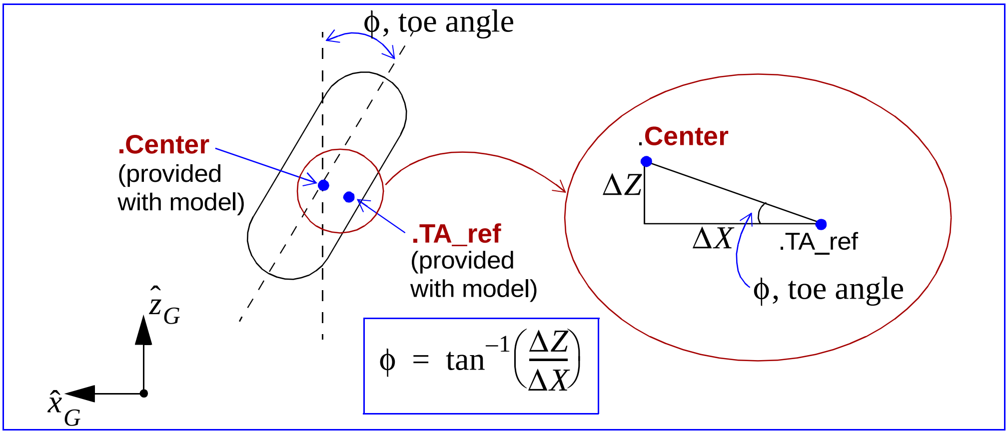
   Tip: From the **Build** menu, point to **Measure**, point to **Point-to-Point**, and then select **New**.

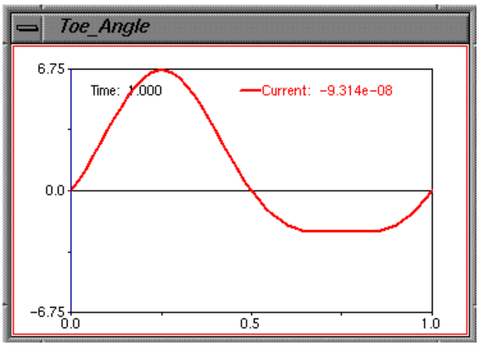


2 Using an ADAMS/Solver function measure, create a toe angle measure using the markers **Spindle_Wheel.Center** and **Spindle_Wheel.TA_ref**

Tip: Use the ATAN function.

Note: You must run a simulation after creating the function to view its plot.

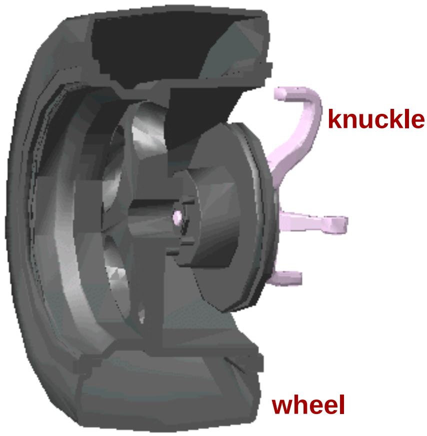




3 In ADAMS/PostProcessor, plot toe angle versus wheel height.

Import CAD-based geometry

Now, you'll import more realistic, CAD-based spindle/wheel geometry, as shown next.



The two geometry files that make up the spindle/wheel are:

- wheel.slp
- knuckle.slp

They are render files, which have an extension of .slp. They were created in Pro/ENGINEER. By default, when you import the files, ADAMS/View names the geometry based on the Pro/ENGINEER assembly from which they came and not based on their file names. In this case, the CAD geometry came from a model named suspensn. Therefore, ADAMS/View names the geometry suspensn and suspensn_2.

When you export your model, ADAMS/View exports one .cmd file (suspension.cmd) and one .shl file for *each* CAD geometry (suspensn.shl and suspensn_2.shl).

To import the geometry:

1 Import the geometry files located in *exercise_dir*/mod_12_suspension_2/suspension_cad.

Tip: From the **File** menu, select **Import**. Then, set **File Type** to **Render** and select to attach the geometry to the part **spindle_wheel**.

2 Turn off the appearance of ADAMS/View spindle geometry so that only the CAD geometry is visible.

Tip: From the **Edit** menu, select **Appearance**.

Save your work

To save your work:

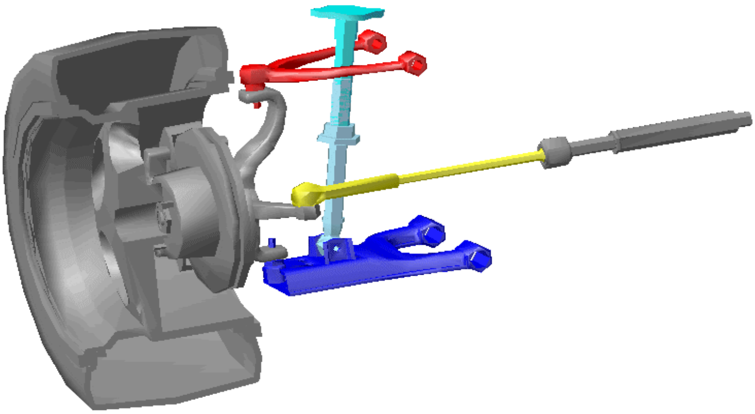
1. Save your model as **suspension_parts.cmd**.
2. Exit ADAMS/View.

Optional tasks

To turn off appearance of ADAMS/View geometry:

1. From the directory ***exercise_dir*/mod_12_suspension/suspension_cad/more_susp_cad**, import the rest of the CAD-based suspension component geometry.

   These geometry files are called render files, which have an extension of .shl. There is one file for each ADAMS part.



2. Turn off the appearance of ADAMS/View geometry so that only the CAD geometry is visible.

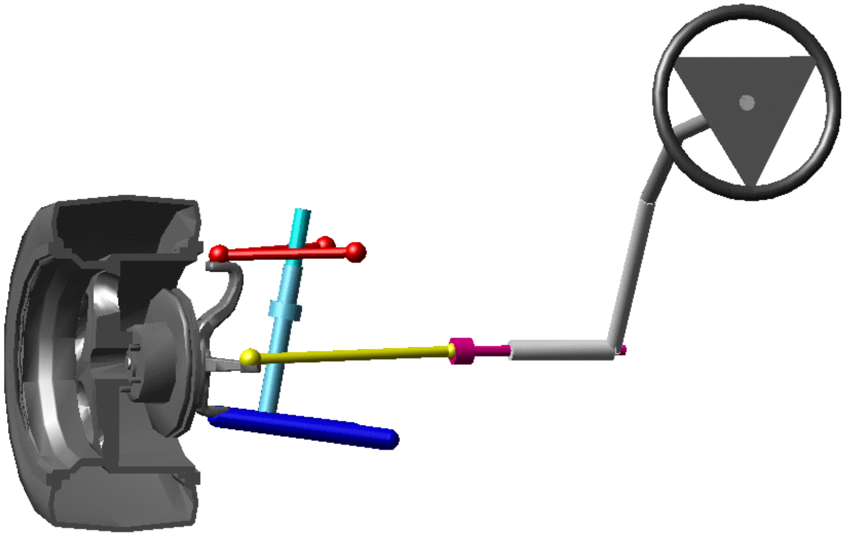
Module review

1 Is there any difference between a point-to-point measure and a function measure using a displacement function (for example, DX(I, J, R))?

2 Where does a CAD file fall in the model hierarchy? What is the CAD file a child of?

13 SUSPENSION-STEERING SYSTEM

Assemble a suspension-steering system and inspect the toe angle that the wheel exhibits at steering wheel angles of 60^o, 0^o, and -60^o.



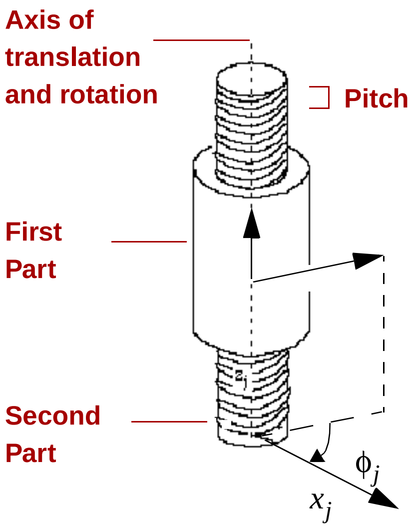
What's in this module

Add-on (complex) constraints

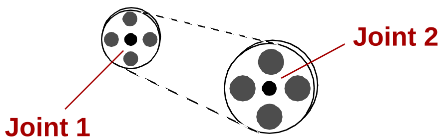
- Set up relationships between existing constraints in a system.
- Connect parts directly and indirectly.

Types of add-on constraints

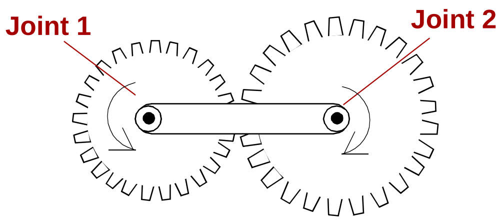
Screw Joints



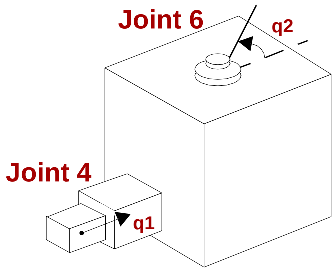
Couplers



Gears



Definition of couplers



- Couplers connect multiple parts indirectly by coupling 2 joints.
- Couplers remove 1 DOF, based on the following equation:

  S1q1 + S2q2 = 0

  where:

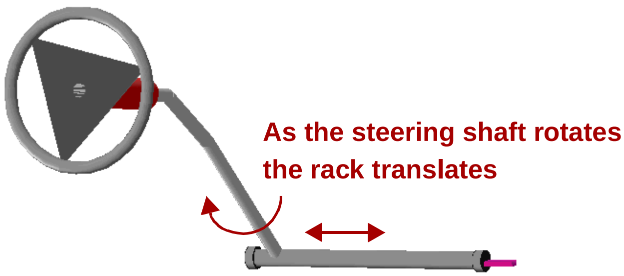
  - S_1, S_2 - scalar multipliers
  - q_1 - allowable DOF in the driver joint
  - q_2 - allowable DOF in coupled joints

Modeling of couplers requires:

- Two joints
- Two scalar multipliers

Note: In ADAMS/View, S_1= -1.
It is also possible to use a three-joint coupler.

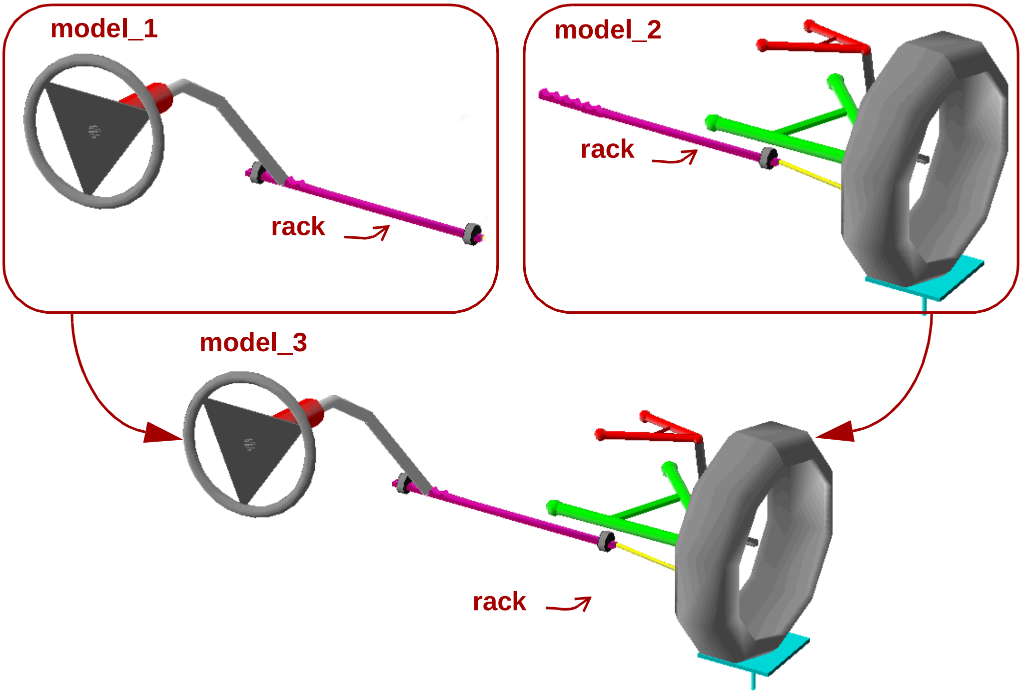
Example of a coupler



Assembling Subsystem Models

When you assemble models

- Any number of models can be assembled.
- Assembling models will create a new model.
- All assembled models (model1, model2) will continue to exist in the database along with the new model (model3).



Parts in assembled models

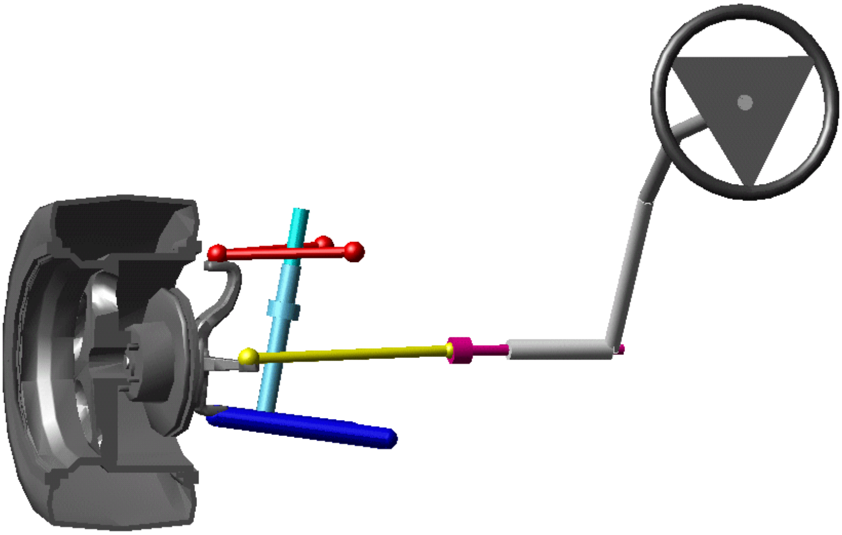
- They maintain their global location and orientation, unless otherwise specified.
- If parts have the same name in different merged models, ADAMS/View will either:
    - Merge them into one part.
    - Rename the parts.

See also: Model Hierarchy on page 22

Workshop 13—Suspension-Steering System

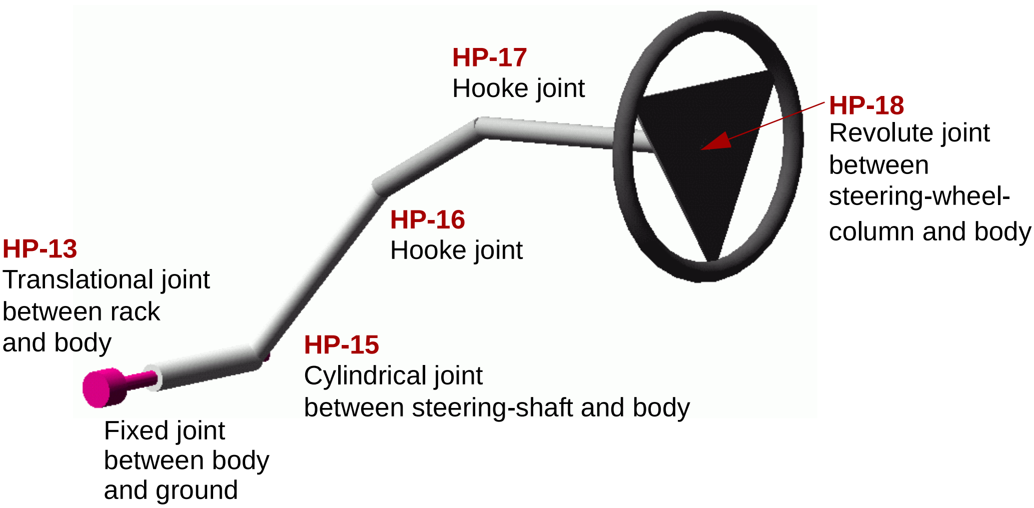
Problem statement

Assemble a suspension-steering system and inspect the toe angle that the wheel exhibits at steering wheel angles of 60°, 0°, and -60°.



Model description

- You will use the following two models in this workshop:
  - A geometric representation of a short-long arm (SLA) suspension subsystem.
  - A geometric representation of a rack-and-pinion steering system.
- The rack-and-pinion steering model is constrained as shown next:



Start the workshop

Note that the file for this workshop is not in the current working directory.

To start the workshop:

1. Start ADAMS/View from the directory ***exercise_dir*/mod_13__susp_steer**.
2. From the directory ***exercise_dir*/mod_12_suspension_2**, import the model that you created in the previous module.

   If you need a fresh copy of the model, change your working directory to *exercise_dir*/mod_12_suspension_2/completed, then import the command file suspension_completed.cmd.

Change working directory

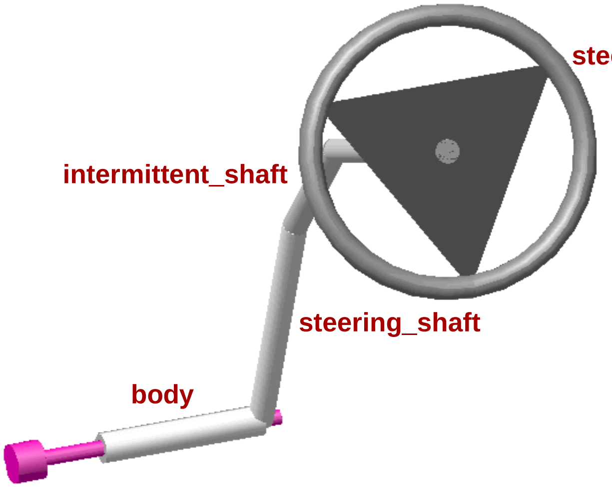
Change the directory to *exercise_dir*/mod_13_susp_steer. Running ADAMS/View in this directory ensures that all saved data gets stored there.

To change the working directory:

1. From the **File** menu, select **Select a Directory**.
2. Change to ***exercise_dir*/mod_13_susp_steer**.

Import the steering model

Import the steering model, shown below. It is in the command file steering_parts_start.cmd. The file contains a model named rack_and_pinion_steering.



To import the model:

- Import the ADAMS model command file **steering_parts_start.cmd**.

Constrain the steering model

Now you'll constrain the steering model. Each time you add a modeling element, you'll simulate the model to verify its movement.

To constrain the steering model:

1. Using the following function, apply a joint motion to the revolute joint on the **steering_wheel_column**:

   Displacement(time) = 45d*sin(360d*time)
2. Simulate the model.
3. Couple the rotation of the **steering_shaft** with the translation of the **rack**, so that for every **7**^{o} of rotation the rack travels **1 mm**.
4. To verify that the rack travels as expected, simulate the model.

Assemble the suspension and steering models

To assemble the models:

1. Assemble the **rack_and_pinion_steering** model with the **suspension** model by doing the following:

   - From the **Tools** menu, select **Command Navigator**.
   - In the Command Navigator, double-click **model**, and then double-click **assemble**.

   The assembled model might be overconstrained.

2. To find duplicate joints, from the **Tools** menu, select **Database Navigator**. Then, do the following:

   - Set the top menu in the Database Navigator to **Graphical Topology**.
   - Double-click the new model name, and then select part names.
   - Delete duplicate joints.

3. To redisplay measures for the model, from the **Build** menu, point to **Measure**, and then select **Display**.

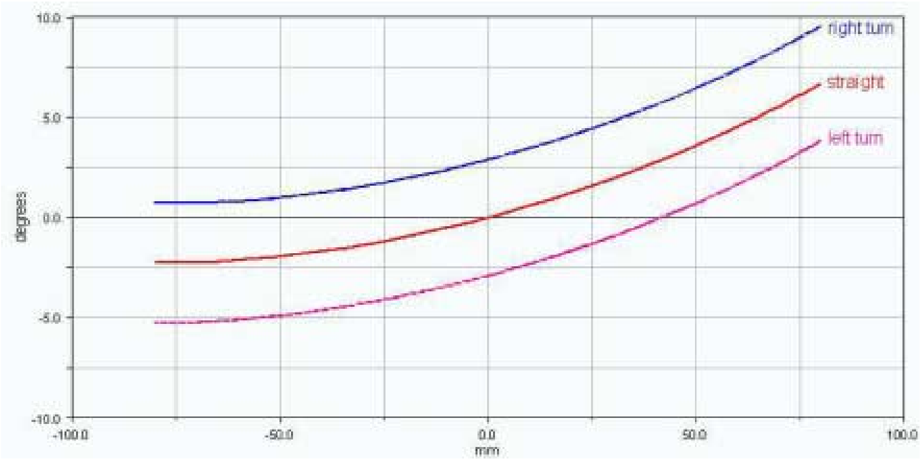
Run and compare a series simulations

In this section, you'll run three simulations, each with different wheel angles. You'll then compare the results of the simulations.

To run and compare a series of simulations:

1. Run a simulation with a 45^{o} wheel angle.
2. Save the simulation results as **right_turn**.
3. Run a simulation with a 0^{o} wheel angle.
4. Save the simulation results as **straight**.
5. Run a simulation with a -45^{o} wheel angle.
6. Save the simulation results as **left_turn**.
7. Start ADAMS/PostProcessor.

8 Plot **toe_angle** versus **wheel height** for all three simulations, on the same plot.



Save your work

To save your work:

1 Save your model.

2 Exit ADAMS/View.

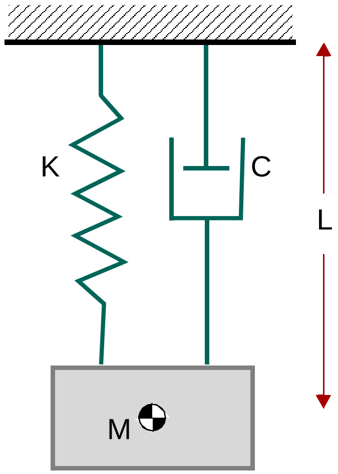
Module review

1. What information do you need to provide ADAMS/View to create a coupler?

2. What is the default name that ADAMS/View assigns to simulation results?

14 SPRING-DAMPER

Create and investigate the linear spring-damper system shown in the following figure, using different types of simulations in ADAMS.



M: 187.224 Kg

K: 5.0 N/mm

C: 0.05 N-sec/mm

L_0: 400 mm

F_0: 0

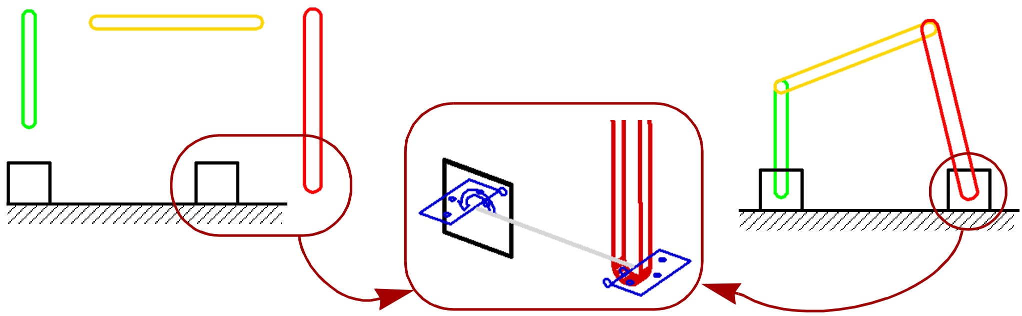
What's in this module:

Definition of assemble simulation

- Attempts to resolve any conflicts in the initial conditions specified for the entities in the model (for example, broken joints).
- Is also known as an **initial conditions simulation**.

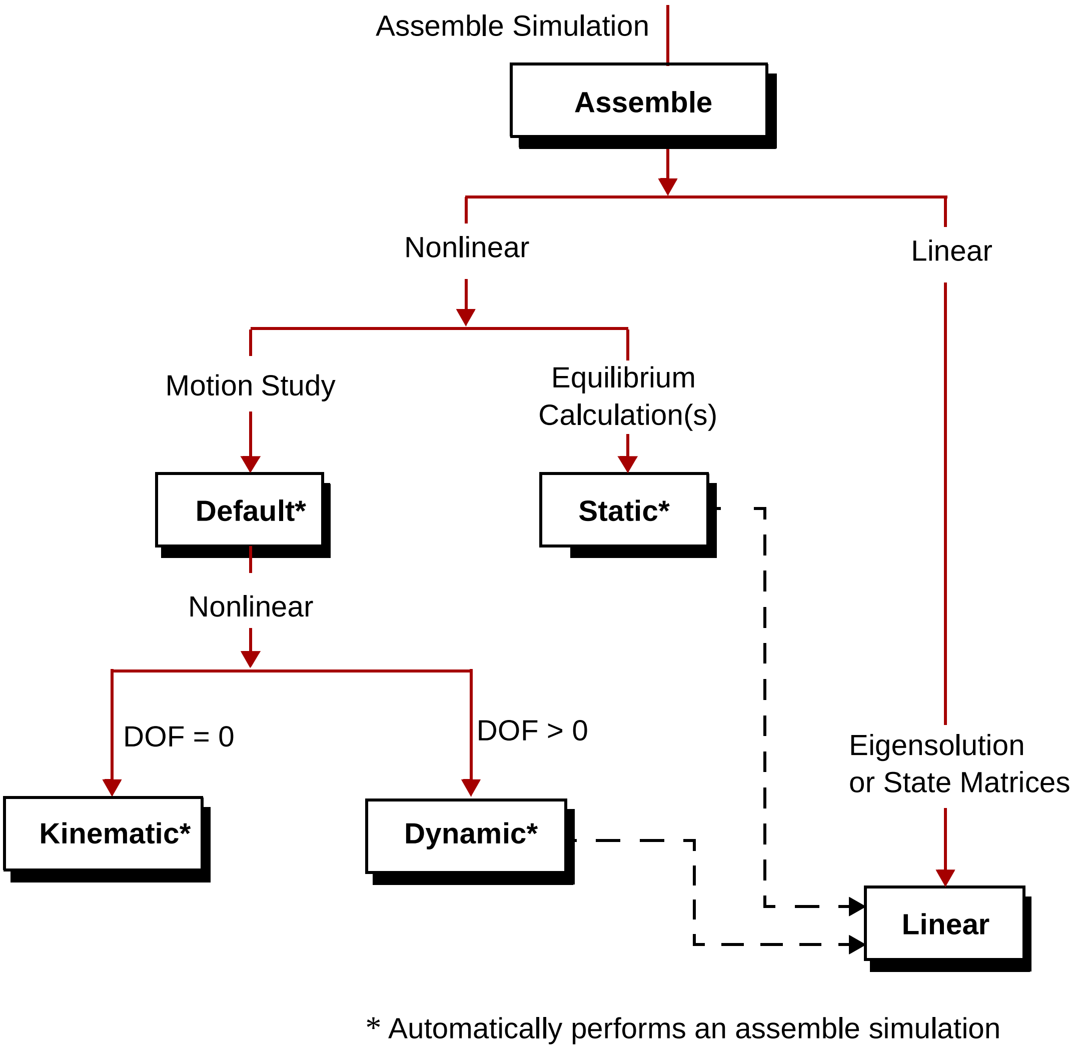
Initial location and orientation of parts

- You specify the initial position and orientation for a part when you create it.
- For a part to be held fixed during the assemble simulation, you can specify up to three positions ($\hat{x}_G$, $\hat{y}_G$, $\hat{z}_G$) and up to three orientations (psi, theta, phi).



Note: Use initial positions sparingly. If you fix the initial positions of too many parts, the assemble simulation can fail.

Simulation Hierarchy

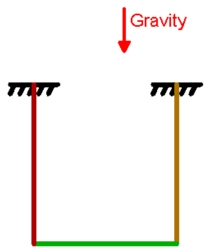


Note: Often a linear simulation is used after a static equilibrium or dynamic simulation.

Types of Simulations

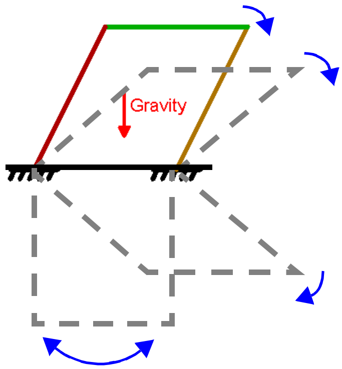
Static

- System DOF > 0.
- All system velocities and accelerations are set to zero.
- Can fail if the static solution is a long way from the initial condition.



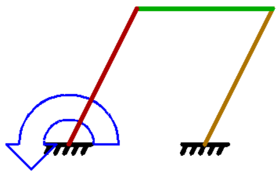
Dynamic

- System DOF > 0.
- Driven by a set of external forces and excitations.
- Nonlinear differential and algebraic equations (DAEs) are solved.



Kinematic

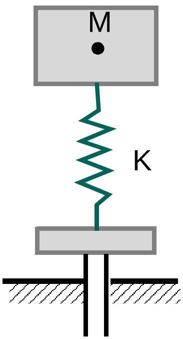
- System DOF = 0.
- Driven by constraints (motions).
- Only constraint (algebraic) equations are being solved.
- Calculate (measure) reaction forces in constraints.



Linear

- ADADMS can linearize the system of nonlinear equations of motion about a particular operating point.
- From the linear set of equations, you can ask for an eigen-simulation to obtain eigenvalues and eigenvectors for the linearized system to:
  - Visualize the natural frequencies and mode shapes of your system.
  - Compare with test data or results data from FEA.

Example of linear simulation



- Must linearize about an operating point (often the equilibrium).
- Extraction of natural frequency.
- Natural frequency = $\sqrt{\frac{K}{M}}$.

Definition of forces

- **Try** to make parts move in certain ways.
- Do not perfectly connect parts together the way constraints do.
- Do not absolutely prescribe movement the way motion drivers do.
- Neither add nor remove DOF from a system.

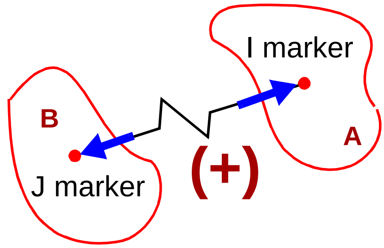
Characteristics of forces

The characteristic:	Defines:
Bodies	Which parts are affected
Points of application	Where the parts are affected
Vector components	How many vector components there are
Orientation	How the force is oriented
Magnitude	If the force is pre-defined or user-defined

Spring-Dampers in ADAMS

Definition of spring-dampers

- They are pre-defined forces.
- They represent compliance:
  - Between 2 bodies.
  - Acting over a distance.
  - Along or about one particular direction.



Characteristics of spring-dampers

The characteristic:	Defines:
Bodies	Two (A, B)
Points of application	Two (I and J marker)
Vector components	One
Orientation (only for translational)	Acts along the line of sight between the I and J markers ■ **Positive** force **repels** the two parts ■ **Negative** force **attracts** the two parts
Magnitude	Pre-defined equation based on either: ■ Stiffness and damping coefficients (linear) ■ Splines based on test data (nonlinear)

See also: Characteristics of a spring-damper on page 289

Magnitude based on stiffness and damping coefficients

- Linear spring-damping relationship can be written as:

  $Force_{SPDP} = -k(q - q_0) - c q + F_0$

  where:

  q - Distance between the two locations that define the spring-damper

  q - Relative speed of the locations along the line-of-sight between them

  k - Spring stiffness coefficient (always > 0)

  c - Viscous damping coefficient (always > 0)

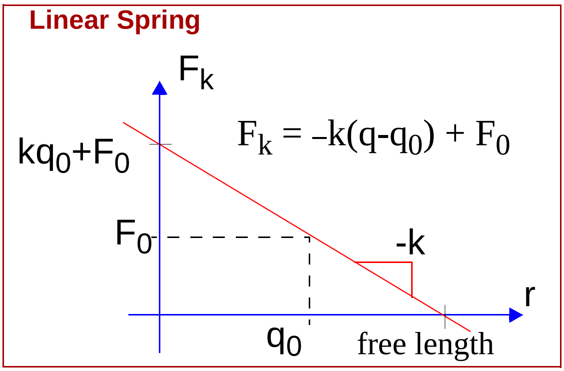
  F_0 - Reference force of the spring (preload)

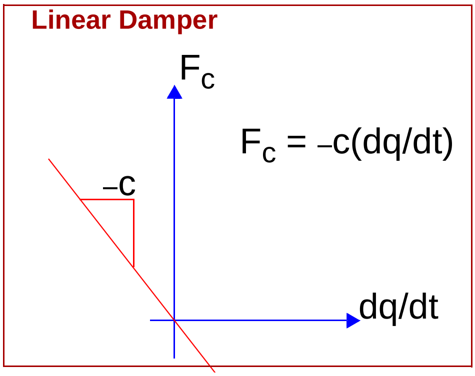
  q_0 - Reference length (at preload, always > 0)

  t - Time

- In ADAMS, the user-defined equation is:

  - k*(DM(I, J) - q_0) - c*VR(I, J) + F_0



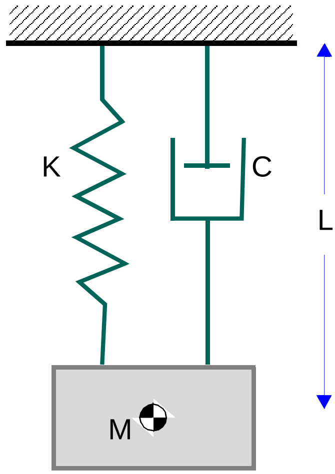


- Spring-damper forces become ill-defined if endpoints become coincident because of undefined direction.

Workshop 14—Spring-Damper

Problem statement

Create and investigate the linear spring-damper system shown next, using different types of simulations in ADAMS.



M: 187.224 Kg

K: 5.0 N/mm

C: 0.05 N-sec/mm

L_0: 400 mm

F_0: 0

Start the workshop

To start the workshop:

1 Start ADAMS/View from the directory ***exercise_dir*/mod_14_spring_damper**.

2 Create a new model called **spring_mass**.

Build and constrain the model

To build and constrain the model:

1 Build the block with the given mass.

2 Constrain the block to move only in the $\hat{y}_G$ direction.

Tip: Add a translational joint.

3 To verify expected behavior, simulate the model.

Add the pre-defined spring-damper

To add a pre-defined spring-damper:

1. Create the spring-damper between the cm marker of the block and a point on ground 400 mm above it.

   Make sure that the spring-damper is aligned along the $\hat{y}_G$ direction.

2. To ensure that the spring-damper length at preload is 400 mm with a preload of 0, from the **Tools** menu, select **Measure Distance**.

Find the force in spring-damper at static equilibrium

To find the force at static equilibrium:

1. Run a static equilibrium simulation.
2. Note the value of the force graphic.
3. Zoom out.

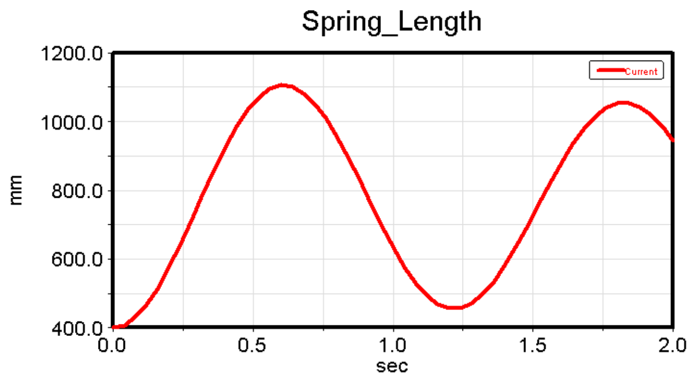
   The block's mass is 187.22 Kg. Therefore, to balance the force of gravity, the spring-damper must generate:

   187.224kg*9806.65mm/s2(=1836.04 N)

Run a simulation and create a measure

To run a simulation and create a measure:

1. To see oscillation, run a **2**-second, **50**-step dynamic simulation.
2. Create a point-to-point measure, named **Spring_Length**, to measure the spring's length. Measure the upper spring-damper attachment point, with respect to the block's cm marker.



The measured value at t=0 should be 400 mm.

Find the natural frequency

To find the natural frequency:

1. Run a linear simulation and linearize about the equilibrium position.
2. Note the natural frequency, and compare this value with that given in Closed-form solution, on page 201.

Save your work

To save your work:

1. Save the model.
2. Exit ADAMS.

Optional tasks

Add a DOF to the model:

1. Modify the translational joint to be a cylindrical joint.
2. Linearize about the static equilibrium position.

   Do the resulting modes make sense?
3. Add a torsional spring-damper that resists the rotation of the cylindrical joint.
4. Linearize about the static equilibrium position.

   Are the results different from those above (no torsional spring-damper)?
5. Do not save your work.

ADAMS results

$$\omega_n = 0.8222\ \text{Hz}$$

$$\omega_n = (0.8222\ \text{Hz})(2 \cdot \pi\ \text{rad}) = 5.168\ \text{rad/sec}$$

Closed-form solution

Checking the natural frequency of the system:

At equilibrium:

$$m\ddot{x} + c\dot{x} + kx = 0$$

$$\ddot{x} + \frac{c}{m}\dot{x} + \frac{k}{m}x = 0$$

Laplace Transform is:

$$s^2 + \frac{c}{m}s + \frac{k}{m} = 0 \Leftrightarrow s^2 + 2\sqrt{\zeta\omega_n}s + \omega_n^{\ 2} = 0$$

Therefore:

$$\omega_n^2 = \frac{k}{m}$$

$$\omega_n = \sqrt{\frac{k}{m}}$$

k = 5 N/mm = 5000 N/m

m = 187.224 kg

$$\omega_n = \sqrt{\frac{5000}{187.224}}\ \text{rad/sec}$$

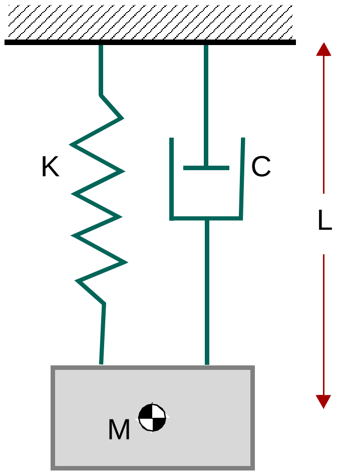
$$\omega_n = 5.168\ \text{rad/sec}$$

Module review

1 At design configuration, do the $\hat{z}$ directions of markers referenced in a revolute joint have to be aligned? Does this information get reported when verifying a model?

15 NONLINEAR SPRING

Investigate the differences between a linear spring and a nonlinear spring using a spline function.



M: 187.224 Kg

K: 5.0 N/mm

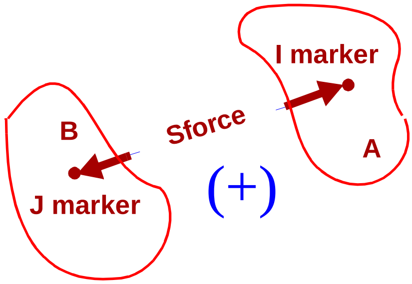
C: 0.05 N-sec/mm

L_0: 400mm

F_0: 0

What's in this module:

Characteristics of action-reaction single-component forces (Sforces)



The characteristic:	Defines:
Bodies	Two (A, B)
Points of application	Two (I and J markers)
Vector components	One
Orientation	Acts along the line of sight (between the I and J markers) ◆ **Positive** force **repels** the two parts ◆ **Negative** force **attracts** the two parts
Magnitude	User-defined

See also: Characteristics of an action-reaction S-force on page 289

Note: ADAMS applies action and reaction forces to the I and J markers that it automatically creates.

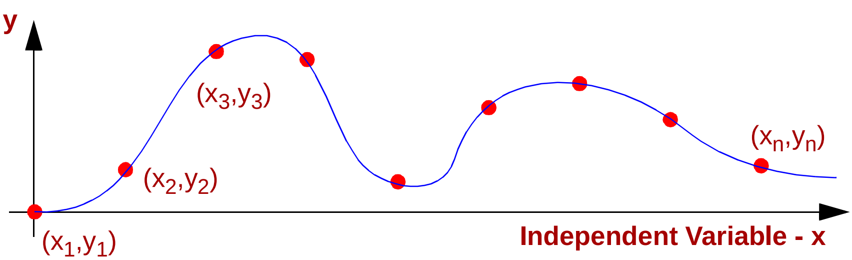
Spline functions

Test data that can be incorporated into a simulation includes

- Empirical data from suppliers or standard tables for:
  - Nonlinear compliances (force versus velocity).
  - Curves for torque versus motor speed (torque versus angular velocity).
- Data taken from physical prototype simulations for:
  - Accelerometer data (acceleration versus time).
  - Tire lateral force as a function of normal force and slip angle.

To incorporate data into a simulation

- **First**, create a spline from either:
  - Data points entered manually into the Spline Editor.
  - Imported test data from a file.



- **Then**, reference the spline through a spline function used in a motion or force. Several interpolation methods are available (using the function type):
  - Cubic-fitting method (CUBSPL)
  - Akima-fitting method (AKISPL)
  - B-spline method (CURVE)

AKISPL Function

Syntax for AKISPL function

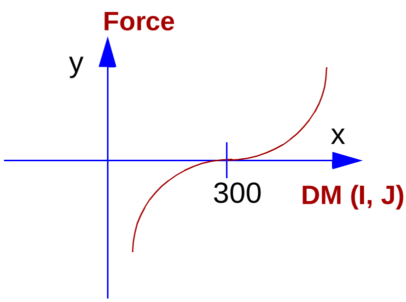
AKISPL (x, z, spline, iord)

- **x** - Independent variable specifying the value along the x-axis.
- **z** - Optionally, a second independent variable specifying the value along the z-axis of the surface being interpolated.
- **spline** - Spline used to map the one-to-one correspondence of the dependent variables (y) against independent variable values (x or z).
- **iord** - An integer variable that specifies the order of the interpolated point (usually 0, but can be 1 or 2).

Example of an AKISPL function

AKISPL (DM(I, J), 0, spline_1, 0)

DM (I, J)	Force
x	y
150	-1000
200	-200
250	-50
300	0
350	50
400	200
450	100

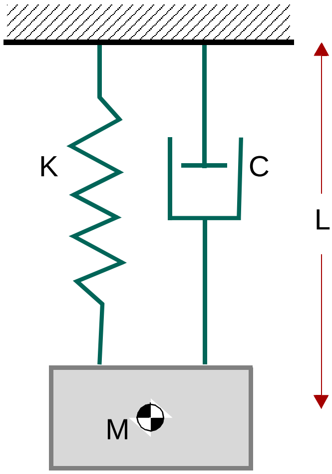


Note: You can create the CUBSPL and CURVE functions exactly as you create the AKISPL function.

Workshop 15—Nonlinear Spring

Problem statement

Investigate the differences between a linear spring and a nonlinear spring using a spline function.



M: 187.224 Kg

K: 5.0 N/mm

L_0: 400mm

F_0: 0

Start the workshop

Start by importing the model you created in the last workshop. Note that this file is not in the current working directory.

To start the workshop:

1. Start ADAMS/View from the directory ***exercise_dir*/mod_15_spring**.
2. From the directory ***exercise_dir*/mod_14_spring_damper**, import the model that you created in the previous module.

   If you need a fresh copy of the model, import the command file spring_damper_completed.cmd from the directory *exercise_dir*/mod_14_spring_damper/completed.

Replace the predefined spring-damper

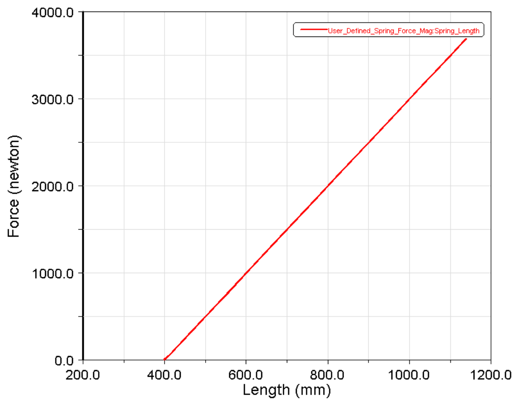
Now you will replace the spring-damper already in the model with a user-defined linear spring damper.

To replace the spring-damper:

1. Delete the predefined spring-damper.
2. Create a user-defined spring using a single-component, action-reaction (two bodies), force with a custom setting.
3. Modify the force function so it behaves like a linear spring:
   - Use the Function Builder.
   - Use the I and J markers of the force object to define the displacement magnitude function within the spring force calculation. You may want to rename these markers for easier reference from within the Function Builder.

   Tip: Create an SFORCE with the K and C characteristic to see what your function expression should look like.You want to create the same expression in Function Builder for your SFORCE using the DM and VR assist dialog boxes.
4. Create a measure, named **spring_force**, to measure the force magnitude in the single-component force.
5. To see oscillations, run a **2-second**, **50-step** simulation.
6. Plot **spring_force** versus **spring_length**.

   Note that the plot has a slope of 5. This is because the value of the stiffness coefficient for the user-defined spring-damper is 5 N/mm.



7. Save the simulation results as **linear_force**.

Change the linear spring to a nonlinear spring

In this section, you change the spring damper you just created to a nonlinear spring. You'll import spring stiffness data to define the spring properties.

To change the spring:

1 To import the spring stiffness data, from the **File** menu, select **Import**.

2 Set the following parameters, and then select **OK**.

- **File type**: **Test Data**
- **Create Splines**
- **File to Read**: ***exercise_dir*/mod_14_spring/spring_data.txt**
- **Independent Column Index**: **1** (Since the first column is the independent column.)
- **Units**: **Force**
- **Model Name**: **.spring_mass**

3 To open **SPLINE_1** in the Spline Editor, from the **Build** menu, point to **Data Element**, point to **Spline**, and then select **Modify**.

4 View the plot to understand the relationship between the deformation (x-axis) and stiffness force (y-axis).

Tip: In the upper right corner, set **View As** to **Plot**.

5 Replace the force function describing the single-component force with an Akima spline function.

Ensure that:

- Independent variable takes into account the spring's free length.
- Signs are correct.
- As the spring lengthens, the single-component force should apply a negative force (tension).

Tip: Use trial and error for the replacement.

Compare the linear and nonlinear forces'

To compare the forces:

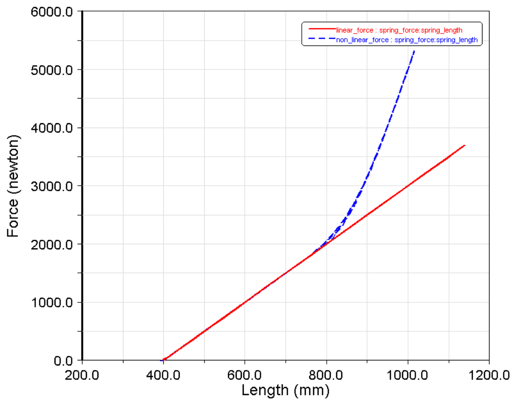
1 Verify that the nonlinear spring is working properly by running a **2-second, 50-step** dynamic simulation.

2 Save the simulation results as **non_linear_force**.

3 Overlay the two plots:

- **spring_force vs. spring_length** for the **linear_force simulation**
- **spring_force vs. spring_length** for the **non_linear_force simulation**

Note that in the nonlinear case, the curve changes slope as **spring_length** increases.



Save your work

To save your work:

1 Save your model.

2 Exit ADAMS/View.

Optional tasks

1 Create a spline in the Spline Editor to represent a nonlinear damping force versus velocity:

From the **Build** menu, point to **Data Element**, point to **Spline**, and then select **New**.

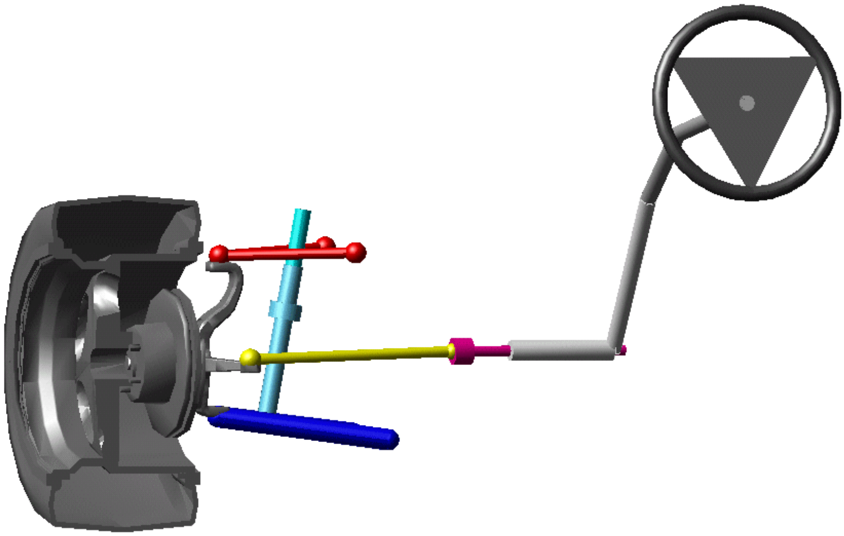
2 Add the spline function representing a damping force to the single-component force function so you have a nonlinear spring-damper in between the mass and ground.

Module review

1 What are the four inputs for a spline function?

16 SUSPENSION-STEERING SYSTEM II

Investigate the effect on toe angle when you replace the idealized constraint between the lower control arm and ground with bushings, while the steering wheel is held at an angle of 0^o.



What's in this module

Bushings

Definition of a bushing

- Pre-defined force.
- Represents compliance:
  - Between two bodies.
  - Along or about three vectors.

Characteristics of a bushing

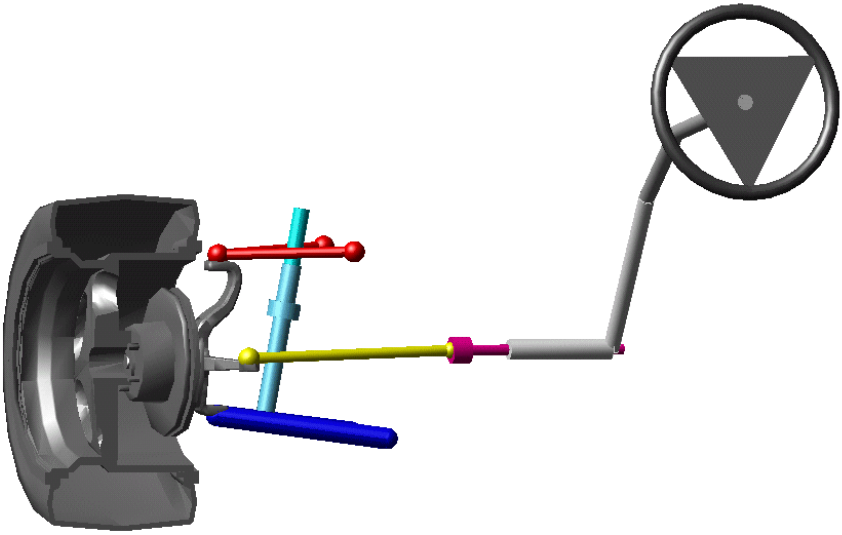
Characteristic:	Description:
Bodies	Two
Points of application	Two (action force at I marker and reaction force at J marker)
Vector components	Three translational and three rotational
Orientations	Based on the J marker
Magnitudes (F_x, F_y, F_z + T_x, T_y, T_z)	Pre-defined equation based on: ■ Stiffness matrix, $[\boldsymbol{K}]$ ■ Damping matrix, $[\boldsymbol{C}]$

See also: Forces Tables (Incomplete), on page 321.

Workshop 16—Suspension-Steering System II

Problem statement

Investigate the effect on toe angle when you replace the idealized constraint between the lower control arm and ground with bushings, while the steering wheel is held at an angle of 0^{o}.



Model description

- The model is the short-long-arm front suspension model combined with a steering model that you created in the previous workshop.
- A spring-damper has been added to represent the force input of a coil-over shock.
- Currently, a revolute joint connects the lower control arm to the frame of the vehicle.
- You are going to replace the revolute joint with two bushings and investigate the differences.

Start the workshop

To start the workshop:

1. Start ADAMS/View from the directory ***exercise_dir*/mod_16_susp_steer_2**.
2. Import the command file **susp_steer_2_start.cmd**.

Run a baseline simulation

You'll start by running a simulation with the model as it currently is to see how it performs with a revolute joint.

To run a baseline simulation:

1. Verify that the steering wheel angle is a constant 0^{o}.
2. Run a simulation for **1 second** with **50** output steps.
3. Save the simulation results as **with_joint**.

Deactivate the revolute joint

Now, instead of removing the revolute joint, you'll just deactivate it so it is not used in simulations.

To deactivate the revolute joint:

1. Right-click the **lower_grnd_rev** revolute joint that currently exists between **Lower_Arm** and ground.
2. Select **(De)activate**.

Create bushings between Lower_Arm and ground

You will need to create two bushings because there are two connection points between Lower_Arm and ground.

To create bushings:

1. From the Main toolbox, point to the **Create Forces** toolstack, and select the **Bushing**  tool.
2. Create the rear bushing with the following properties, using the options **2 Bod - 1 Loc**, **Pick Feature**:
   - **First Part - Lower_Arm**
   - **Second Part - ground**
   - **Location - HP4**
   - **Direction Vector (+z axis) - Z-direction of .Lower_Arm.bushing_ref marker**
3. Modify the bushing to reflect the following properties:

K_{matrix}	C_{matrix}	$Preload_{matrix}$
$\begin{bmatrix} 2.9e7 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2.9e7 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1e8 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1e6 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1e6 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 7.3e5 & 0 & 0 & 0 & 0 & 0 \\ 0 & 7.3e5 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1.5e6 & 0 & 0 & 0 \\ 0 & 0 & 0 & 4e5 & 0 & 0 \\ 0 & 0 & 0 & 0 & 4e5 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$

4. Create the forward bushing with the following properties:
   - **First Part - Lower_arm**
   - **Second Part - ground**
   - **Location - HP5**
   - **Direction Vector (+z axis) - Z-direction of .Lower_Arm.bushing_ref marker**
5. Modify the bushing to reflect the properties given in Step 3 above.

Run a simulation to view the effect of adding the bushing

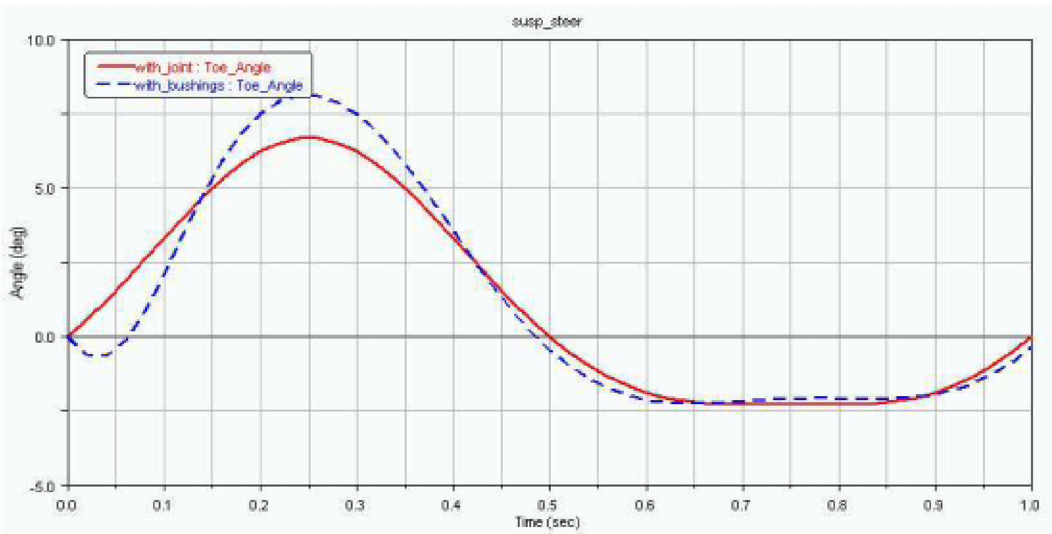
To run a simulation:

1. Run a static simulation followed by a dynamic simulation for **1 second** with **50** output steps.
2. Save the simulation results as **with_bushings**.

Review the results

To review the results:

1. Start ADAMS/PostProcessor.
2. Create a plot that contains the toe_angle measure using the simulation results **with_joint** and **with_bushings** as a function of time.
3. Estimate the difference in the maximum toe angle between the two simulations and use it to answer Step 4 in Module review, on page 219.



Save your work

To save your work:

1. Save your model.
2. Exit ADAMS/View.

Optional tasks

1. Replace the revolute joint between **Upper_Arm** and ground with two bushings. Use the same bushing properties given for the bushing between **Lower_Arm** and ground.
2. Run a static simulation followed by a dynamic simulation for **1 second** with **50** output steps.
3. Save the simulation results as **with_all_bushings**.
4. Compare these toe_angle results with those from the previous two simulations.

Module review

1. What was the approximate difference in the maximum toe angle that was a result of removing the revolute joint and replacing it with bushings?

   ____

   ____

2. Why did you perform a static simulation before the dynamic simulation after you added the bushings?

   ____

   ____

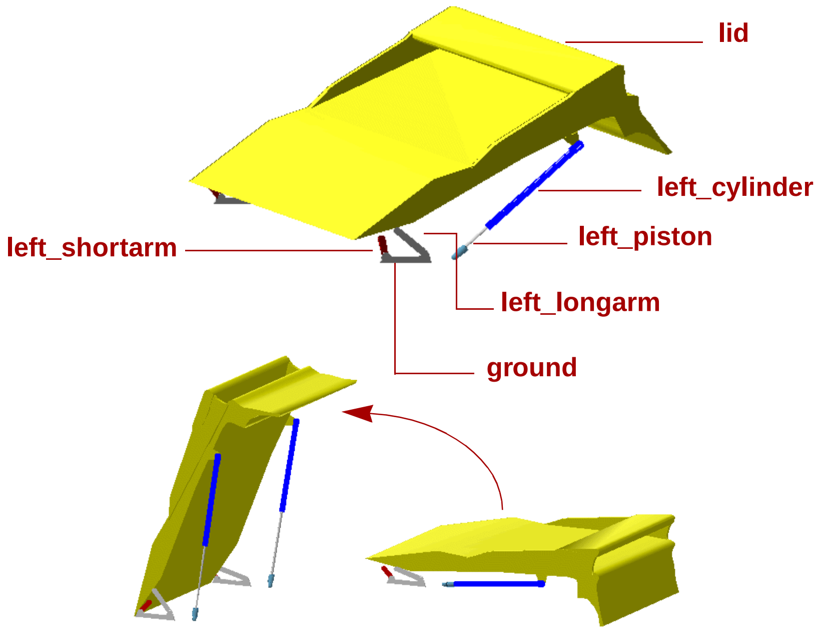
3. Why did you not have to perform a static simulation before the dynamic simulation when the Lower_Arm was constrained with the revolute joint?

   ____

   ____

17 HATCHBACK I

Create the forces required to open the hatchback for the given Mazda MX-6 model.



What's in this module:

Impact Functions

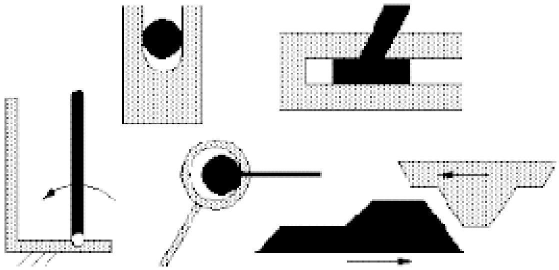
Impact functions in ADAMS

- Are used with user-defined force elements to model contacts, impacts, collisions, and so on.
- Mimic nonlinear spring and damping forces that turn on and off depending on the distance between two objects.
- Just like a compression-only spring-damper, ADAMS turns the force on when the distance between two objects, q, becomes less than the user-specified reference distance, q_0:

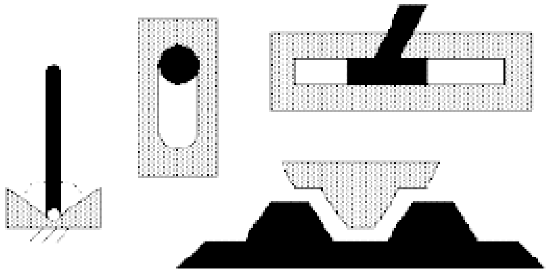
$$F_{\text{IMPACT}} = \text{Off}, \text{ if } q > q_0$$

$$F_{\text{IMPACT}} = \text{On}, \text{ if } q \le q_0$$

Applications of one-sided impact functions (IMPACT)



Applications of two-sided impact functions (BISTOP)



Impact Functions...

Syntax for IMPACT function

IMPACT(q, $\dot{q}$, q_1, k, e, c_{max}, d)

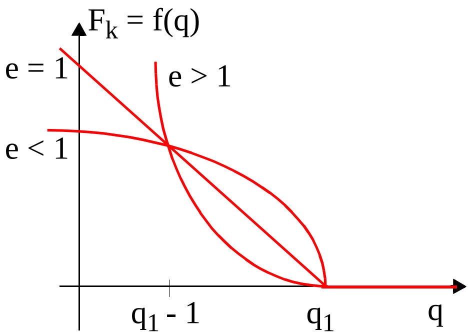
- q - Actual distance between the two objects (defined with a displacement function)
- $\dot{q}$ - Time rate of change of the variable q
- q_1 - Trigger distance used to determine when the contact force turns on and off; it should be specified as a real, constant value
- k - Stiffness coefficient
- e - Stiffness force exponent
- c - Damping coefficient
- d - Damping ramp-up distance

In ADAMS, the one-sided impact force is calculated as

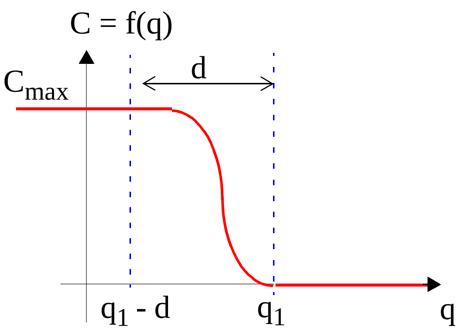
$$F = 0 \text{ if } q > q_1$$

$$F = k(q_1 - q)^e - c_{max}\dot{q}*\text{STEP}(q, q_1 - d, 1, q_1, 0) \text{ if } q \le q_1$$

Compression-only spring force from one-sided IMPACT function



Compression-only damping force from one-sided IMPACT function



Velocity Functions

Definition of velocity and acceleration functions

- Returns scalar portions of velocity or acceleration vector components (translational or rotational).

Syntax for velocity functions

- VM(I,[J])
- VR(I,[J])
- VX, VY, VZ(I,[J],[R], [L])

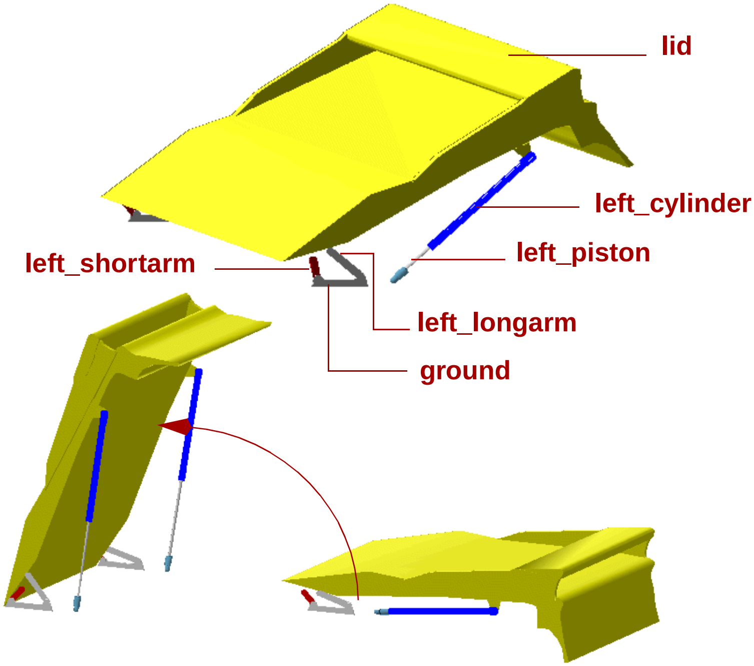
Notes:

- Velocity function, VR, is used to define velocity along the line of sight, which is commonly used in spring-dampers.
- If the markers are separating: VR > 0.
- If the markers are approaching: VR < 0.

Workshop 17—Hatchback I

Problem statement

Create the forces required to open the hatchback for the given Mazda MX-6 model.



Model description

- When compressed, the force in each gas shock is 460 Newtons.
- The motion of the assembly is limited by stops in the gas shocks at full extension.
- Parts are constrained as shown next:

Location:	Parts:	Type:
POINT_1	left_shortarm and ground	Revolute
POINT_4	left_longarm and ground	Revolute
POINT_2	left_shortarm and lid	Spherical
POINT_6	left_cylinder and lid	Spherical
POINT_8	right_cylinder and lid	Spherical
POINT_3	lid and left_longarm	Hooke
POINT_5	ground and left_piston	Hooke
POINT_7	ground and right_piston	Hooke
POINT_56	left_piston and left_cylinder	Translational
POINT_78	right_piston and right_cylinder	Translational

Start the workshop

To start the workshop:

1. Run ADAMS/View from the directory ***exercise_dir*/mod_17_hatchback_1**.
2. Import the model command file **hatchback_start.cmd**.

Deactivate movable parts not used for simulation

To deactivate parts:

1. Deactivate **right_shortarm**.

   Tip: Right-click the part.
2. Deactivate **right_longarm**.

Create forces to represent gas shocks with limit stops

To create forces:

1 Create markers on the relevant parts at **POINT_5** and **POINT_6**, and use these markers to create the spring-damper in the next step.

2 Create a spring-damper between **left_piston** and **left_cylinder** at **POINT_5** and **POINT_6** using the following parameters:

- **Stiffness: 0.21578 (N/mm)**
- **Damping: 1.0 (N-sec/mm)**
- **Preload: 460 N**

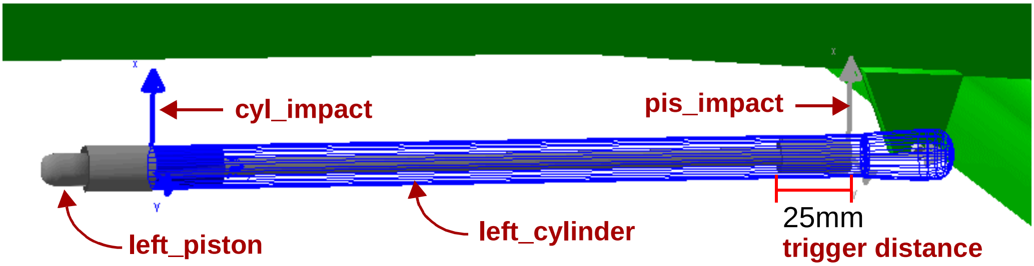
3 Create a spring-damper between **right_piston** and **right_cylinder** at **POINT_7** and **POINT_8** using the following parameters:

- **Stiffness: 0.21578 (N/mm)**
- **Damping: 1.0 (N-sec/mm)**
- **Preload: 460 N**

4 Create an impact force between **left_piston** and **left_cylinder** at markers **pis_impact** and **cyl_impact**, using the following parameters:

- **Stiffness: 1e5 (N/mm)**
- **Stiff exponent: 1.01**
- **Damping: 100 (N-sec/mm)**
- **Bump stops (trigger distance): 25 mm**
- **Damping ramp-up distance: 1e-3 mm**

Note that **pis_impact** belongs to the **left_piston** and **cyl_impact** belongs to the **left_cylinder**.



5 Create an impact force between **right_piston** and **right_cylinder** at markers **pis_impact** and **cyl_impact** using the following parameters:

- **Stiffness: 1e5 (N/mm)**
- **Stiff exponent: 1.01**
- **Damping: 100 (N-sec/mm)**
- **Bump stops (trigger distance): 25 mm**
- **Damping ramp-up distance: 1e-3 mm**

Simulate the model

Simulate the model to make sure that the hatchback opens and stops at a reasonable angle.

Save your work

To save your work:

1 Save your model.

2 Exit ADAMS/View.

Optional tasks

Do not save these changes you make in this section because you will use this model in the next workshop.

- Close the lid after it has opened all the way.

  There are many ways to do this. Use the method with which you are most comfortable.

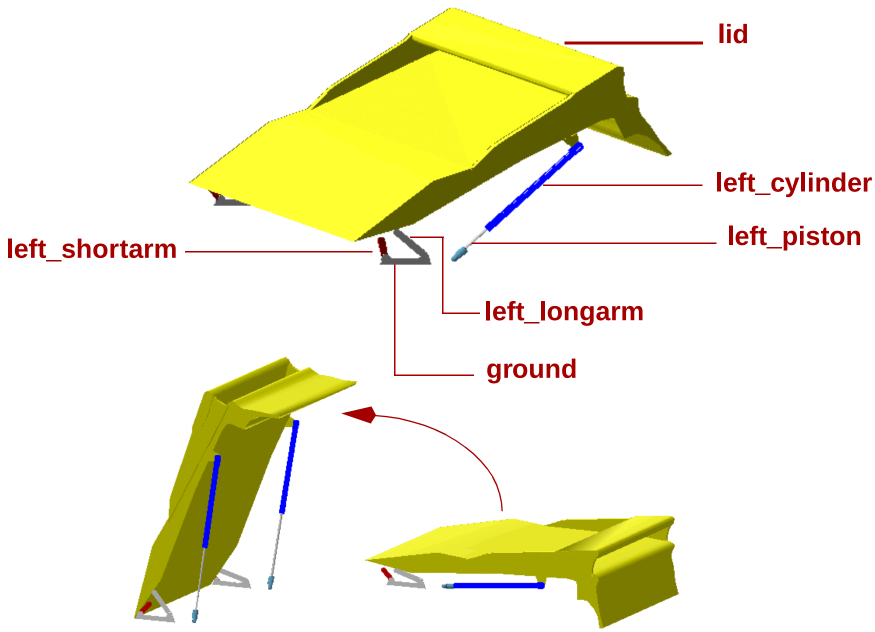
Module review

1 Are there any limitations to the trigger distance used in an IMPACT function? In other words, can you choose *any* value?

2 Could you use the IMPACT function to represent a hardening compression-only spring?

18 HATCHBACK II

Find the approximate maximum force at the winglet required to close the lid in three seconds, for the given Mazda MX-6 hatchback model.



What's in this module:

STEP Function

Definition of a STEP function

- In ADAMS, the STEP function approximates an ideal mathematical step function (but without the discontinuities).
- Avoid discontinuous functions because they lead to solution convergence difficulties.
- The STEP function steps quantities, such as motions or forces, up and down, or on and off.

Note: A STEP function is used when a value needs to be changed from one constant to another.

Syntax for STEP function

STEP (q, q_1, f_1, q_2, f_2)

where:

q - Independent variable

q_1 - Initial value for q

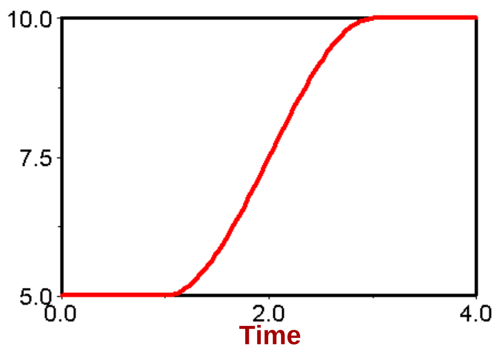
f_1 - Initial value for f

q_2 - Final value for q

f_2 - Final value for f

Note: q1 < q2

Example



STEP (time,1,5,3,10)

Scripted Simulations

In ADAMS/View there are two ways to run a simulation

- Scripted
- Interactive

Simulation scripts

- Let you program the simulation before submitting the simulation.
- Let you quickly repeat a simulation with the same set of parameters.
- Let you perform more sophisticated simulations.
- Are required for design studies, design of experiments, and optimization simulations.
- Simulation scripts are children of a model, and are, therefore, saved in a command file.

Types of scripted simulations in ADAMS/View

- Simple run
- ADAMS/View commands
- ADAMS/Solver commands

Scripted simulations based on ADAMS/Solver commands

- ADAMS/Solver commands let you perform sophisticated simulations, such as:
  - Changing model parameters during a simulation.
  - Using different output step sizes over different simulation intervals (versus specifying only one duration and output step size).
  - Using different solution parameters (such as convergence tolerance) over different intervals.
- Example of a simulation script that changes model topology while you work on your model:

```
output/noseparator
simulate/dynamic, end=3.0, steps=30
deactivate/joint, id=3
simulate/dynamic, duration=2.0, steps=200
```

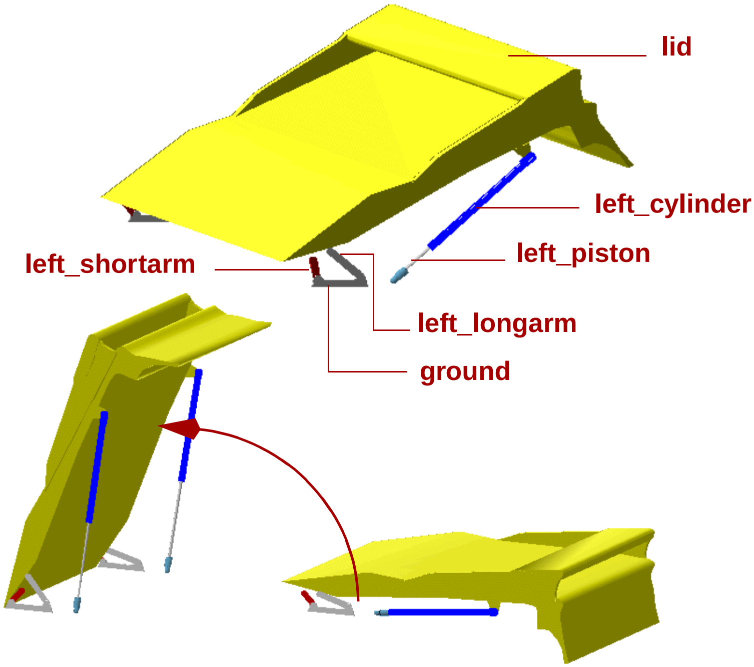
Before:

After:

Workshop 18—Hatchback II

Problem statement

Find the approximate maximum force at the winglet required to close the lid in three seconds, for the given Mazda MX-6 hatchback model.



Model description

In this workshop, you use the model you built in the module Hatchback I.

Start the workshop

To start the workshop:

1 Start ADAMS/View from the directory ***exercise_dir*/mod_18_hatchback_2**.

2 From the directory ***exercise_dir*/mod_17_hatchback_1**, import the model that you created in the previous module.

If you need a copy of the model, import the command file hatchback_1_completed.cmd from the directory *exercise_dir*/mod_17_hatchback_1/completed.

Determine steady-state rotation of left_shortarm

To determine rotation:

1. Create a measure named **shortarm_rotation** for the rotation of the **left_shortarm** with respect to **ground**. Measure the rotational displacement of the **left_shortarm_rev** joint.

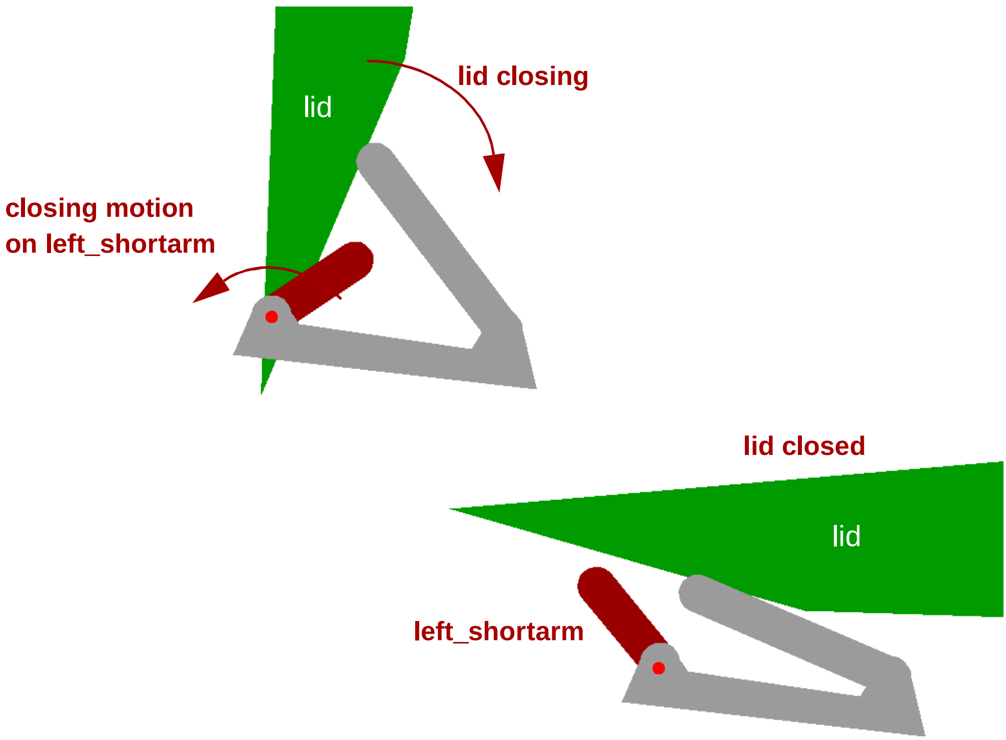
   Tip: Select .ground.MAR_7 as the From/At marker to ensure the correct sign on the angle measure.

2. Plot the **shortarm_rotation** versus **time**.
3. From the **shortarm_rotation** plot, determine the steady-state angle of the **left_shortarm**.

Close the lid

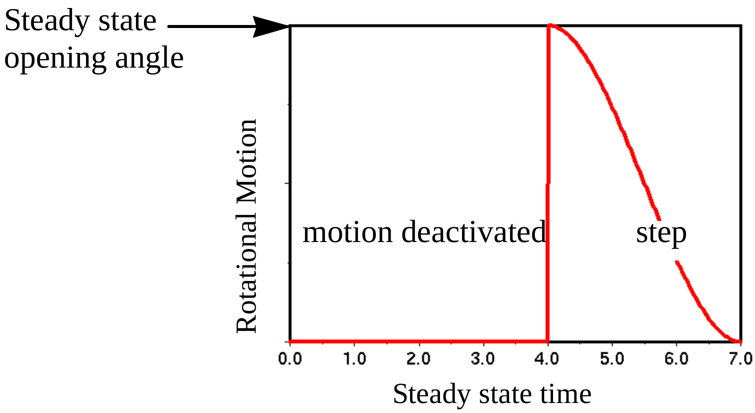
Currently the lid opens because of the preload in the springs and stops opening because of the impact forces.

You need to rotate the left_shortarm to close the lid, as shown next. To rotate the left_shortarm, apply a joint motion to the left_shortarm_rev joint.



To create a motion to close the lid:

1. Create a joint motion on the **left_shortarm_rev** joint.
2. Use a STEP function to modify the motion to drive the **lid** to the closed position:
   - Start the STEP function at the steady-state rotation of the **left_shortarm** at **4** seconds (determined earlier).
   - End the STEP function at **0**^{o} rotation of the **left_shortarm** at **7** seconds.



Perform a scripted simulation

In this section, you'll create a simulation script containing ADAMS/Solver commands that deactivate the motion and run a simulation, then activate the motion and run a second simulation.

To create the script:

1. From the **Simulate** menu, point to **Simulation Script**, and then select **New**.
2. Set **Script Type** to **ADAMS/Solver Commands**.
3. Enter the following ADAMS/Solver commands:

```
DEACTIVATE/MOTION, id=1
SIMULATE/DYNAMIC, END=4, STEPS=40
ACTIVATE/MOTION, id=1
SIMULATE/KINEMATIC, END=7, STEPS=30
```

4. Select **OK**.

To perform a scripted simulation:

1 From the **Simulate** menu, select **Scripted Controls**.

2 Enter the name of the script that you created.

3 Select the **Play** tool.

Create measures

To create measures:

1 Create a motion measure where the characteristic is the torque required to close the lid.

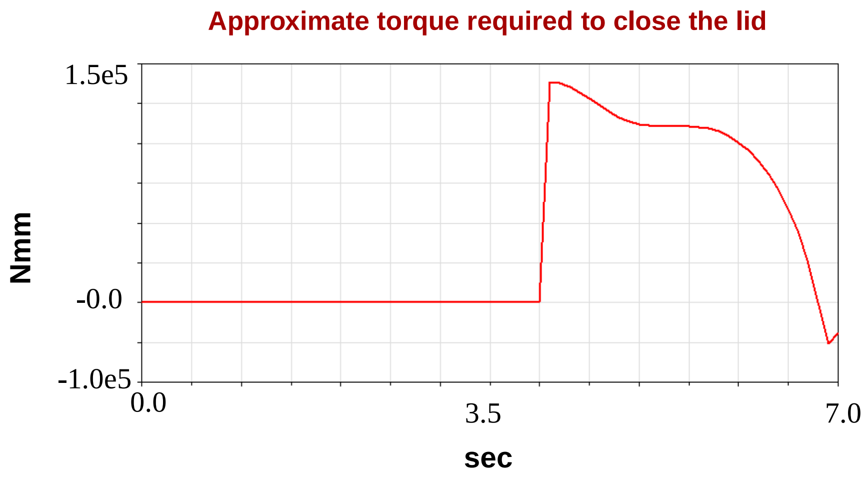
2 Deactivate this measure because it is dependent on the motion that is deactivated when the scripted simulation starts.

3 From the **Edit** menu, select **Deactivate**.

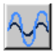
Inspect measures

To inspect measures:

1 Plot **reaction torque** versus **time**.



2 Edit the curve to plot the approximate force required to lower the lid in 3 seconds.

To edit the curve use the **Scale a Curve** tool  to divide motion torque by a moment arm of 700 mm:

- To access the **Scale a Curve** tool, in ADAMS/PostProcessor, from the **View** menu, point to **Toolbars**, and then select **Curve Edit Toolbars**.

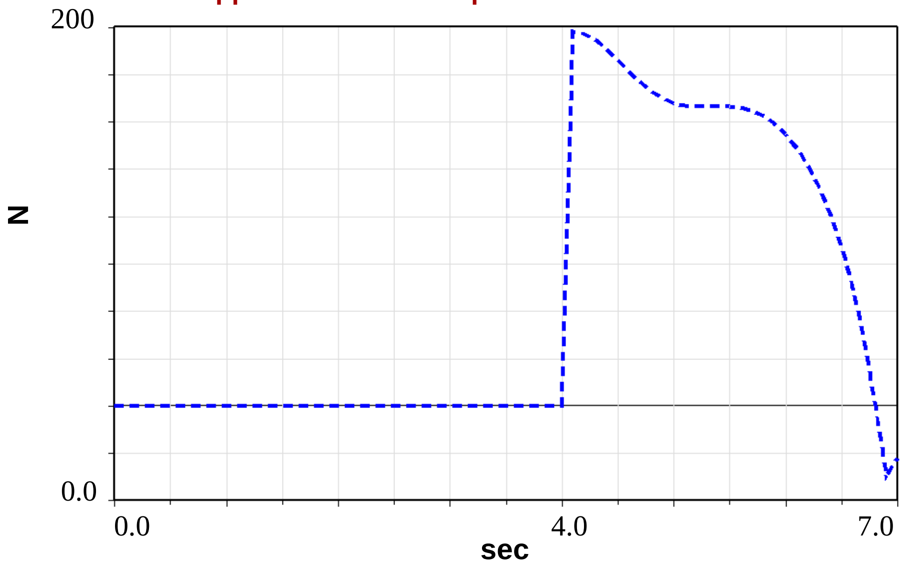
  A new toolbar appears.

- Select the **Scale a Curve** tool, and follow the instructions on the status bar.

3 Delete the motion torque curve to display only the force curve.

Tip: Use appropriate time limits (4 to 7 sec).

Approximate force required to close the lid



4 Note the approximate maximum force required to close the lid. Use the value to answer Question 1 in Module review on page 240.

Save your work

To save your work:

1 Save your model.

2 Exit ADAMS/View.

Optional tasks

1. Deactivate the motion.
2. Apply a single-component force perpendicular to the winglet.

   The magnitude of the force should be zero from 0-4 sec, so the lid can open.

   The magnitude should step up to a constant value, enough to close the lid back to or past its original position in three seconds (4-7 seconds).
3. Simulate the model.

   Because you do not modify the model topology, you do not need a script for this simulation.
4. Install stops that keep the lid from closing too far (or past its original location).

Module review

1. What is the approximate maximum force required to close the lid?

   ____

   ____

2. Is it possible to modify a force from one constant value to another *instantaneously*, such as shutting off a motor's torque?

   ____

   ____

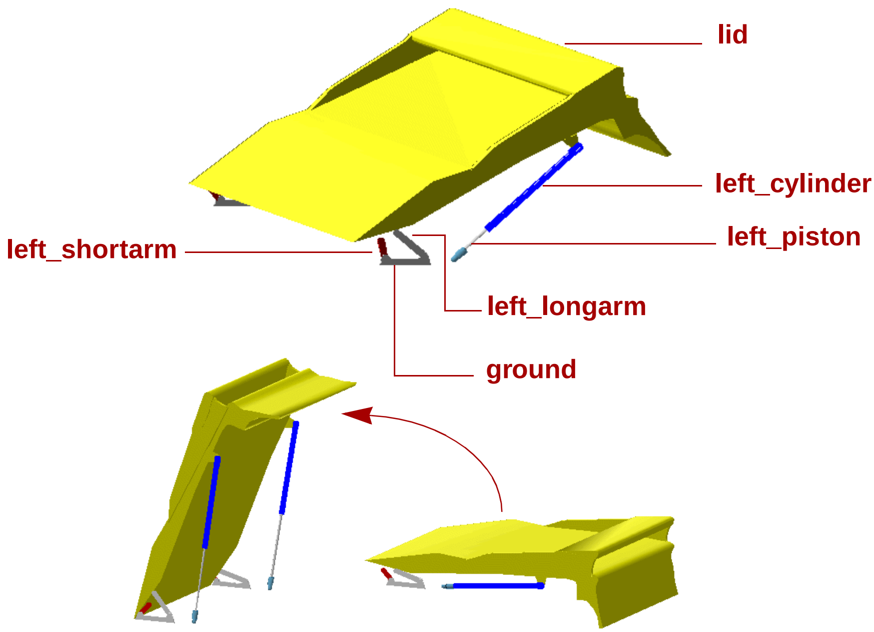
3. Is it possible to use different output step sizes over different intervals by submitting an interactive simulation?

   ____

   ____

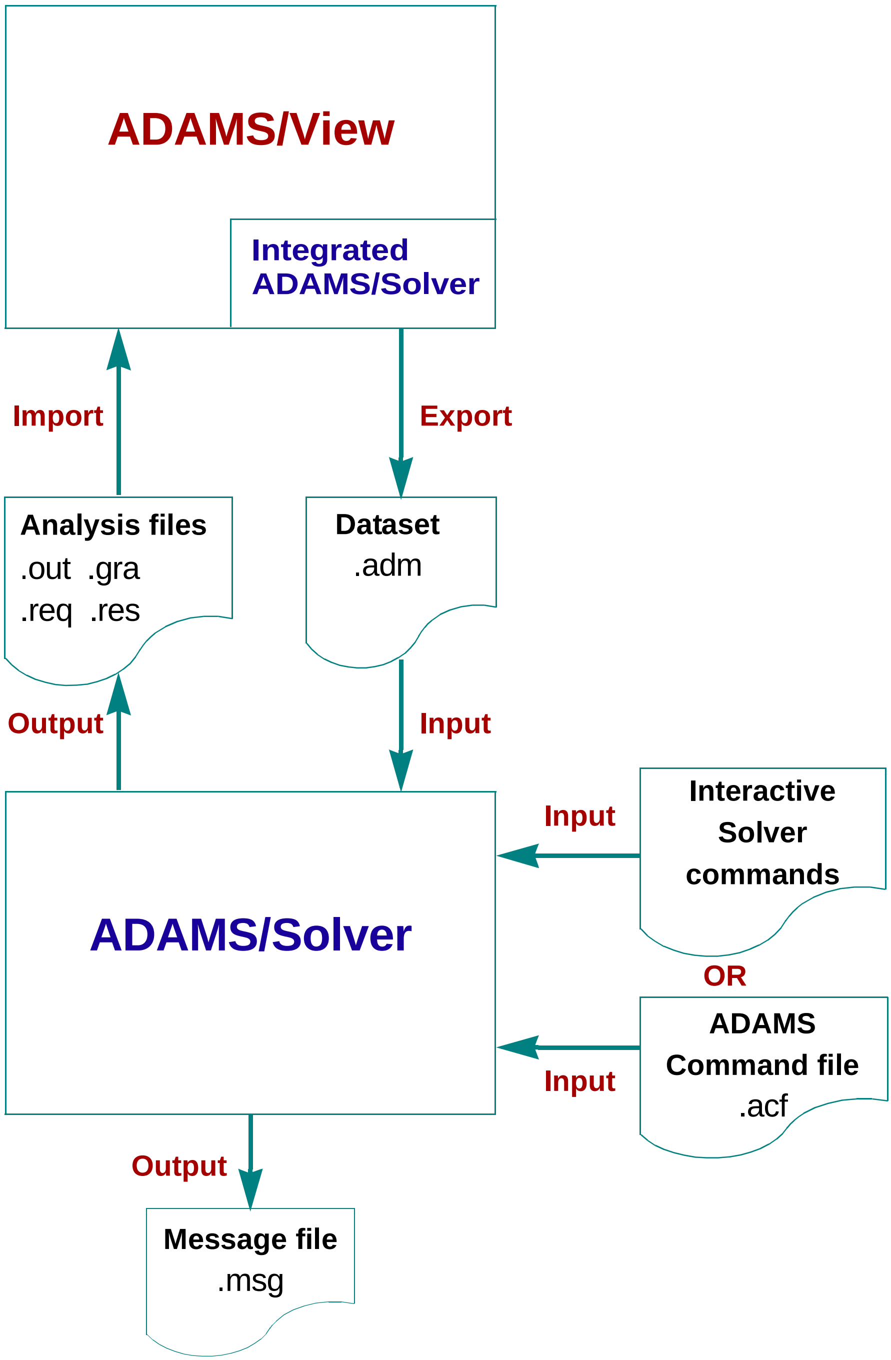
19 HATCHBACK III

Use ADAMS/Solver to simulate the given Mazda MX-6 hatchback model.



What's in this module:

ADAMS/Solver Overview



Files in ADAMS/Solver

ADAMS/Solver dataset files (.adm)

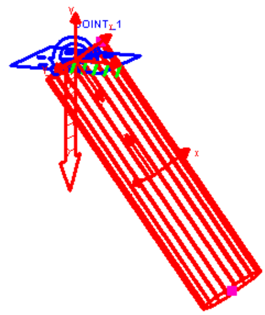
- Statements define an element of a model such as a part, constraint, force, and so on.
- Functions are numeric expressions that define the magnitude of an element such as a force or motion.

ADAMS/Solver command files (.acf)

Commands define an action that needs to be taken during a simulation.

See also: ADAMS/Solver Commands on page 234

Example of an ADAMS/Solver Dataset (.adm) File




```
Pendulum
!-------------------------------- SYSTEM UNITS --------------
UNITS/FORCE = NEWTON, MASS = KILOGRAM, ,LENGTH =
MILLIMETER, TIME = SECOND
!------STATEMENTS FROM ORIGINAL DATASET -----
!
MATERIAL/1, NAME = steel, YOUNGS_MODULUS = 2.07E+005,
, POISSONS_RATIO = 0.29
, DENSITY = 7.801E-006
!
PART/1, GROUND
!
MARKER/1, PART = 1
!
MARKER/5, PART = 1, QP = 175, -225, 0
!
PART/2, MASS = 70.94, CM = 3, IP = 2.01E+006, 1.80E+005
, 2.01E+006, MATERIAL = steel
!
MARKER/2, PART = 2, REULER = 37.87498365D, 90D, 0D
!
MARKER/3, PART = 2, QP = 175, -225, 0, REULER = 37.87498365D, 0D, 0D
!
MARKER/4, PART = 2
!
GRAPHICS/1, CYLINDER, CM = 2, LENGTH = 570.08, RADIUS = 71.26
!
JOINT/1, REVOLUTE, I = 4, J = 1
!
REQUEST/1, DISPLACEMENT, I = 3, J = 5, RM = 5
ACCGRAV/JGRAV = -9806.65
OUTPUT/REQSAVE, GRSAVE
RESULTS/
!
MOTION/1, ROTATIONAL, JOINT = 1, FUNCTION = 30.0d * time
!
END
```

Simulations in stand-alone ADAMS/Solver

- Interactive:
  - Not scripted: enter commands one by one.
  - Scripted: use an ADAMS/Solver command file (.acf).
- Batch - Run multiple jobs in the background using an ADAMS/Solver command file (.acf).

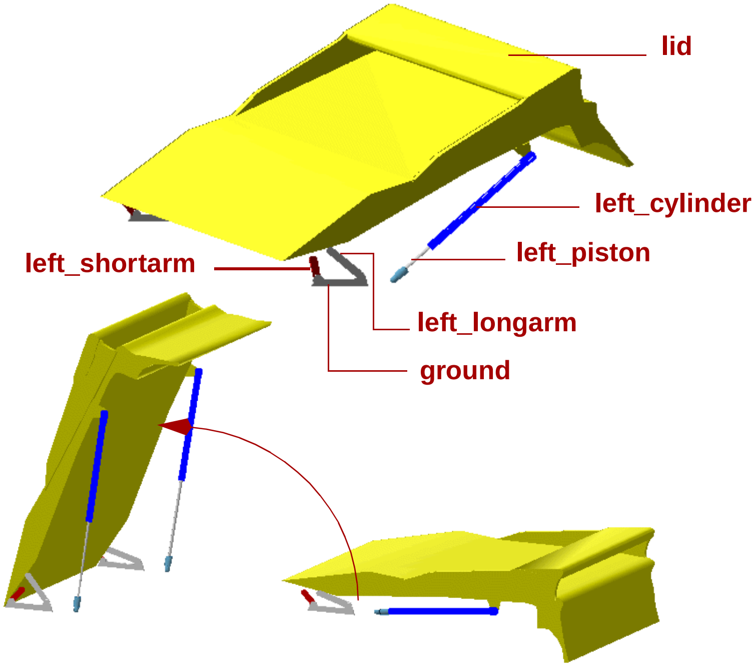
Note: ADAMS/Solver command files must start with the name of the model to be analyzed and must end with a STOP command.

You can run simulations externally in ADAMS/Solver from within ADAMS/View

Workshop 19—Hatchback III

Problem Statement

Use ADAMS/Solver to simulate the given Mazda MX-6 hatchback model.



Model description

In this workshop, you use the model you saved in Workshop 18—Hatchback II, on page 235.

Start the workshop

1 Start ADAMS/View from the directory ***exercise_dir*/mod_19_hatchback_3**.

2 From the directory ***exercise_dir*/mod_18_hatchback_2**, import the model that you created in the previous module.

If you need a copy of the model, import the command file hatchback_2_completed.cmd from the directory *exercise_dir*/mod_18_hatchback_2/completed.

Export a dataset (.adm) file:

1 From the **File** menu, select **Export**.

2 Enter the following, and then select **OK**.

- **File Type**: **ADAMS/Solver dataset.**
- **File Name**: **hatchback.adm.**

Create an ADAMS command file (.acf)

1 Open a text editor (UNIX: vi or Jot; Windows NT: Notepad or Wordpad), and create an ADAMS/Solver command file (.acf) that contains the following commands:

```
hatchback.adm (the .adm extension is optional)
hatchback_test1
OUTPUT/NOSEPARATOR
DEACTIVATE/MOTION, id=1
SIMULATE/DYNAMIC, END=4, STEPS=40
ACTIVATE/MOTION, id=1
SIMULATE/KINEMATIC, END=7, STEPS=30
STOP
```

2 Save the file as **hatchback.acf**.

Perform a simulation in stand-alone ADAMS/Solver

To perform a simulation in stand-alone ADAMS/Solver, you use the ADAMS Program Menu, a menu- and text-based interface that allows you to enter information on the command line.

To perform simulation:

1. Perform a simulation in stand-alone ADAMS/Solver using the command file. Use the commands:
   - **adamsxx -c** (Displays the ADAMS Program Menu, -c is not needed on NT.)
   - **ru-s** (runs ADAMS/Solver with standard ADAMS executable.)
   - **i** (Sets interactive mode - you may not need this step on NT.)
   - **hatchback.acf** (Identifies the .acf ADAMS/Solver file and runs the simulation.)
   - **exit** (Exits ADAMS/Solver.)

Modify the dataset (.adm) file

Now change the spring stiffness in the .adm.

To change the spring stiffness:

1. In a text editor, open **hatchback.adm**.
2. Modify the value of spring stiffness (for both springs) to **0.24 N/mm**.
3. Save the file as **hatchback2.adm**.

Modify the ADAMS command file (.acf)

Modify the .acf to run with hatchback2.adm.

To modify the .acf:

1. In a text editor, open **hatchback.acf**.
2. Modify the first and second lines of the .acf file so they are:

   hatchback2

   hatchback_test2
3. Save the file as **hatchback2.acf**.

Workshop 19—Hatchback III...

Perform a simulation in stand-alone ADAMS/Solver:

- Using the new command file, perform a stand-alone ADAMS/Solver simulation.

Compare the results of the two simulations in ADAMS/View:

Import both sets of results (**hatchback_test1**, **hatchback_test2**) into ADAMS/View, and then compare them.

To import and compare the results:

1 From the **File** menu, select **Import**.

2 Enter the following, and then select **OK**.

- **File Type**: **ADAMS/Solver Analysis (.req, .gra, .res)**
- **Model Name**: **hatchback**

3 Inspect both simulations using animations and plotting.

4 Exit ADAMS/View.

Optional tasks

Simulate the model in ADAMS/Solver:

1. Simulate interactively but without a script (no .acf file).
2. In a UNIX shell, enter the following commands:
   - **adams10 -c**
   - **ru-standard**
   - **i**
   - **<CR>** (Because you do not have an ADAMS/Solver command file (.acf), press the enter key).

   ADAMS/Solver starts.
3. In ADAMS/Solver, enter:
   - **hatchback** (the name of your ADAMS Dataset (.adm) file)
   - **hatchback_test3** (the desired output file names .gra, .res, .out, and so on)

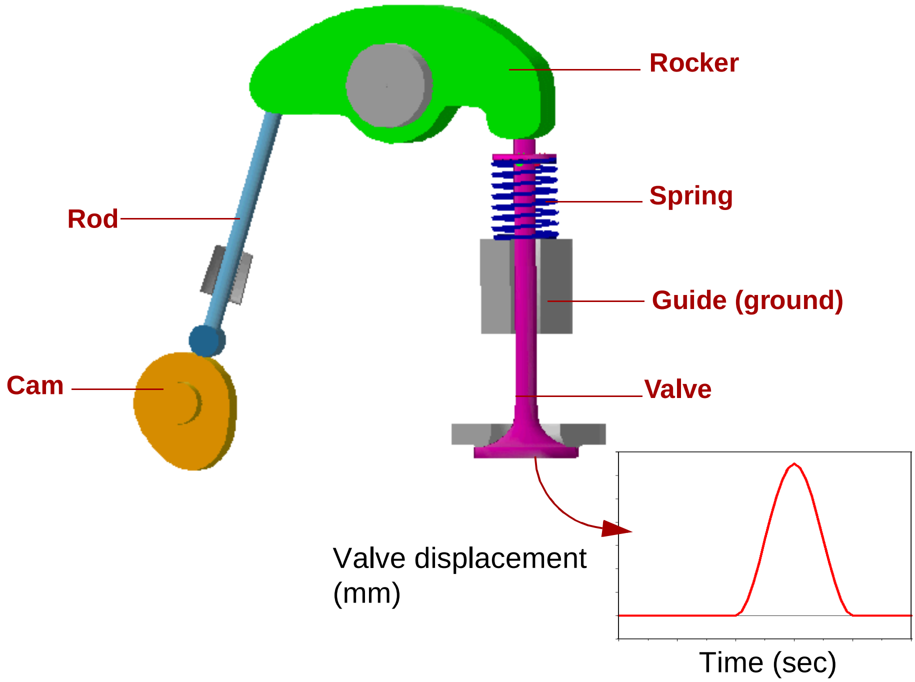
   ADAMS/Solver reads in the file and performs the assemble simulation.
4. At the ADAMS command prompt, enter commands one at a time in the same order in which they appear in the .acf file.
5. After entering all the commands, exit ADAMS/Solver and import your results into ADAMS/View so you can inspect them using animations and plotting.

Module review

1 What is the difference between a statement and a command?

20 CAM - ROCKER - VALVE

Design a cam profile based on desired valve displacement, and ensure that there is no follower liftoff when the cam is rotated at 3000 rpm.



What's in this module:

Definition of spline from point trace

- A point trace tracks a location of a marker or circle over time with respect to another part.
- ADAMS/View can create a two- or three-dimensional spline from a trace.
- Creating a spline from a trace is used to back-calculate (reverse engineer) the shape of an existing part based on its motion (cam synthesis).

Notes:

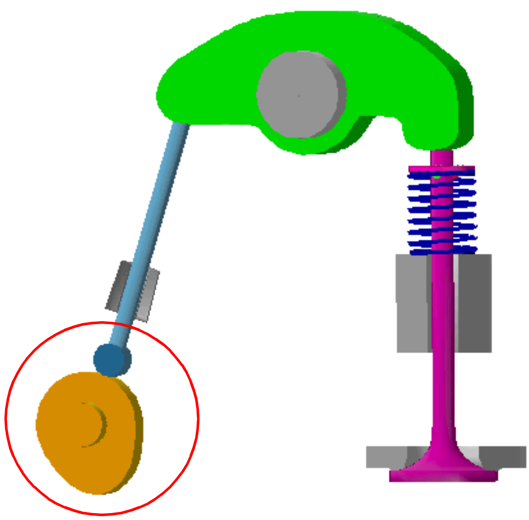
- When you trace an object and create a spline from it, the point or circle should move in a smooth, even path.
- If the path is closed, you should simulate for one cycle only.

Types of curve constraints in ADAMS

- Point-on-curve
- Curve-on-curve

Curve-on-curve constraints

- Used where a curved edge on one part always follows a curved edge on a different part.
- Remove two DOF.
- Modeling of curve-on-curve constraints requires:
    - Two parts
    - Two curves that will always remain in contact
- Typical applications include general cam-to-cam systems.



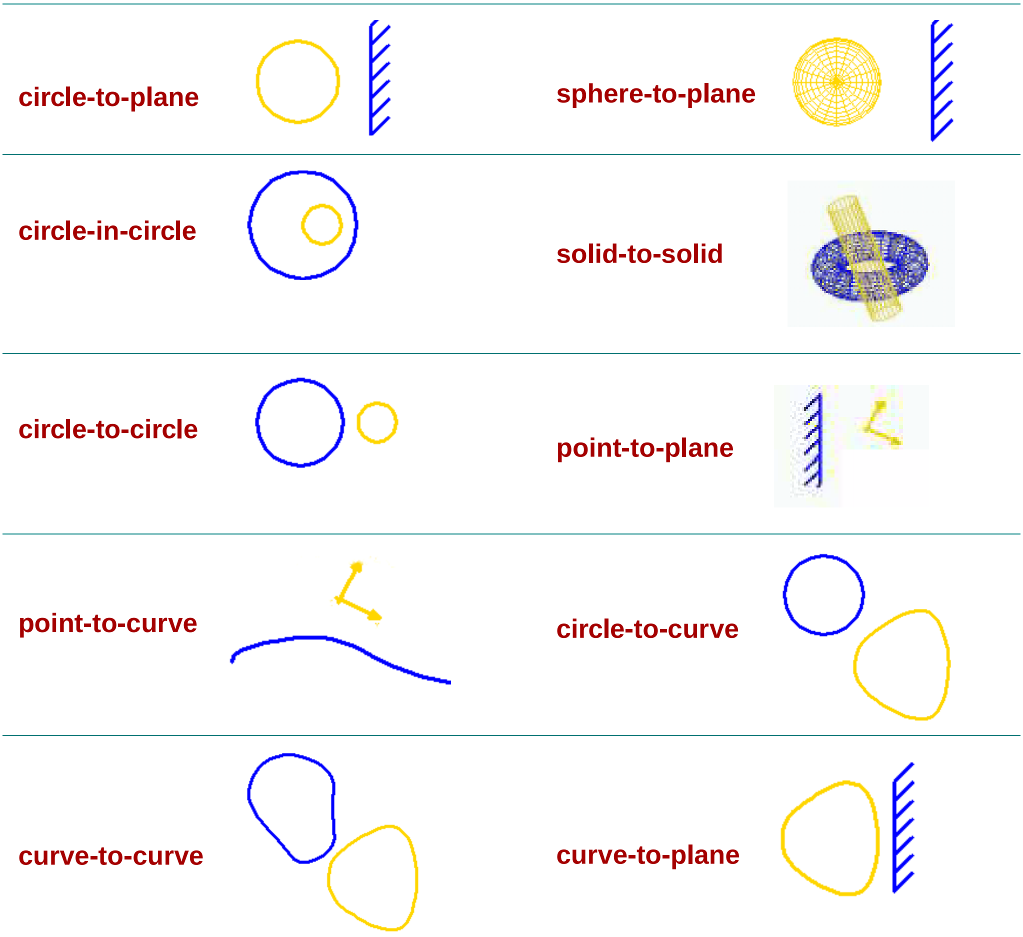
Note: Curve-on-curve constraints do not allow lift off.

See also: DOF removed by curve constraints on page 320

Contact forces

- Are special forces acting on parts that are activated when part geometries come in contact with each other.
- Have values that are determined by a set of contact parameters identical to those in the IMPACT function.
- Multiple contact forces can be combined to create more complex contacts.

Contact pairs in ADAMS



Automated Contact Forces...

Things to note while creating automated contact forces

- Circle-to-curve
- Point-to-curve
- Curve-to-curve

The xy planes of the two reference markers must be parallel.

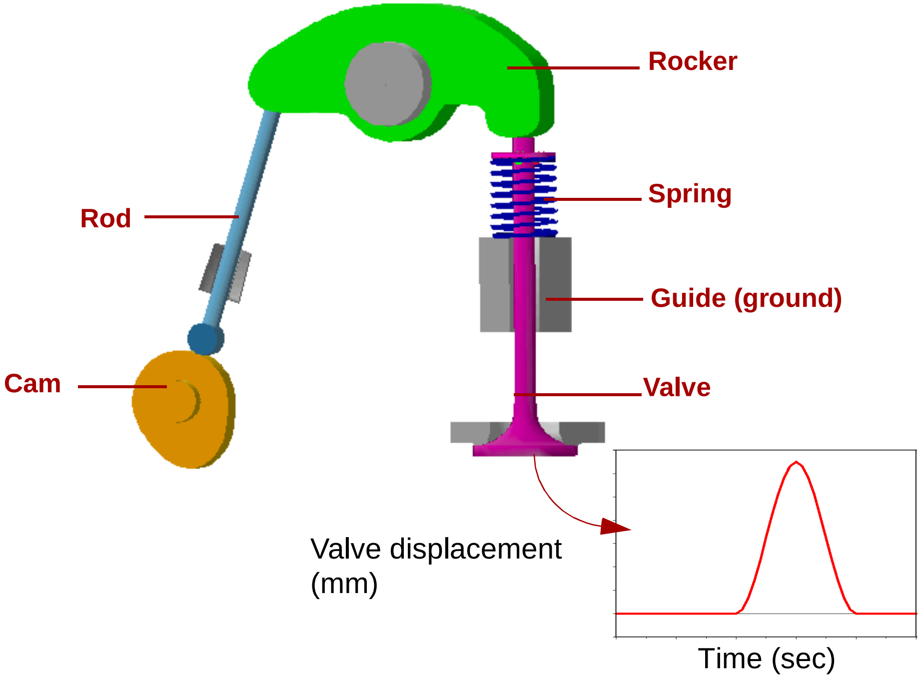
- Sphere-to-plane
- Circle-to-plane

The z-axis of the reference marker (normal to the plane) must point away from the plane and at the circle or sphere.

Workshop 20—Cam-Rocker-Valve

Problem statement

Design a cam profile based on desired valve displacement, and ensure that there is no follower liftoff when the cam is rotated at 3000 rpm.



Model description

- The model represents a valvetrain mechanism.
- The cam is being rotated at a velocity of 1 rotation per second.
- The rocker pivots about a pin attached to the engine block (ground).
- The valve displaces up and down as the rocker moves.
- When the valve moves, it lets small amounts of air in the chamber below it (not modeled here).

Start the workshop

1 Open ADAMS/View from the directory ***exercise_dir*/mod_20_camrocker.**

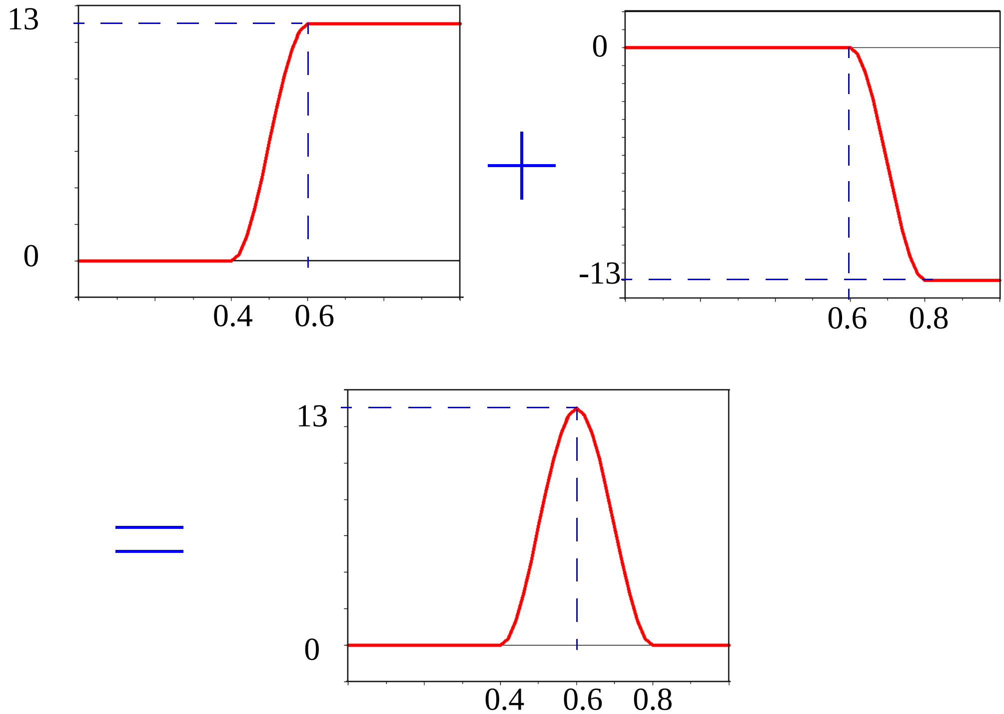
2 From the directory *exercise_dir*/mod_19_camrocker/valve_train_start.cmd, import the model command file **valve_train_start.cmd**.

The file contains a model named **valve_train**.

Apply motion

1 Apply joint motion to the valve-to-guide joint such that its displacement appears as seen in the following figure:

Add two step functions.



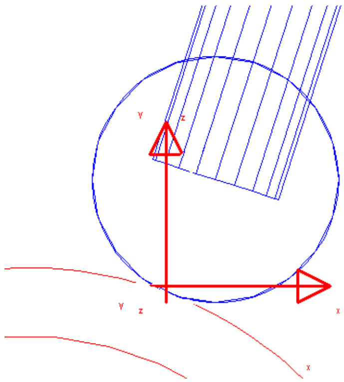
2 Run a simulation for **1** second, using **100** steps to verify that the valve displaces as a result of the joint motion.

Create a cam profile

Use a point trace to create a cam profile.

To use a point trace:

1. From the **Review** menu, select **Create Trace Spline.**
2. Select the circle of the **Rod** and then the **Cam**.
3. Verify that you now have a spline representing the cam profile.



4. Run a simulation to verify that the **Rod** appears to move along the surface of the **Cam**.

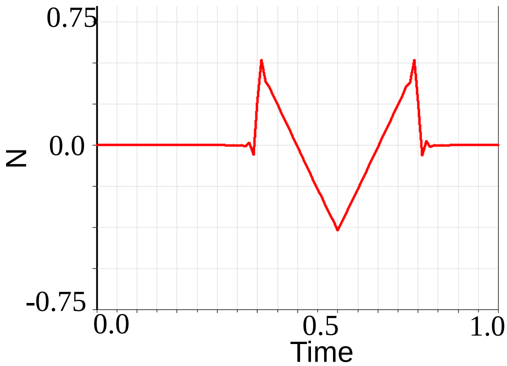
Constrain the rod to the cam

1. Delete the joint motion on the valve to ground joint.
2. Create a curve-on-curve constraint between the **Rod** and the **Cam**.
3. Run a simulation to verify that the new constraint works.

Measure the force in the curve-on-curve constraint

- Create a force measure for the curve-on-curve constraint. Measure the force along the z-axis of **ref_marker**, which belongs to the rod.

  The curve-on-curve constraint applies a negative force that keeps the rod follower on the cam, avoiding any liftoff.

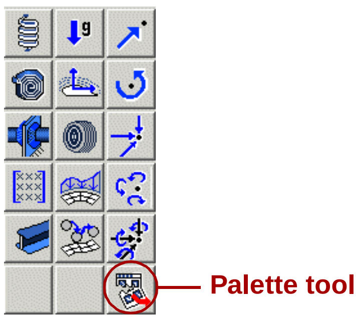


Make the cam-to-rod contact more realistic

Now you'll replace the curve-on-curve constraint with a circle-to-curve contact force.

To replace the curve-on-curve with circle-to-curve contact force:

1 From the Main Toolbox, right-click the **Create Forces** toolstack, and then select the **Display Palette** tool.



2 From the **Create Forces** palette, select the **General Force-Based Contact Force** tool .

3 Enter the following contact array values (right-click the **Contact Array** text box and point to **Create** to enter parameters):

- **Stiffness (K): 1e6 (N/mm)**
- **Force exponent (e): 1.5**
- **Damping (C): 10 (N-sec/mm)**
- **Penetration depth (d): $1e^{-3}$ mm**
- **Static coefficient of friction** (μ_s)**: 0.08**
- **Slip velocity (v_s): 1 (mm/sec)**
- **Dynamic coefficient of friction** (μ_d)**: 0.05**
- **Transition velocity** (v_t)**: 2 (mm/sec)**

4 Run a simulation to check if liftoff occurs.

Prevent liftoff using a spring-damper

1 Add a spring-damper between the plate near the top of the valve and the top of the guide (which is part of ground) using the given construction points and the following parameters:

- **Stiffness (K): 20 (N/mm)**
- **Damping (C): 0.002 (N-sec/mm)**
- **Length at preload: design position**
- **Preload: 100 N**

2 Find the static equilibrium of the model.

Do not reset the model before going on to the next step.

3 Run a dynamic simulation to view the effects of the spring starting from static equilibrium.

4 Modify rotational motion on the cam to a speed of **3000 rpm**.

5 Modify the spring-damper characteristics (stiffness, damping, and preload) to prevent liftoff based on the new rotational speed of the cam.

6 To check for liftoff, measure the contact force.

Save your work

1 Save the model.

2 Exit ADAMS/View.

Optional tasks

Change the shape of the follower:

1 Delete the circle-to-curve contact force between the **Rod** circle and the **Cam** curve.

2 Add a spline to the **Rod** that is a shape other than a circle and use that as the follower.

Change the grid spacing to **5mm** in the **x** and **y** directions and draw the new follower geometry on the grid.

3 Recreate the contact force between the **Rod** and the **Cam** using the new spline as the follower.

Use a curve-to-curve contact force.

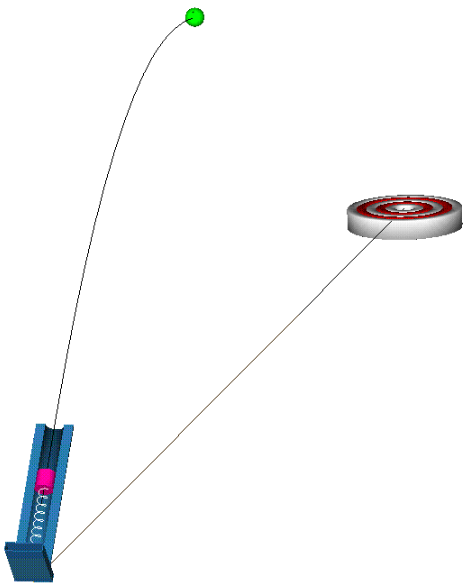
Module review

1 How many DOF are removed by adding a curve-on-curve constraint?

2 How many DOF are removed by a curve-to-curve force?

21 TARGET PRACTICE I

Complete the construction of a parametric gun.



What's in this module:

Multi-Component Forces

Types of multi-component forces

- Vector force (three translational components)
- Vector torque (three rotational components)
- General force vector (three translational, three rotational components)

Characteristics of vector force

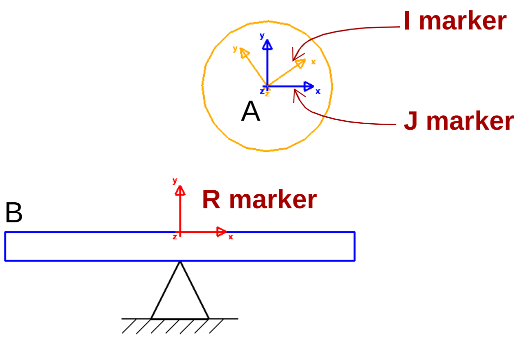
The characteristic:	Defines:
Bodies	Two
Points of application	Two (action force at *I marker* and reaction force at floating *J marker*)
Vector components	Three translational
Orientations	Based on reference marker (*R marker*)
Magnitudes (F_x, F_y, F_z)	User-defined

Notes:

- The floating J marker always maintains the same location as the I marker.
- The characteristics of other multi-component forces conceptually work the same way.

Example of a force vector

- A vector force representing a contact between a ball and a cantilever:



I marker belongs to part A

J marker belongs to part B but floats its location with the I marker

R marker belongs to part B

- Because the J marker belongs to part B, the force acts on part B when the bodies collide.
- Because the J marker moves with the I marker, part B knows where to apply the reaction force.

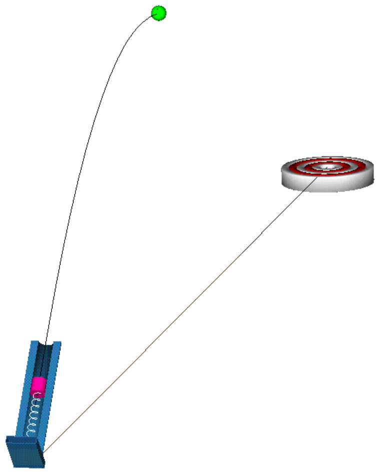
Note: In the example, the J and R markers must belong to the same part. However, the R marker can belong to *any* part.

See also: Characteristics of a multi-component force on page 321

Workshop 21—Target Practice I

Problem statement

Complete the construction of a parametric gun.



Model description

- Currently, the model has all the geometry, parts, and constraints it needs.
- There is a multi-component force between the marble and the launch pad that represents the contact between those two objects (including friction).
- There is a single component force representing the spring force between the hammer and the launch pad.
- The single-component force is designed so that it changes characteristics depending on the type of simulation being run (static versus dynamic):
    - If a static simulation is run, the spring has a free length of 40 mm.
    - If a dynamic simulation is run, the spring has a free length of 100 mm.
- Therefore, initially run a static simulation, so the marble falls on the hammer and compresses the spring a little. Then when you run a dynamic simulation, the spring thinks it is compressed a great deal, and shoots the marble.
- The model is already parameterized with variables describing the elevation angle of the launch pad and the stiffness and damping of the spring.
- You will only modify the stiffness of the spring.
- Initially, the spring stiffness is 20 N/mm.

Start the workshop

Import the file to build the model target_practice.

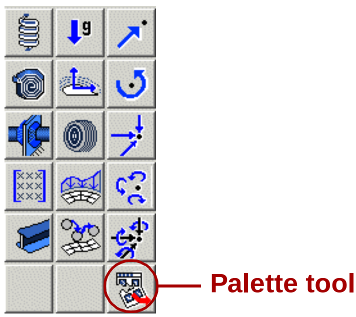
To start the workshop:

1. Start ADAMS/View from the directory ***exercise_dir*/mod_21_target_practice_1**.
2. Import the model command file **target_practice_start.cmd**.

Create a contact between the hammer and the marble:

1. Run a simulation to see the forces that affect the model in its current configuration.
2. Create a marker on the hammer to represent the plane in the sphere-to-plane contact force that you will create next.

1. From the Main Toolbox, right-click the **Create Forces** toolstack, and then select the **Display Palette** tool.

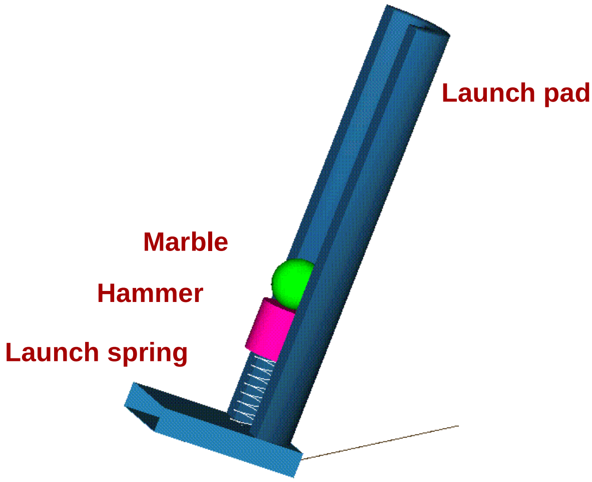


2. From the **Create Forces** palette, select the **General Force-Based Contact Force** tool .

3. Create a sphere-to-plane contact force that represents the contact between the marble and the hammer, using the following parameters (right-click the **Contact Array** text box and point to **Create** to enter parameters):

   - **k: 100 (N/mm)**
   - **e: 1.01**
   - **c: 1.0 (N-sec/mm)**
   - **d: 0.2 mm**

The marble is not initially in contact with the hammer; therefore, you must precede each dynamic simulation that you submit by a static equilibrium simulation.



4 Using ADAMS/Solver commands, create a script that performs a static equilibrium simulation followed by a dynamic simulation.

The script will make running simulations easier and will be needed for the design study later.

5 Create a measure of the marble's global x displacement.

Simulate the model

1 Run a simulation.

2 Save simulation results.

Include aerodynamic drag force

1 Create a multi-component force between the **marble** and **ground** whose directions are aligned with the global. Use the following parameters to describe global x and y components of the drag force:

- $F_x = -1/2*\rho*V_x*V_m*C_d*A$
- $F_y = -1/2*\rho*V_y*V_m*C_d*A$

where:

- ρ = 1.3e-9 kg/mm3 = density of air
- V_x = global x component of the marble's velocity
- V_y = global y component of the marble's velocity
- V_m = magnitude of the marble's velocity
- C_d = 0.45 = drag coefficient
- $A = \pi r^2$ = two-dimensional area of the marble face

2 Compare the x displacement of the marble for each set of simulation results (with and without aerodynamic forces).

Save your work

1 Save your model.

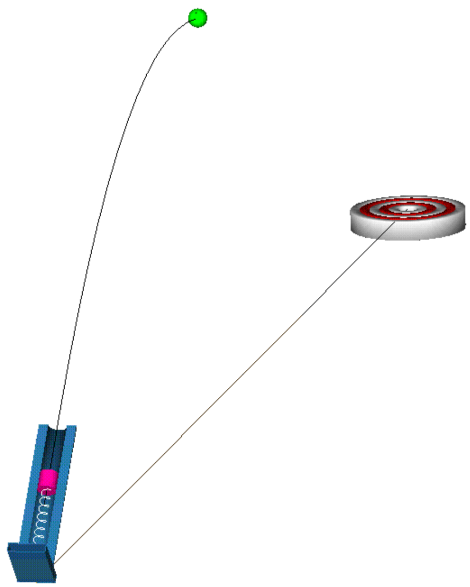
2 Exit ADAMS/View.

Module review

1 What defines the force directions (F_x, F_y and F_z) in a force vector?

22 Target Practice II

Find the launch-spring stiffness for the given model of a parametric gun that will allow the marble to hit the target.



What's in this module:

Sensors

- Monitor any quantity of interest in a model during a simulation, and take a specified action when the quantity reaches or exceeds a critical value.
- Take one of the following actions:
  - Completely stop the simulation.
  - If used with a script, sensors halt the current simulation and continue with the next command in the script.
- A sensor basically represents an *If/Then* statement:

  If quantity = value (+/- tolerance)

  Then take a specified action

Example of using sensors with scripts

- Monitor the reaction force in a constraint and deactivate the constraint when the force exceeds a specified value.
- Monitor the distance between two objects and reduce the solution step size just before contact, to avoid convergence problems.

Design variables

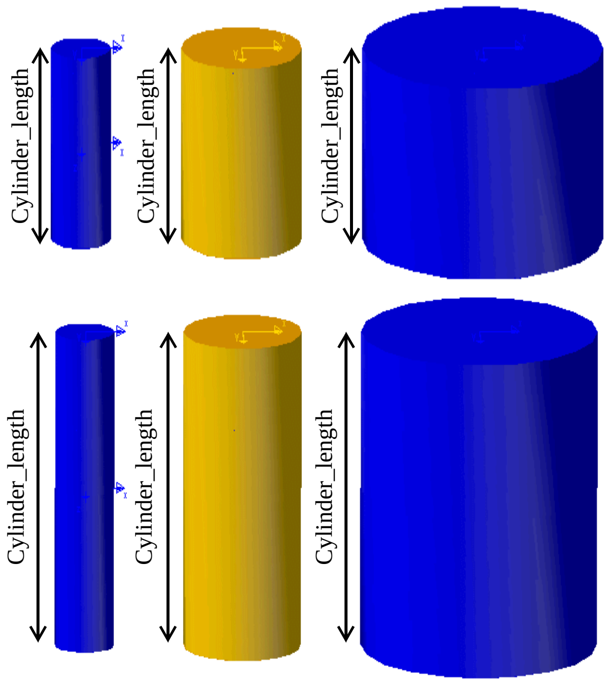
- Define independent parameters that can be tied to objects.
- Organize the critical parameters of the design into a concise list of values that can be easily reviewed and modified.

Example

You can create a design variable called cylinder_length to control the lengths of all three cylinders as shown next:

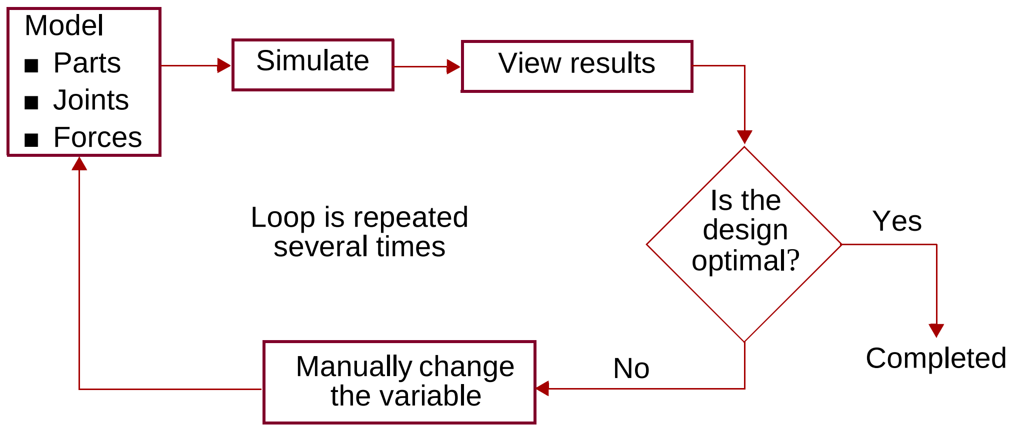
Cylinder_length = 150

Cylinder_length = 300

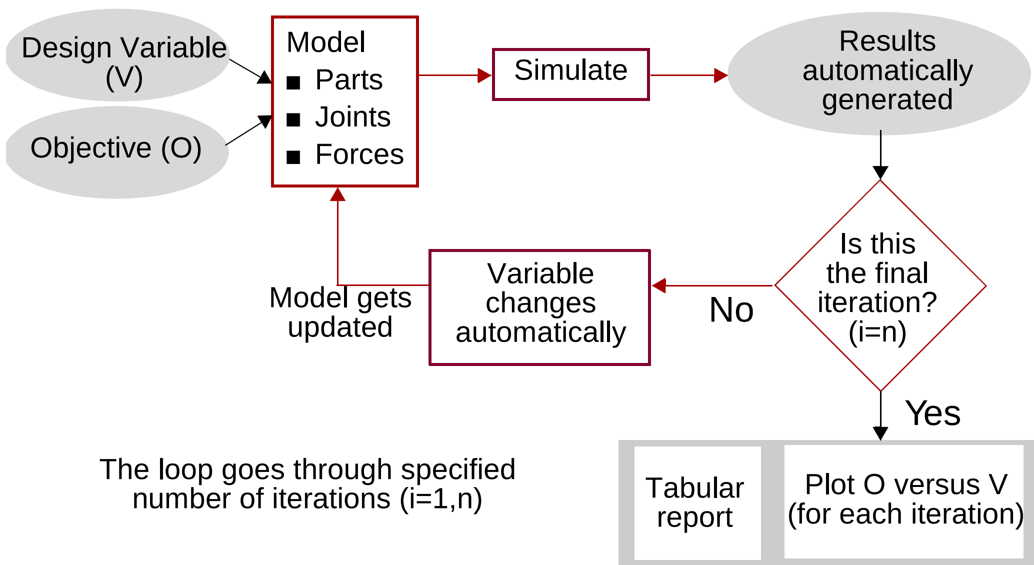


Note: You can also use parametric analyses to automatically run a series of simulations that vary your design variables.

Trial and error method (manual iterations)



Design study method (automated iterations)



Definition of a design study

- Varies a single design variable (V) across a range of values.
- Runs a simulation at each value.
- Reports the performance measure for each simulation.

From the results generated, you can determine

- The best value for V among the values simulated.
- The approximate design sensitivity of V (rate of change of performance measure with respect to V).

Sensitivity, S, at iteration, *i*

$$S_i = \frac{1}{2}\left(\frac{O_{i+1} - O_i}{V_{i+1} - V_i} + \frac{O_i - O_{i-1}}{V_i - V_{i-1}}\right)$$

- Looking at Trial 4 (i = 4):

$$S_4 = \frac{1}{2}\left(\frac{(-0.62784) - 0.017103}{10.7 - 10.6} + \frac{(-0.017103) - 0.58166}{10.6 - 10.5}\right)$$

S_4 = -6.0475

- S_4 is the approximate slope at Trial 4 (tip_y_loc=10.6) in the plot.

Design Study Summary

Model Name : stamp
Date Run : 15:48:55 23-Dec-98

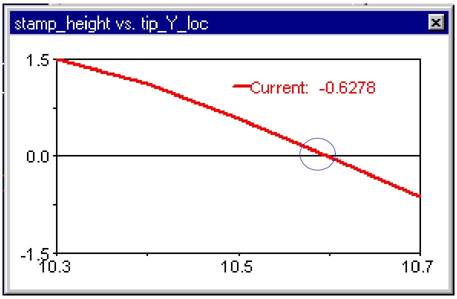
Objectives

O1) Minimum of stamp_height
Units : inch
Maximum Value: 1.48945 (trial 1)
Minimum Value: -0.627838 (trial 5)

Design Variables

V1) tip_Y_loc
Units : inch

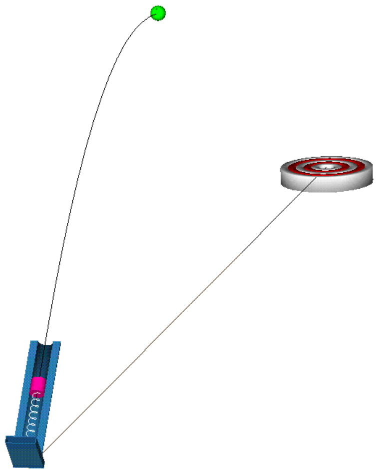
Trial	stamp_height	tip_Y_loc	Sensitivity
1	1.4894	10.300	-3.6131
2	1.1281	10.400	-4.5389
3	0.58166	10.500	-5.7262
4	-0.017103	10.600	-6.0475
5	-0.62784	10.700	-6.1073



Workshop 22—Target Practice II

Problem statement

Find the launch spring stiffness for the given model of a parametric gun that will allow the marble to hit the target.



Model description

In this workshop, you use the model you created in Target Practice I on page 268.

Start the workshop

Import the file to build the model target_practice.

To start the workshop:

1 Start ADAMS/View from the directory ***exercise_dir*/mod_22_target_practice_2**.

2 From the directory ***exercise_dir*/mod_21_target_practice_1**, import the model that you created in the previous module.

If you need a copy of the model, import the command file target_practice_1_completed.cmd from the directory *exercise_dir/* mod_21_target_practice_1/completed.

Track when the simulation is complete

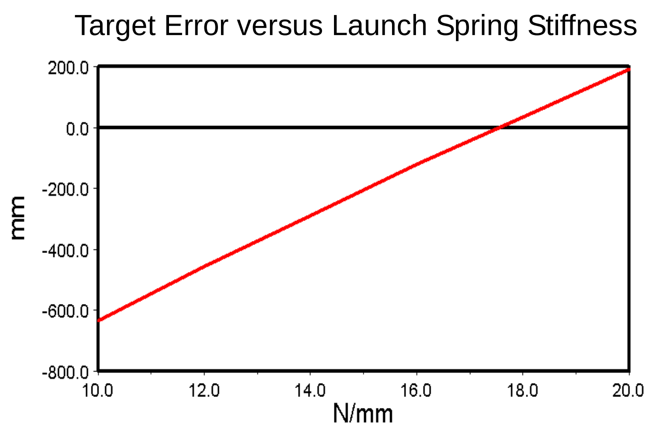
1. Create a measure that tracks the global y displacement of the center of the marble (**.target_practice.marble.cm**) with respect to the ground plane (**.target_practice.ground.GCS**).
2. Create a sensor to determine when the ball passes the global xz plane.

   Have the sensor monitor the global y displacement measure created earlier, and when it is less than or equal to 0.0 mm, direct the sensor to:

   - Generate an additional output step.
   - Terminate the current simulation step and stop the simulation script.

Set up a design study

1. Create a measure of the global x distance from the target center (**.ground.target**) to the marble center (**.marble._cm**) and name it **target_error**.
2. Run a design study that gives the last value of **target_error** over six default levels of the existing design variable, **launch_spr_stiffness**.



3. Note the approximate value of stiffness at which the marble hits the target.

   Use it to answer Question 1 in Module review on page 281.

Save your work

1. Save your model.
2. Exit ADAMS/View.

Workshop 22—Target Practice II...

Optional tasks

Modify design studies:

1 Run a design study where you vary **launch_spr_damping** from **0.02** to **0.10 Nsec/mm**.

2 Run a design study where you vary launch pad **elevation_angle** from **10** to **60 degrees**.

Module review

1 What is the approximate value of stiffness at which the marble hits the target?

23 Recommended Practices

What are the recommended practices in debugging a model?

What's in this module:

Crawl-walk-run

- Try to understand the mechanism from a physical standpoint.
- Use building blocks of concepts that have worked in the past.
- Add enhancements to the model while testing periodically.
- Build kinematic models before building dynamic models.
- Use motions to check models before applying forces.
- Use motions which start with zero velocity.
- Verify enhancements to a complex model on a simpler model first.

Geometry associativity errors

Geometry may be added to the wrong part.

Mass properties

- Using imported CAD-created geometry (IGES, STL, and so on) can yield inaccurate mass properties.
- Ensure inertia matrix is realistic.
- Use aggregate mass for a quick check of system mass and inertia.
- Use the Table Editor to do a quick check and potentially fix individual part masses and inertia.
- Small part mass and inertia lead to unrealistically high frequencies.

Initial velocities

Check to see that part initial velocities are consistent (look in the .out file).

Dummy parts

- Whenever possible, avoid using them.
- If absolutely needed, constrain all six DOF and assign a mass of 0.0 (not 1e-20).

Design configuration

- Build a model close to assembled position.
- Build a model close to a stable equilibrium position, if possible.

Fixed joints

- Not needed, since two or more parts can be combined or merged into a single part.
- An extra part with a fixed joint adds unnecessary equations to your system.
- When locking a part to ground, enormous torque may develop due to large moment arms.
- If absolutely needed, then add fixed joints at the center-of-mass (cm) location of lightest part.
- If locking a part to ground, consider assigning a very large mass/inertia to it so it can behave like ground.

Note: Whenever possible, avoid using fixed joints.

Universal joints

When a universal joint is at 90^{o}, you get a singular matrix.

Motion

- Motion elements should only be functions of time.
- Do not use motion (functions) as a function of variables (or states).

Note: Avoid redundant constraints.

Modeling Practices: Compliant Connections

Spring-dampers

- Ensure that the marker endpoints (DM(I,J)) are never superimposed.
- Watch out for springs with very stiff spring constants.
- Watch out for springs with no damping.

Bushings

Watch out for bushings with large rotations.

Modeling Practices: Run-time Functions

Function Builder

- Assists in building functions.
- Assists in function verification.
- Has function plot capability.

Velocity

Make sure velocities are correct in force expressions. For example, in the damping function: -c*VX(i, j, j, _), the fourth term is missing.

Splines

- Approximate forces with smooth, continuous splines.
- Extend the range of spline data beyond the range of need.
- Cubic splines (CUBSPL) work better on motions than Akima.
- Akima splines (AKISPL) work better on forces than Cubic.
- The Akima interpolation method is faster and can be defined as a surface, but its derivatives are generally discontinuous.

IMPACTs/BISTOPs

- Do not use 1.0 for exponent on IMPACT or BISTOP functions.
- Models with IMPACTs/BISTOPs should have slight penetration in design position when doing statics.

Measures

- Set up measures of your run-time functions.
- Set up measures of components of your run-time functions.
- Ensure that your function will not try to divide by zero.

Modeling Practices: Run-time Functions

Contacts

- Do not use 1.0 for exponent on IMPACT or BISTOP functions.
- Models with contacts should have slight penetration in design position when doing statics.

Tires

- Models with tires should have slight penetration in model position when doing statics.
- If only rear tires penetrate, the static position could be a “handstand.”

Units

- Use consistent units throughout the model (time, mass, stiffness, damping, and so on).
- Choose units (mass, force, time, and so on) that do not require using very large or very small numbers.
- Be wary when your model contains numbers like 1e+23 or 1e-20.
- Use appropriate units—when modeling large models such as an aircraft landing on a runway, length units of millimeters may not be appropriate. Conversely, when modeling small models such as a power window switch (made up of small moving parts), using length units of meters may not be appropriate.
- Use reasonable time units—high frequencies may be better modeled with time units of milliseconds rather than seconds.

Gravity

- Check magnitude and direction.
- Check for multiple gravity elements.

Debugging Tips

Model verify

- Lists number of moving parts, number of each type of constraint.
- Lists Gruebler's count and actual DOF count.
- Lists redundant constraints.
- Reports misaligned forces/force elements, joints, and so on.
- Helps identify and eliminate causes for input warning (don't ignore).

Model topology

- Text or graphical model topology.
- Table Editor provides spreadsheet-like overview of model content.

Icon feedback

Broken icon in design configuration probably means incorrectly defined joint or force.

Table Editor

Convenient way to inspect and modify models (particularly large ones).

Interactive simulation

By default, is turned on.

Model display update

As ADAMS performs the simulation, you have the option to get immediate graphical feedback of the simulation at every:

- Output step
- Integration step
- Iteration

Icons visible during simulation

This may help you monitor behavior of model components.

Subroutines

- Check for their existence.
- While debugging a model, eliminate user subroutines so that they are not the source of the error.

Gravity

Turning gravity off can accentuate modeling errors and make debugging easier.

Statics

- When applicable, perform an initial static first.
- If static solution fails:
  - Turn on **Model display update = at every iteration** to provide additional insight.
  - Identify and eliminate the undesired static configuration—there could be more than one static configuration and ADAMS could be finding the undesired one.
- Check to see if there are any floating parts.
- Check the signs of applied forces.
- Experiment with Alimit/Tlimit/Maxit/Stability.
- Check if impacts are initially in contact; if not, they should be.
- Running an initial dynamic simulation can help you determine why the model is not finding static equilibrium.

Dynamics

- If integrator fails to **start-up**:
  - Check sign and magnitude of forces.
  - Look at accelerations to understand what is happening.
  - Perform initial static analysis first.
  - Try a quasi-static simulation.
  - Try changing integrator parameter - HINIT.
  - Try a different integrator.
- If integrator fails in the **middle of a simulation**:
  - Look at animation and plots until failure, to understand simulation.
  - Decrease integrator parameter - HMAX.
  - Do not let the integrator step over important events.
  - Short duration events, such as an impulse can be captured by setting the maximum time step, HMAX, to a value less than the impulse width.
  - Use HMAX so ADAMS/Solver acts as a fixed-step integrator
  - Decrease error.
  - Try a different integrator.
- If integrator takes **very small steps**:
  - Look for sudden changes in force and motion input.
  - Rescale model to get more uniform numbers.

Velocities at time=0

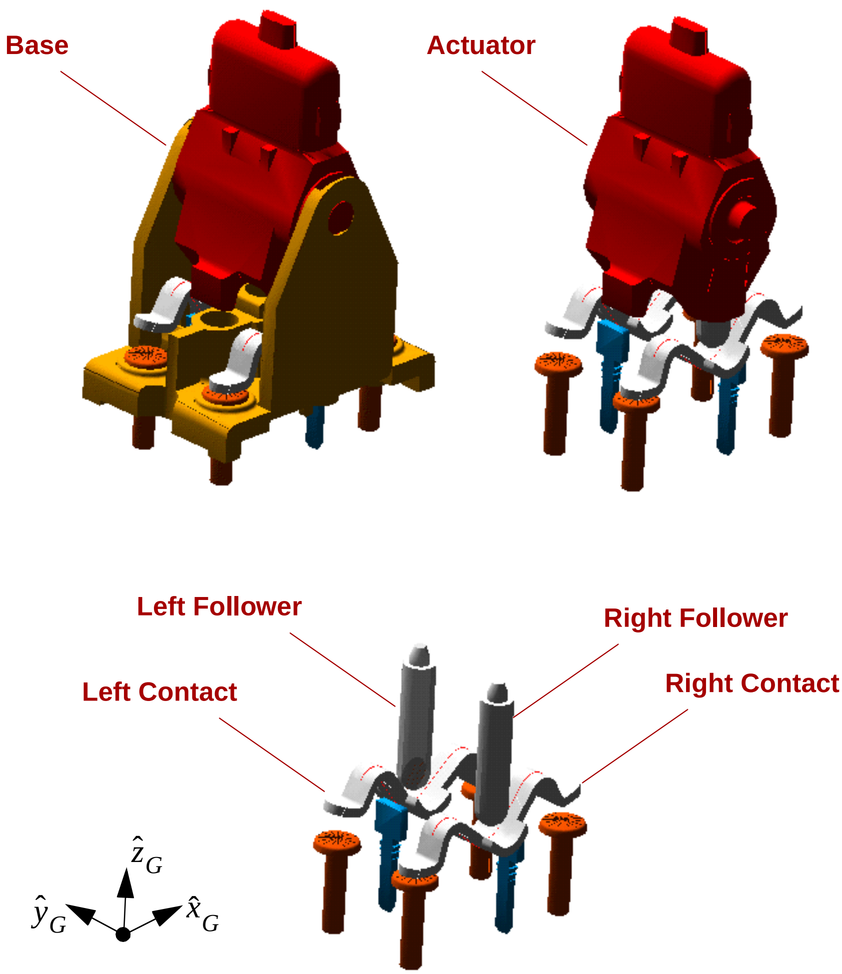
Check initial velocities using the .out file.

24 Switch Mechanism Workshop

Problem statement

Determine the minimum force necessary to toggle the switch mechanism to the forward and rearward directions.

Switch Mechanism



Model description

The given switch model contains parts with shell geometry and mass properties:

Table 1. Parts

Part name:	Geometry (shells):	Mass (grams):
base (ground)	base_main_geo	(ground)
	base_right_front_geo	
	base_right_mid_geo	
	base_right_rear_geo	
	base_left_front_geo	
	base_left_mid_geo	
	base_left_rear_geo	
actuator	actuator_geo	1.52
right_contact	right_contact_geo	0.51
left_contact	left_contact_geo	0.51
right_follower	right_follower_geo	0.1076
left_follower	left_follower_geo	0.1076

The switch model contains construction points for adding the necessary modeling elements to address the problem statement. They are:

Table 2. Construction Points

Point:	Description:
POINT_1	Actuator to base pivot location
POINT_2	right_follower to actuator spring lower location
POINT_3	left_follower to actuator spring lower location
POINT_4	right_follower to actuator spring upper location
POINT_5	left_follower to actuator spring upper location

Table 2. Construction Points *(continued)*

Point:	Description:
POINT_6	Contains z-coordinate of base contact plane with left_contact and right_contact at four corners
POINT_7	left_contact to base idealized constraint location
POINT_8	right_contact to base idealized constraint location
POINT_9	Location of base front contact surface with actuator
POINT_10	Location of base rear contact surface with actuator
POINT_11	Location of actuator front contact point with base
POINT_12	Location of actuator rear contact point with base
POINT_13	Location of right_contact to base mid-contact point
POINT_14	Location of left_contact to base mid-contact point
POINT_15	Location of force application

The switch model is mounted such that the model’s global negative x-axis defines forward, positive z-axis defines up, and y-axis defines left (see problem statement).

Section I: Test the right half with constraints only

You can think of the switch mechanism as consisting of two halves. In this exercise, first constrain the right half of the mechanism and perform a kinematic simulation to visually verify correct motion. Then, add spring and contact forces to the right half to ensure that the mechanism actually toggles. Then, add more detail to the right half, introduce the left half, and then finally perform a system-level simulation.

This section emphasizes the crawl-walk-run method. In this section, you will crawl!

To import the model:

1. Start ADAMS/View from the directory ***exercise_dir*/switch_workshop**.
2. From the directory ***exercise_dir*/switch_workshop/switch_start.cmd**, import the model command file **switch_start.cmd.**

   This file contains commands to build a model named **switch**.

To set up modeling environment:

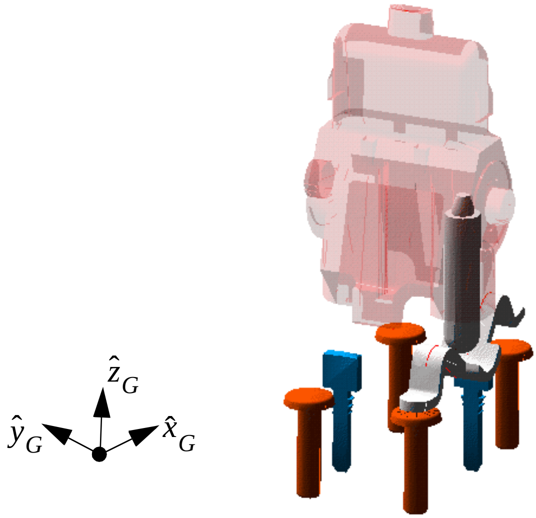
1. Turn the working grid off.
2. To modify the preset views in ADAMS/View so that they are relevant to the global coordinate system, from the **Tools** menu, select **Command Navigator**.

   The Command Navigator appears.
3. In the Command Navigator, from the **View** menu, select **Management**, and then select **Orient**.

   See Model description, on page 296 for an explanation of how the global axes are defined.
4. Preset the front view. All other views will be defined based on this front view. Ensure that the preset views (front, top, and so on) are correctly defined based on the global coordinate system.
5. To set the transparency of the actuator part to 80%, right-click the actuator, and then select **Appearance**.
6. Turn off the visibility of the base part geometry, **base_main_geo**.
7. Deactivate the **left_contact** and **left_follower** parts.
8. Turn off the visibility of the **left_contact** and **left_follower** parts.

9 Set gravity in the global negative-z direction.

The model view should look as follows:

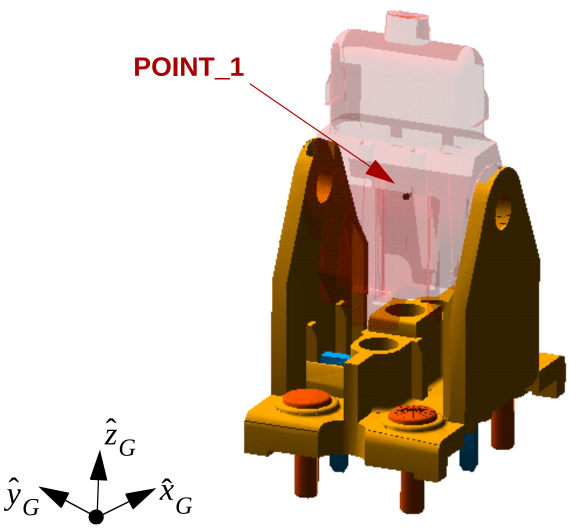


To add constraints:

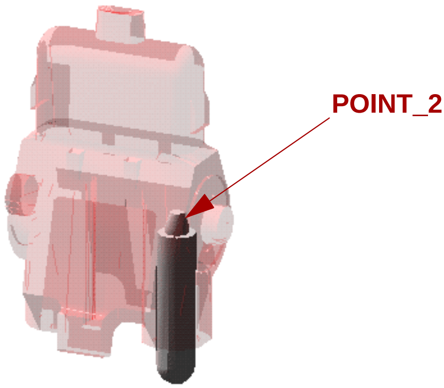
1 Display icons.

2 Constrain the actuator to the **base** at **POINT_1** such that the only relative allowable degree of freedom is rotation about $\hat{y}_G$.

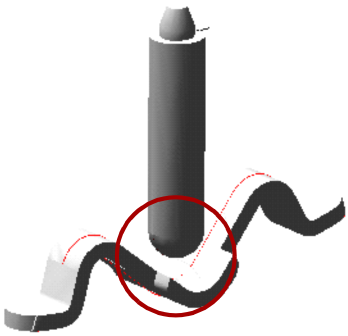
Establish a reference marker with global orientation on the base (ground) part that makes picking of global direction vectors easy. Setting the color and size of the marker aids in referencing it later.



3 Constrain the **right_follower** to the **actuator** at **POINT_2** such that the only relative allowable degree of freedom is translation along $\hat{z}_G$.

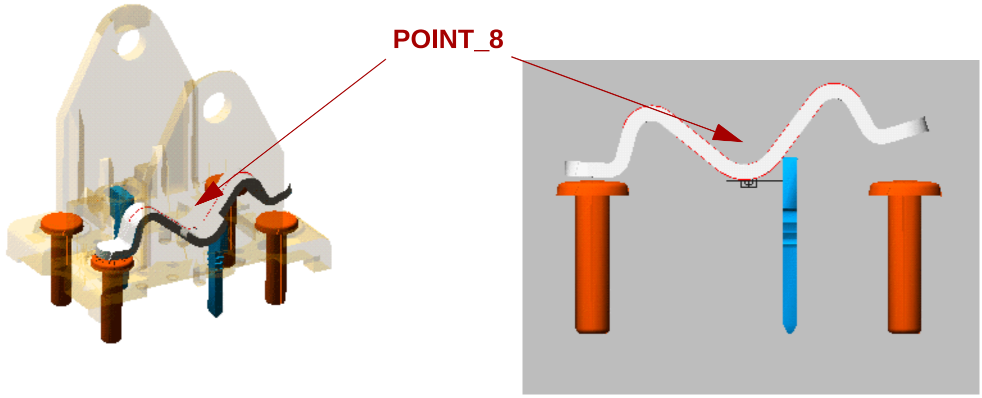


4 Constrain the tip of the **right_follower** to the upper curve on the **right_contact** part.



When creating the curve-to-curve constraint, select the red circle at the tip of the right_follower part, parallel to the global-xz plane.

5 Constrain the **right_contact** part to the base at **POINT_8** such that the only relative allowable degree of freedom is translation along $\hat{y}_G$.



This might not seem intuitive, but it ensures that there are no redundant constraints in the model. It is a good modeling practice to remove all redundant constraints in your system prior to performing a simulation.

6 Add displacement **joint motion** to the actuator-to-base revolute joint such that the actuator oscillates sinusoidally with an amplitude of 15.1 degrees and one cycle per second.

Test the model:

1 Verify the model.

Your system should have 0 degrees of freedom and no redundant constraints at this configuration. If it does not, inspect the model to determine the discrepancy.

2 Simulate the model kinematically to visually verify correct motion, using an end time of 1 second with 100 output steps.

3 Save your work.

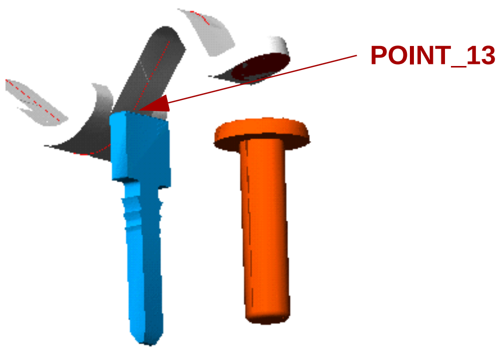
Section II: Test the right half with front and rear contacts

Change the constraints on the right_contact part so that it can rotate and make contact with the right front and rear terminals on the base part (it will rock back and forth like a see-saw). Use the curve-to-curve constraint created earlier.

In this section you will start to walk!

To add detail to the connections between the right_contact and base parts:

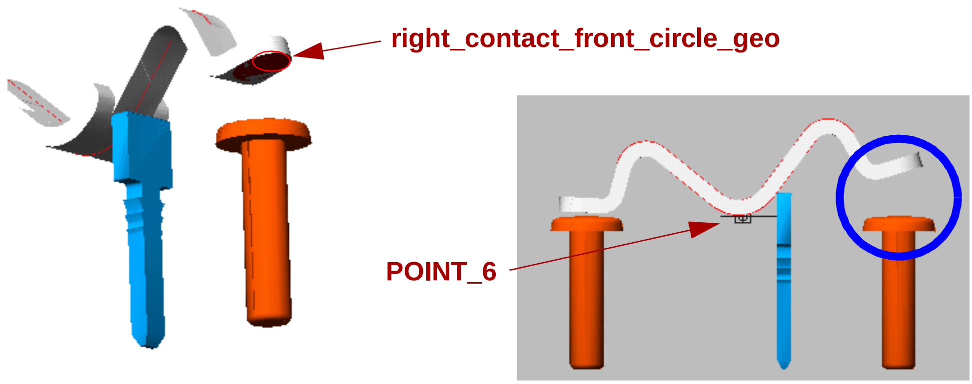
1. Remove the translational joint constraining the **right_contact** to the base at **POINT_8**.
2. Constrain the **right_contact** to the base at **POINT_13** such that the only allowable degree of freedom is rotation about $\hat{y}_G$.



3 Create a circle-to-plane contact force between the front end of the **right_contact** part and the front right corner of the **base** part. Use a plane that is parallel to the global xy-plane at **POINT_6**.

First, create a marker on the base part at POINT_6 whose xy-plane defines the plane of contact and z-axis defines the side of the plane used by the contact force. When creating the marker, try using the Z-Axis orientation option. This marker will be used as the plane marker in the circle-to-plane contact force.

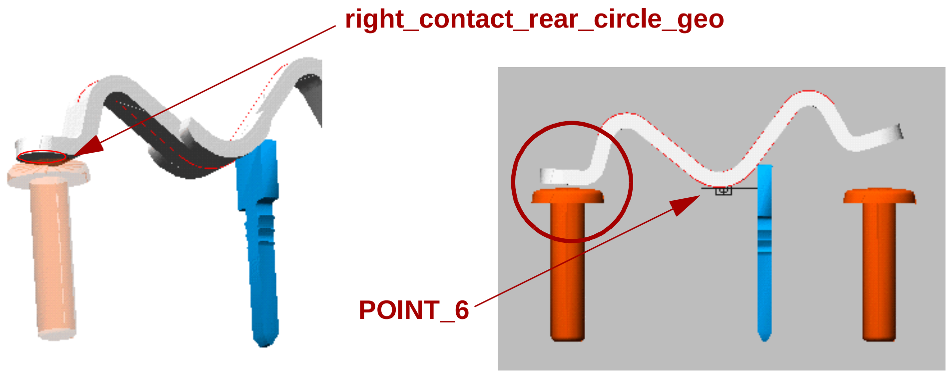
We recommend that you rename the plane marker used in this circle-to-plane force, as you will create three other forces of this type which all use the same contact plane.



The contact parameter should be:

- **Stiffness: 1e8 (milliNewton/mm)**
- **Force exponent: 2.2**
- **Damping: 1e1 (milliNewton-sec/mm)**
- **Penetration depth: 1e-3 mm**
- **Static friction: off**
- **Dynamic friction: off**

4 Create a circle-to-plane contact force between the rear end of the right_contact part and rear right corner of the base part. Use a plane that is parallel to the global xy-plane at POINT_6 and the same contact parameters used in Step 3.

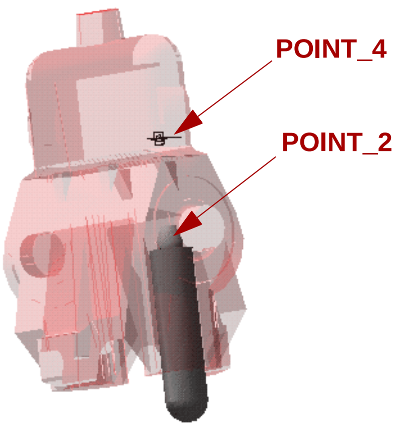


Use the same plane marker and contact array used in Step 3.

To add spring force to the right half:

1 Create a spring between the **right_follower** at **POINT_2** and actuator at **POINT_4** using the following parameters:

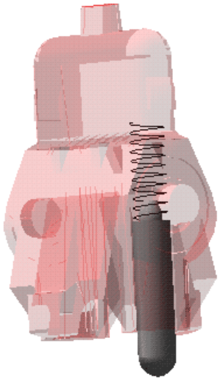
- **Stiffness: 852 (milliNewton/mm)**
- **Damping: 0.1 (milliNewton-sec/mm)**
- **Free length: 9 mm**



You need markers to create the spring. First create markers for each endpoint belonging to the appropriate parts.

2 Override default spring geometry by using these custom parameters:

- **Coil count: 10**
- **Diameter of spring: 2.5 mm**
- **Damper diameter at ij: 0, 0**
- **Tip length at ij: 0, 0**
- **Cup length at ij: 0, 0**



With nothing selected, from the **Edit** menu, select **Modify**.

Filter on geometry, then double-click **SPRING_1**, then select **spring_graphic** (not damper_graphic). To make it stand out, change the color to white.

To test the model:

1 Verify the model.

The system should now have one degree of freedom and one redundant constraint.

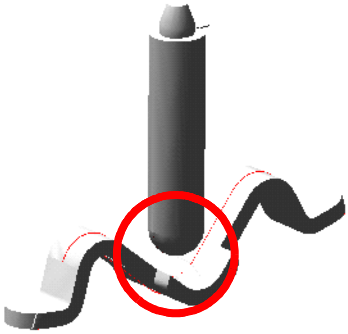
At this time, does the redundant constraint affect what you are doing?

2 Simulate the model to visually verify correct motion.

Perform an initial static simulation, immediately followed by a 1-second, 200-step dynamic simulation.

To replace the right_follower to right_contact curve-to-curve constraint with a force:

1. Remove the **curve-to-curve constraint** between the tip of the **right_follower** and the upper curve on the **right_contact** part.
2. Create a **circle-to-curve contact force** between the tip of the **right_follower** and the upper curve on the **right_contact** part. Use the same contact array used in Step 3.



   Verify that the arrow displayed when the curve is selected indicates the side of the curve used by the contact force.

To test the model:

1. Verify the model.

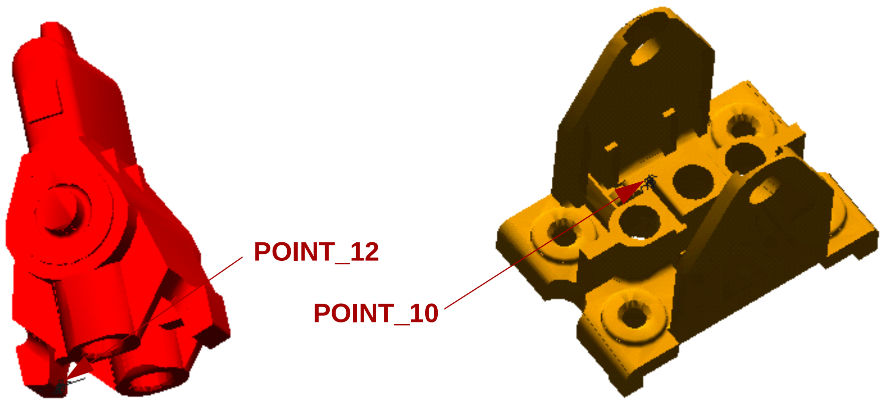
   Your system should have two degrees of freedom and no redundant constraints.
2. Simulate the model to visually verify correct motion.

   Perform an initial static simulation, immediately followed by a 1-second, 200-step dynamic simulation.

To create stops (forces) between the actuator and base:

1 Create a **sphere-to-plane contact force** between the rear end of the **actuator** and the **base** part. As the actuator rotates, its sphere strikes a surface parallel to the global-yz plane on the base. Use the following parameters:

- **Sphere:** sphere on the actuator part at POINT_12 with a radius of 0.1 mm
- **Plane:** parallel to global yz-plane at POINT_10
- **Contact parameters:** same as in Step 3



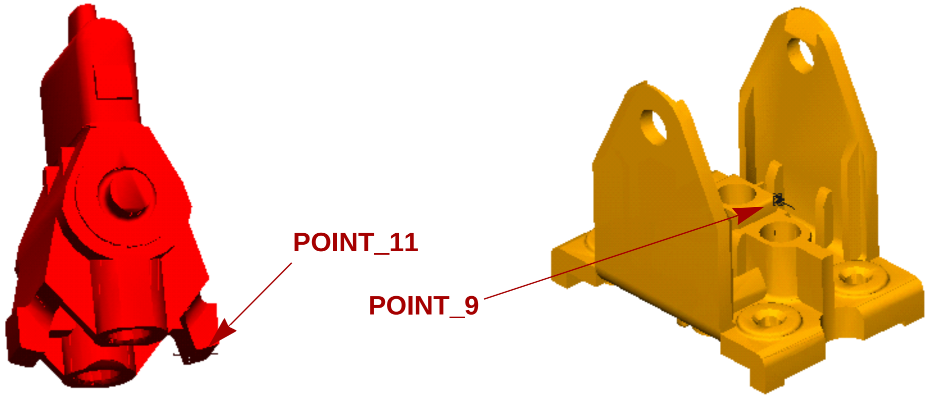
First create a sphere on the actuator part at POINT_12. Use this sphere in the sphere-to-plane contact force.

Then create a marker on the base part at POINT_10 whose xy-plane defines the plane of contact, and whose z-axis defines the side of the plane used by the contact force. Use this marker as the plane marker in the sphere-to-plane contact force.

Modify the contact force and turn on the graphic force display for the first body (.actuator).

2 Create a sphere-to-plane contact force between the front end of the actuator and the base part. Use the following parameters:

- **Sphere:** sphere on the actuator part at POINT_11 with a radius of 0.1 mm
- **Plane:** parallel to global yz-plane at POINT_9
- **Contact parameters:** same as in Step 3



First create a sphere on the **actuator** part at **POINT_11**. Use this sphere in the sphere-to-plane contact force.

Then, create a **marker** on the base part at **POINT_9** whose xy-plane defines the plane of contact and whose z-axis defines the side of the plane used by the contact force. This marker will be used as the plane marker in the sphere-to-plane contact force.

Turn on the force display for the first body (.actuator) again.

To test the model using the actuator motion input:

1 Verify the model.

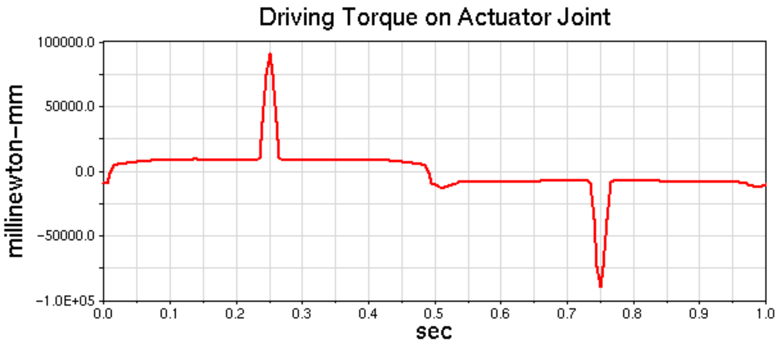
Your system should have two degrees of freedom and no redundant constraints.

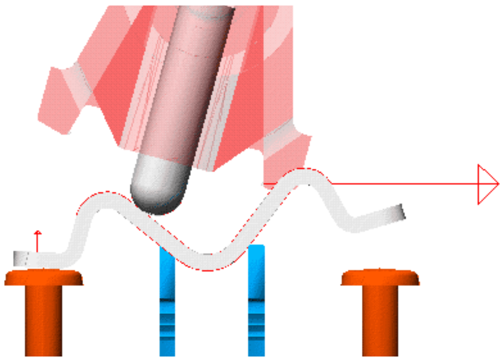
2 Simulate the model to visually verify correct motion.

Perform an initial static simulation, immediately followed by a 1-second, 200-step dynamic simulation.

At this time, you should see force graphics representing the stops at the actuator’s extreme angular displacement configurations. If these stop forces are not returning a non-zero value, further inspect the model.

3 In ADAMS/PostProcessor, plot the torque at the actuator's revolute joint (due to the motion input) versus time. On the same page, animate the model. Can you explain the shape of the curve? Is this intuitive?

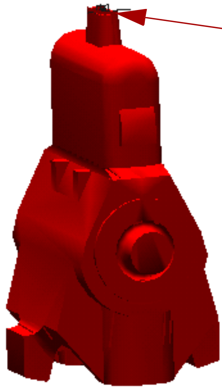




To create force application:

1. Remove the **motion** applied to the revolute joint constraining the **actuator** to the **base**.
2. Apply a **force** to the actuator part at **POINT_15** in the positive $\hat{x}_G$ direction, moving with the body. Use the following function:

   f(t) = -100*time



To test the model:

1. Verify the model.

   Your system should have three degrees of freedom and no redundant constraints.
2. Create a measure based on the force magnitude of the right front contact force between the right_contact part and the base part.
3. Create a sensor that triggers when the force magnitude of the right front contact force (measured in the above step) is greater than or equal to 1mN within a tolerance of 1e-3 mN.

   When sensed, ADAMS/Solver should terminate the current simulation step and continue the simulation script.

   Use the Function Builder to assist in referencing the expression you are monitoring. In the Function Builder, get object data for measures, select **Browse**, expand the force of interest, and then select **force**. Then insert the object name into the text box of the Function Builder.

Remember, the force applied to the switch is a function of time. Before you run the simulation, you do not know how much force needs to be applied to toggle the switch; therefore, you do not know how long to simulate. For that reason, you create the sensor. You will purposely simulate for a larger amount of time than is needed, letting the sensor stop the simulation when the switch has been toggled.

4 Simulate the model to visually verify correct rearward toggle motion using a simulation script based on the following ADAMS/Solver commands:

```
SIM/STATIC
SIMULATE/DYNAMICS,END=10.0,DTOUT=0.1
DEACTIVATE/SENSOR,ID=<your right front sensor id #>
SIMULATE/DYNAMICS,DURATION=0.5,DTOUT=0.1
```

By using this simulation script, the model will simulate until the switch is toggled (assuming it toggles before 10 seconds), at which time the sensor is deactivated and the model simulates an additional 0.5 seconds to review follow-on transient behavior.

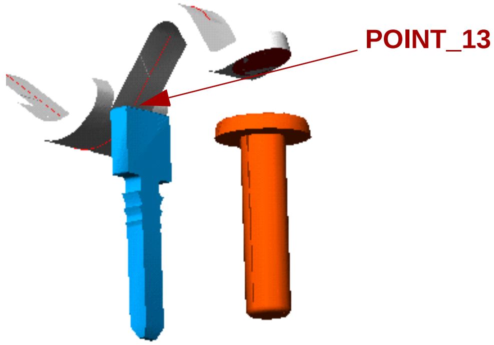
5 Save your work.

Section III: Refine the right half of the mechanism

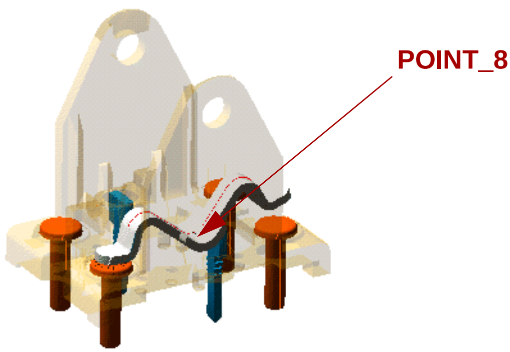
Replace the pivoting constraint at POINT_13 (the lower_contact to base revolute joint) with a more realistic connection that accounts for dynamic phenomena like sliding and liftoff.

To refine right_contact connections:

1 Remove the **revolute joint** constraining the **right_contact** to the base at **POINT_13**.



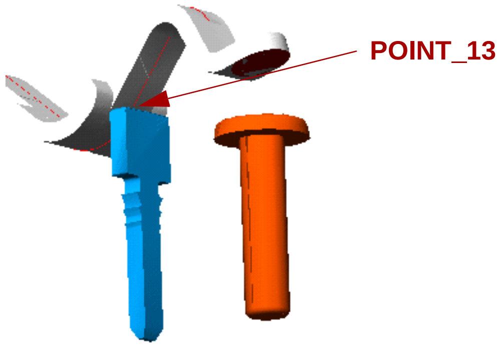
2 Constrain the **right_contact** to the base at **POINT_8** such that the only allowable degrees of freedom are translation along $\hat{z}_G$ and rotation about $\hat{y}_G$.



This involves creating two joint primitives.

You must ensure that the J marker of each primitive belongs to the base part, and not to the right_contact part. This will absolutely affect the simulation. See the instructor if you do not fully understand this concept.

3 Create a **point-to-curve contact force** between the underside on the **right_contact** part and the mid-contact point, **POINT_13**, on the base. Use the same contact parameters as in Step 3



First create a marker on the base part at POINT_13. Use this marker as the point marker in the point-to-curve contact force.

You must verify that the arrow displayed when the curve is selected indicates the side of the curve used by the contact force.

To test the model:

1 Verify the model.

Your system should have four degrees of freedom and no redundant constraints.

2 Simulate the model to visually verify correct rearward toggle motion using a simulation script based on the following ADAMS/Solver commands:

```
simulate/static
simulate/dynamics, end=10, dtout=0.01
deactivate/sensor, id=<your right front sensor id #>
simulate/dynamics, duration=0.5, dtout=0.01
```

Note the force at which the switch toggles to the rearward direction when accounting for only the right_follower and right_contact parts and corresponding connections.

3 Save your work now.

In Sections IV and V you will incorporate the left half of the switch and add friction.

Section IV: Add the left half

Since the right half of the switch mechanism is working properly at this time, follow Step 2 through Step 3 and add the left half of the mechanism. If desired, you can use a different crawl-walk-run method to connect parts in the left half.

Below is a copy of the key locations that will help you define the left half connections:

Table 3. Key Locations

Point:	Description:
POINT_1	Actuator to base pivot location
POINT_2	right_follower to actuator spring lower location
POINT_3	left_follower to actuator spring lower location
POINT_4	right_follower to actuator spring upper location
POINT_5	left_follower to actuator spring upper location
POINT_6	Contains z-coordinate of base contact plane with left_contact and right_contact at four corners
POINT_7	left_contact to base idealized constraint location
POINT_8	right_contact to base idealized constraint location
POINT_9	Location of base front contact surface with actuator
POINT_10	Location of base rear contact surface with actuator
POINT_11	Location of actuator front contact point with base
POINT_12	Location of actuator rear contact point with base
POINT_13	Location of right_contact to base mid-contact point
POINT_14	Location of left_contact to base mid-contact point
POINT_15	Location of force application

To connect the left_follower and left_contact:

- Reintroduce the **left_follower** and **left_contact** parts by reactivating them and ultimately connect these parts to the switch mechanism the same way that you did for the right_follower and right_contact parts.

To test the model:

1. Verify the model.

   Your system should now have seven degrees of freedom and no redundant constraints.

2. Simulate the model to determine rearward toggle motion using a simulation script based on the following ADAMS/Solver commands:

```
SIM/STATIC
SIMULATE/DYNAMICS,END=10.0,DTOUT=0.1
DEACTIVATE/SENSOR,ID=<your right front sensor id #>
SIMULATE/DYNAMICS,DURATION=0.5,DTOUT=0.01
```

3. Note the force at which the switch toggles to the rearward direction now, when accounting for both halves of the mechanism?

4. Simulate the model to determine forward toggle motion using a simulation script based on the following ADAMS/Solver commands:

```
SFORCE/<original input sforce id #>, FUNCTION=100*TIME
SIM/STATIC
SIMULATE/DYNAMICS,END=10.0,DTOUT=0.1
DEACTIVATE/SENSOR,ID=<your left rear sensor id #>
SIMULATE/DYNAMICS,DURATION=0.5,DTOUT=0.01
```

   Notice how you are reversing the input force applied to the actuator part through an ADAMS/Solver command in the simulation script, as opposed to in the model's design configuration. You can modify a force on-the-fly.

5. Note the force at which the switch toggles to the forward direction when accounting for both halves of the mechanism?

6. Save your work.

Section V: Refine the switch

You will now refine your model to account for friction.

To add friction to the circle-to-curve contact forces:

1. Modify the circle-to-curve contact force between the tip of the right_follower and the upper curve on the right_contact part such that static and dynamic friction is accounted for. Use the following default parameters for contact friction:

   - **Static Friction Coefficient: 0.09**
   - **Slip Velocity: 1 mm/sec**
   - **Dynamic Friction Coefficient: .05**
   - **Transition Velocity: 10 mm/sec**

   This will involve creating another contact array, since you do not necessarily want the other contact forces to account for friction.

2. Modify the circle-to-curve contact force between the tip of the left_follower and the upper curve on the left_contact part such that static and dynamic friction is accounted for. Use the same contact array you used in the previous step.

To test the model:

1. Verify the model.

   Your system should still have seven degrees of freedom and no redundant constraints.

2. Simulate the model to visually verify correct rearward toggle motion using a simulation script based on the following ADAMS/Solver commands:

```
SIM/STATIC
SIMULATE/DYNAMICS,END=10.0,DTOUT=0.1
DEACTIVATE/SENSOR,ID=<your right front sensor id #>
SIMULATE/DYNAMICS,DURATION=0.5,DTOUT=0.01
SIMULATE/DYNAMICS,DURATION=2.0,DTOUT=0.1
```

3 Simulate the model to visually verify correct forward toggle motion using a simulation script based on the following ADAMS/Solver commands:

```
SFORCE/<original input sforce id #>, FUNCTION=100*TIME
SIM/STATIC
SIMULATE/DYNAMICS,END=10.0,DTOUT=0.1
DEACTIVATE/SENSOR,ID=<your left rear sensor id #>
SIMULATE/DYNAMICS,DURATION=0.5,DTOUT=0.01
SIMULATE/DYNAMICS,DURATION=2.0,DTOUT=0.1
```

Note the force at which the switch toggles to the forward and rearward directions when accounting for friction in the contact between the follower parts and the contact parts. Were the effects of friction negligible in this mechanism?

A TABLES

This appendix contains tables that describe the various elements in ADAMS/View.

What's in this module:

Constraints Tables (Incomplete)

Table 4. Joints—Degrees of Freedom Removed

	Translation along X:	Translation along Y:	Translation along Z:	Rotation about X:	Rotation about Y:	Rotation about Z:	Total:
Fixed							
Revolute							
Translational							
Cylindrical							
Universal/hooke/ Constant velocity							
Spherical							
Planar							

Table 5. Curve Constraints—Degrees of Freedom Removed

	Translation along X:	Translation along Y:	Translation along Z:	Rotation about X:	Rotation about Y:	Rotation about Z:	Total:
Point-to-curve							
Curve-to-curve							

Table 6. Joint Primitives—Degrees of Freedom Removed

	Translation along X:	Translation along Y:	Translation along Z:	Rotation about X:	Rotation about Y:	Rotation about Z:	Total:
Orientation							
Inline							
Parallel axis							
Inplane							
Perpendicular							

Forces Tables (Incomplete)

Table 7. Pre-Defined Forces (Flexible Connections)

	Translational Spring-Damper:	Torsional Spring-Damper:	Bushing:	Beam:	Field:
Number of Bodies Affected					
Points of Application					
Number of Components					
Direction/ Orientation					
Magnitude					

Table 8. User-Defined Forces (Applied Forces)

	Single-Component Forces				Multi-Component Forces	
	Between 2 Bodies Translational:	Between 2 Bodies Rotational:	1 Body - Space Fixed:	1 Body - Moving:	Vector Force/ Torque:	General Force:
Number of Bodies Affected						
Points of Application						
Number of Components						
Direction/ Orientation						
Magnitude						

Constraint Tables (Completed)

Table 9. Joints—Degrees of Freedom Removed

	Translation along X:	Translation along Y:	Translation along Z:	Rotation about X:	Rotation about Y:	Rotation about Z:	Total:
Fixed	✓	✓	✓	✓	✓	✓	6
Revolute	✓	✓	✓	✓	✓		5
Translational	✓	✓		✓	✓	✓	5
Cylindrical	✓	✓		✓	✓		4
Universal/hooke/ Constant velocity	✓	✓	✓			✓	4
Spherical	✓	✓	✓				3
Planar			✓	✓	✓		3

Table 10. Curve Constraints—Degrees of Freedom Removed

	Translation along X:	Translation along Y:	Translation along Z:	Rotation about X:	Rotation about Y:	Rotation about Z:	Total:
Point-to-curve	✓	✓					2
Curve-to-curve	✓	✓					2

Table 11. Joint Primitives—Degrees of Freedom Removed

	Translation along X:	Translation along Y:	Translation along Z:	Rotation about X:	Rotation about Y:	Rotation about Z:	Total:
Orientation				✓	✓	✓	3
Inline	✓	✓					2
Parallel axis				✓	✓		2
Inplane			✓				1
Perpendicular						✓	1

Forces Tables (Completed)

Table 12. Pre-Defined Forces (Flexible Connections)

	Spring-Damper Translational:	Spring-Damper Torsional:	Bushing:	Beam:	Field:
# Bodies Affected	2	2	2	2	2
Points of Application	2 (I & J markers)	2 (I & J markers)	2 (I & J markers)	2 (I & J markers)	2 (I & J markers)
Number of Components	1	1	6	6	6
Direction/ Orientation	Line of sight between the (I & J markers)	Z-axis of J marker	J marker	J marker	J marker
Magnitude	Defined by parameters, such as stiffness, damping, cross-sectional area.				

Table 13. User-Defined Forces (Applied Forces)

	Single-Component Forces				Multi-Component Forces	
	Between 2 Bodies Translational:	Between 2 Bodies Rotational:	1 Body - Space Fixed:	1 Body - Moving:	Vector Force/ Torque:	General Force:
Number of Bodies Affected	2	2	1	1	2	2
Points of Application	2 (I & J markers)	2 (I & J markers)	1 (I marker)	1 (I marker)	2 (I & J markers)*	2 (I & J markers)*
Number of Components	1	1	1	1	3	6
Direction/ Orientation	Line of sight between I and J markers	A-axis of J-marker	Z-axis of J-marker	Z-axis of J marker	R marker	R marker
Magnitude	Defined by whole functions of which the user must take ownership.					

* The J markers created for a vector force/torque and a general force are floating markers.

B ANSWER KEY

What's in this appendix:

Answer Key ...

Answer Key for Workshop 1

Question 1, page 21: 269 mm

Question 2, page 21: 269 mm. This is the same as the previous results.

Question 3, page 21: 267.87 mm

Question 4, page 21: 6 - 5 make up the stamper mechanism, while 1 makes up the part parcels.

Question 5, page 21: 8 - 7 are on the stamper mechanism, while 1 keeps the parcels moving translationally.

Question 6, page 21: Nothing, the conveyor is simply a graphic attached to ground. It adds nothing to the model other than for animation purposes.

Answer Key for Workshop 2

Question 1, page 38: There are nine constraints (two revolute, one translational, three inplane, one orientation, one motion, one curve_curve). Motions are considered constraints; these will be covered in detail later in the course.

Question 2, page 38: Yes

Question 3, page 38: No, part geometry is a direct child of a part. Part geometry is a "grandchild" of a model..

Question 4, page 38: Status bar

Question 5, page 38: Our technical support staff prefers to receive .cmd files. They are smaller in size, and platform independent. Using .bin files is sometimes unavoidable, however.

Question 6, page 38: The find tool only searches the currently open pdf. The search tool (the one with binoculars) searches all the documents.

Answer Key for Workshop 3

Question 1, page 55: 1.635 pound mass based on geometry and density

Question 2, page 55: 100 lbf/foot

Question 3, page 55: Approximately 4.3 lbf

Answer Key ...

Answer Key for Workshop 4

Question 1, page 74: 4903 mm

Question 2, page 74: 9807 mm/sec

Question 3, page 74: 9807 mm/sec^2

Question 4, page 74: Coordinate system *markers*

Question 5, page 74: The ground part is automatically created - it must exist in every model. It serves as a reference frame for the model.

Question 6, page 74: No, because ADAMS cannot calculate a volume for two-dimensional objects. You can, however, *assign* mass properties to a part that is made up of two-dimensional geometry.

Answer Key for Workshop 5

Question 1, page 87: ~1.06 sec (can vary slightly depending on the sampling rate chosen).

Question 2, page 87: ~3180 mm (can vary slightly depending on the sampling rate chosen).

Question 3, page 87: The system constraint takes precedence.

Question 4, page 87: You would have to constrain the stone to ground with a revolute (pin) joint.

Answer Key for Workshop 6

Question 1, page 109: F_x = -29.9N, F_y = 17.24 N

Question 2, page 109: Approximately 0.61 Hz

Question 3, page 109: No, but if the two differ, the initial conditions in the constraint always override the initial conditions of a part.

Question 4, page 109: .human_hip.femur.MAR_1 and .human_hip.hip_bone.MAR_1. Draw it out on the board in the hierarchy format. Which one is I and which one is J depends on the order in which the parts were selected when creating the constraint.

Can the I and J markers for a joint belong to the same part? Why?, 109: No, a constraint constrains two different bodies to one another.

Answer Key for Workshop 7

Question 1, page 129: Between 16.5^o and 17^o (Exactly 16.7^o).

Question 2, page 129: Friction is only automated for revolute, translational, cylindrical, spherical, and hooke/universal constraints. By using forces, however, you can model friction on other joints.

Question 3, page 129: I and J markers are automatically created when you add a joint, motion, or force to a system. ADAMS uses I and J marker's relative displacement, velocity, and so on to define equations that describe part movement.

Question 4, page 129: Once the joint crosses the stiction threshold velocity, it exits the stiction phase and the maximum stiction displacement is ignored until the joint reenters the stiction phase (comes to rest). One of these two parameters is reached first, the other parameter is ignored until the joint enters the stiction phase again.

Answer Key for Workshop 8

Question 1, page 144: Construction geometry is two-dimensional, and solid geometry is three-dimensional.

Answer Key for Workshop 9

Question 1, page 152: I and J markers. The I marker belongs to the first body you selected when the creating the joint. The J marker belongs to the second body you selected.

Question 2, page 152: The orientation of the I and J markers. For example, if you added translational motion to a translational joint, the z-axis of the I and J markers would describe the axis of translation. The z-axis direction is positive.

Question 3, page 152: Yes. Even though they do not restrict movement, they still prescribe movement, therefore, removing degrees of freedom.

Question 4, page 152: Yes. You must measure the torque generated by the motion **not** the revolute joint.

Answer Key for Workshop 10

Question 1, page 160: The order in which you chose the bodies (parts) should be the same as the order in which you chose the corresponding locations and orientations.

Answer Key for Workshop 11

Question 1, page 168: A joint motion uses a joint to determine its direction and location. A point motion does not require a joint; it needs two bodies.

Answer Key for Workshop 12

Question 1, page 178: No. The point-to-point measure is just a quicker and easier way to create a function measure of the displacement of one marker with respect to another.

Question 2, page 178: A CAD file represents geometry in a model. Therefore, it is a child of a part.

Answer Key for Workshop 13

Question 1, page 188: You need to provide the two (or three) joints, and the one (or two) scalar coefficients for the constraint equation.

Question 2, page 188: Last_run

Answer Key for Workshop 14

Question 1, page 202: No, in the design configuration they do not have to be aligned. If they are not aligned, however, ADAMS warns you during a model verify or during the assemble simulation. Also, during the assemble simulation, ADAMS realigns the markers for you.

Answer Key for Workshop 15

Question 1, page 211:

- First independent variable
- Second independent variable
- Spline name
- Derivative order

Answer Key for Workshop 16

Question 1, page 219: ~1.46 degrees (.0255 radians)

Question 2, page 219: To remove the initial transient effects in the dynamic system because of mismatches in the preloading of the bushings.

Question 3, page 219: Because the model was kinematic, in this case (DOF=0), there is no initial transient response because you have specified the motion of the system for all points in time.

Answer Key for Workshop 17

Question 1, page 229: Yes, it must be greater than zero (not negative and not zero).

Question 2, page 229: Yes, to model a hardening compression-only spring, the exponent, e, must be greater than 1.

Answer Key for Workshop 18

Question 1, page 240: Approximately 200 N.

Question 2, page 240: Yes. For example, you could simulate the model with output step sizes of 0.01 seconds. When that simulation is complete, don't reset the model. Start another simulation with a step size of 0.001. The results of that simulation will be seamless, but you will notice a change when the step size changes. The animation changes speeds. A common reason for doing this is if you want the simulation to use smaller step sizes or be more accurate before a contact.

Question 3, page 240: No, a Simple Run script only allows for one simulation.

Answer Key for Workshop 19

Question 1, page 251: A statement describes an element in a model, such as a part or force. A command tells ADAMS/Solver what to do with the model, such as simulate it or deactivate it.

Answer Key for Workshop 20

Question 1, page 264: A curve-on-curve constraint removes two translational DOF.

Question 2, page 264: A curve-to-curve contact force removes no DOF.

Answer Key for Workshop 21

Question 1, page 271: The reference marker (R marker)

Answer Key for Workshop 22

Question 1, page 281: Between 17 and 18 N/mm