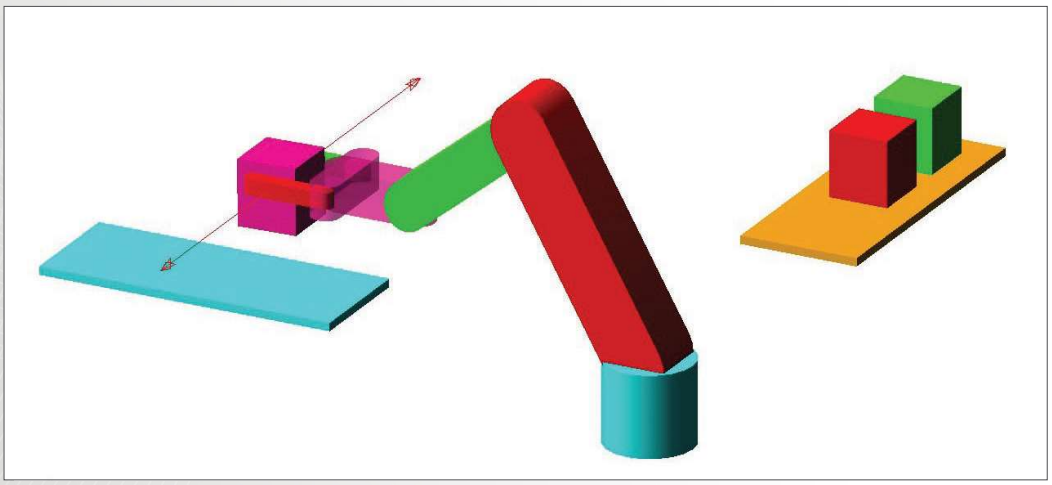
Example 18: Robot Arm



Workshop Objectives

- Construct a robot arm in Adams
- Manipulate the working grid for use with multi-planar part layouts
- Create a gear constraint between revolute joints
- Use a SFORCE to apply griping torque to a robot manipulator
- Define 3D object contact and friction
- Synchronize joint motions and motor toques to perform a complex task

Software Version

Adams 2013.2

Step 1. Build the Lower Links

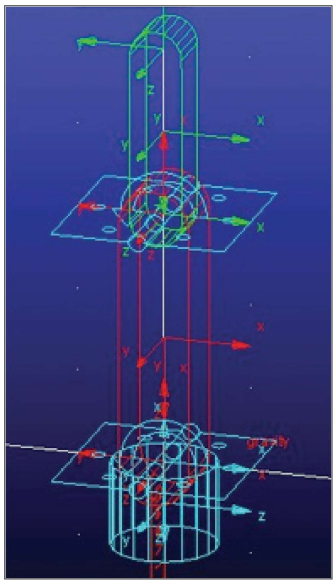
a. Start Adams/View.
b. Create a new model. **(Model Name = robot_arm, Units = mmks, Gravity = -y earth)**
c. Create a **Link** from **(0, 0, 0)** to **(0, 150, 0)** and rename it **lower_link**.

- Note that the working grid, by default, snaps to 50mm spaced grid points, which make it easy to select the specified points.

d. Build another from **(0, 150, 0)** to **(0, 250, 0)**. Rename it **middle_link.**
e. Modify the link geometries of **lower_link** to have a **width** and **depth** of **40**. For **middle_link**, set its link geometry to a **width** and **depth** of **30**.
f. Build a **Cylinder** with a start point at **(0, 0, 0)** and an end point at **(0, -50, 0).** Rename it **base**.
g. Modify the cylinder **radius** to be **30**.
h. Select the **Revolute Joint** follow the instruction in the status bar to create the following joints.

First Body	Second Body	Location
lower_link	base	0,0,0
middle_link	lower_link	0,150,0

- The order of the body selections is important because it determines which direction is positive for applied motion. After all of the joints are created, the model will be tested to make sure motions act in the correct direction. If not, this can easily be changed.



Step 2. Change the Working Grid

a. Go to **Settings >> Working Grid**
b. Set **Orientation** to **Global YZ** plane and spacing to **10mm x 10mm.**
c. Select **OK**.
d. Press **Shift + R** to change to **right view**.

- The XY working grid was ideal for creating the lower links and base, but the YZ is much more convenient for building the manipulator. Not only does this cause the cursor to snap to points in the appropriate plane, but also allows the use the use to the default **Create Normal to Grid** option on Joints and other entities.

Step 3. Build the Manipulator I

a. Decrease the size of the icons to make working near them easier:

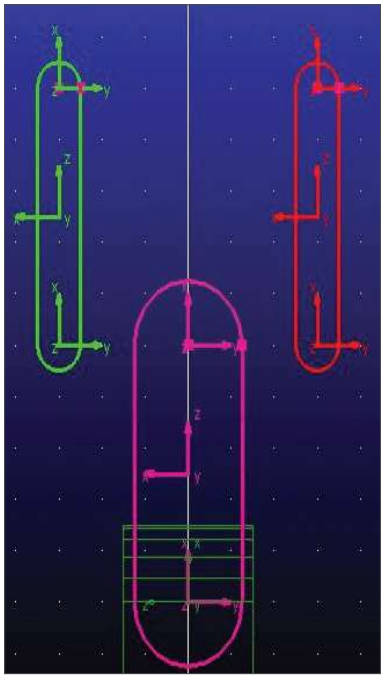
- Go to **Settings >> Icons**… and type **10** in the **New Size** box

b. Build and resize links for the manipulator as shown below.

c. Referring to the image below, (your colors may differ) rename the **MAGENTA** link **manipulator_base**, the **RED** link **gripper_right** and the **GREEN** link **gripper_left.**

d. Save your work.

Link	End Points	Width	Depth
manipulator_base	(0, 300, 0) (0, 250, 0)	20mm	20mm
gripper_right	(0, 350, -30) (0, 300, -30)	10mm	10mm
gripper left	(0, 350, 30) (0, 300, 30)	10mm	10mm



Step 4. Build the Manipulator II

For this simulation, the robot will be grasping 40mm square cubes. Use precision move to correctly space the grippers.

a. Select **gripper_left.**

b. Select the **Position:Move** icon in the Main Toolba 

c. Type **5mm** in the **distance** box.

d. Select **Vector** and select any vector in the global z direction

e. Repeat for **gripper_right**, moving it left **5mm** instead.

Finish building the geometry for the manipulator base part

f. Select the **Rigidbody:Link** icon. 

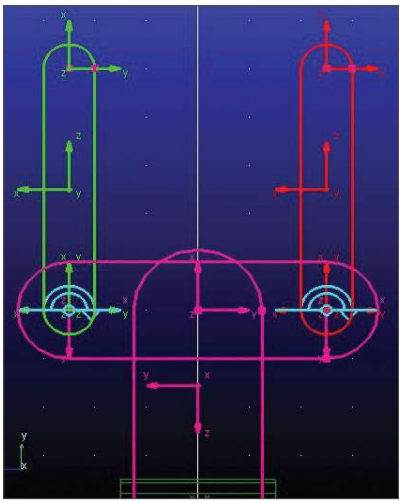
g. Change from **New Part** to **Add to Part** in the drop-down menu

h. Select the check boxes **Width** and **Depth**, and enter **20** into each field

i. Select the **manipulator_base** part, then select the lower markers of each gripper part to define the new link geometry

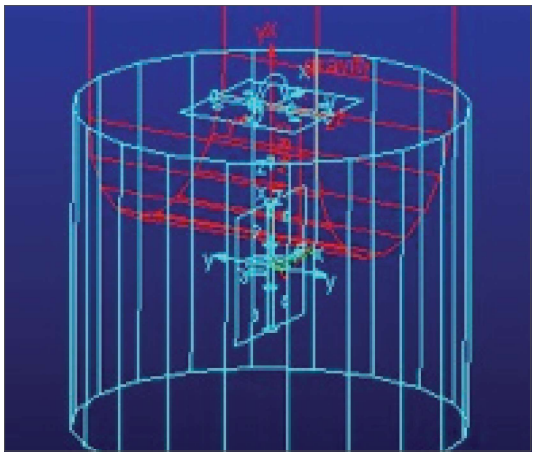
Note that this is a new geometry added to the manipulator_base part, not a new part.

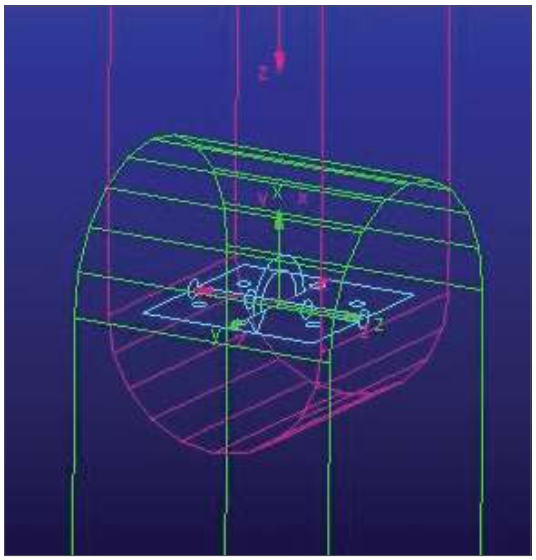
j. Build **revolute joints** between each of the **grippers** and **manipulator_base** as shown, selecting the grippers at the first bodies



Step 5. Define the Remaining Joints

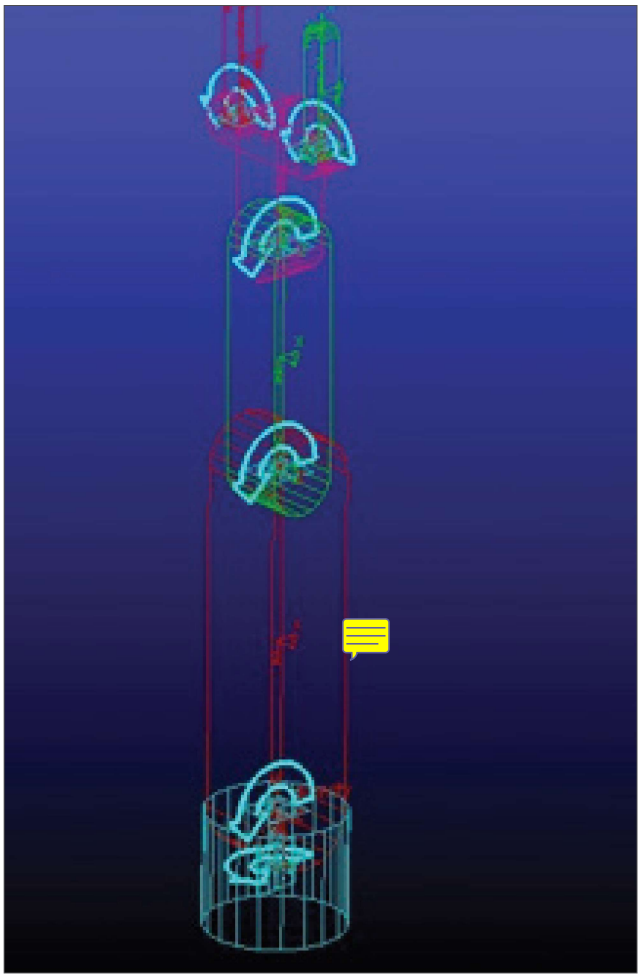
a. Switch the working grid orientation to the **Global XY**.
b. Build a revolute joint between **manipulator_base** and **middle_link** at **(0, 250, 0)**
c. Switch the working grid orientation to **Global XZ.**
d. Build a revolute joint between **ground** and **base** at the **bases .cm marker**, which is at **(0, -25, 0).**





Step 6. Add Motions to Test the Model

a. Click the **Rotational Joint Motion** tool from the Main Toolbar. Use the default speed of **30.0.**
b. Select the joint between **lower_link** and **base**.
c. Rename the newly created motion **motor_1.**
d. Modify **(Right Click >> motor_1 >> Modify…)** the newly created motion. Add a negative sign to the function line to reverse its direction
e. Add motions to the other 5 revolute joints, using the default speed of 30 for each motion. It is not necessary to change the direction of sign/direction of these motions.
f. Rename each of the motions as shown.
g. Simulate for the default of **5 seconds** and **50 steps**.



Step 7. Test the Model

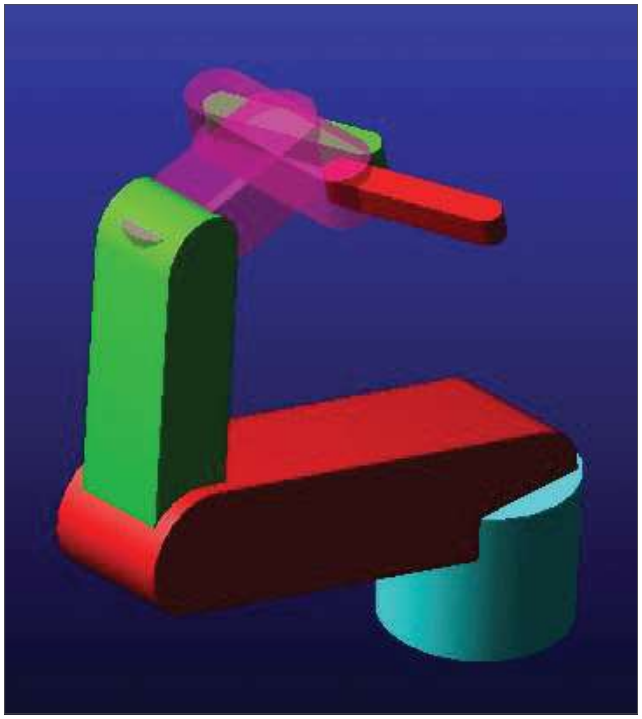
a. Click on the **animation** icon. 

b. Use the slider bar to navigate to **frame 26.**

c. Shown is the **iso (Shift+I) shaded (Shift+S)** view of the model at frame 26 (time 2.5000). Manipulator_base has been made transparent for visualization.

   a. If your model does not behave as shown, attempt to make the necessary changes.

   b. Likely, the issued can be fixed by reviewing the slides (5-8 for position related issues, and 12-13 for joint related issues). If this does not resolve the issue, load robot_arm_shortcut1.bin and continue from there.



Step 8. Build Objects to Grasp I

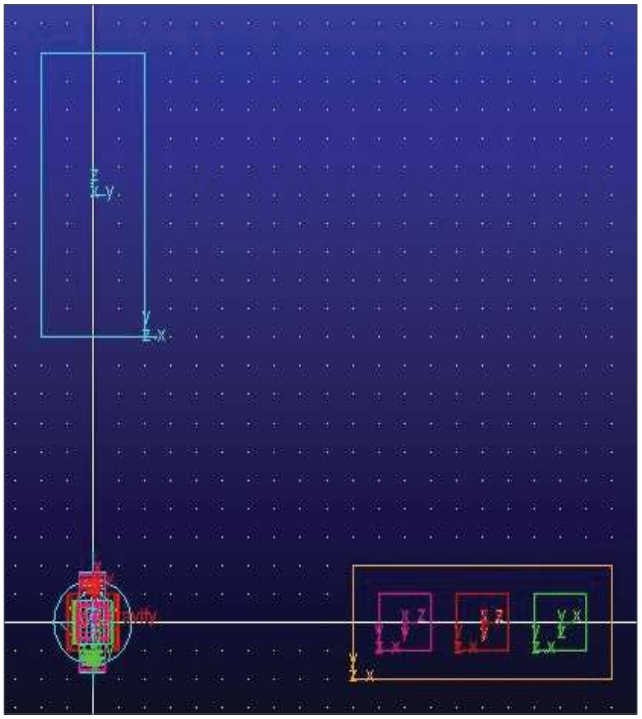
a. Switch working grid to **Global XZ** and spacing to **20 x 20.** Switch to **top view**.

b. Use the **Rigid Body: Box** to build boxes as shown below by selecting the appropriate corner locations.

c. Modify the block geometry of each part, changing the **Z component of Diagonal Corner Coords** to **-5** for the large 'platforms' and 40 for each of the cubes.

d. Rename the geometries of the newly created bodies as shown below.

e. Change the mass of each cube to **20g. Modify >> Define Mass By: User Input**



Step 9. Build Objects to Grasp II

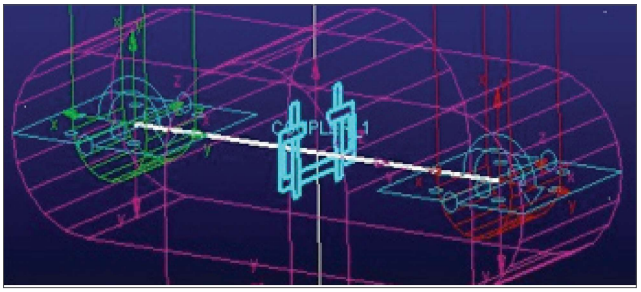
Now we will fix the platforms to ground, and create contact between the boxes and platforms so they do not fall into space.

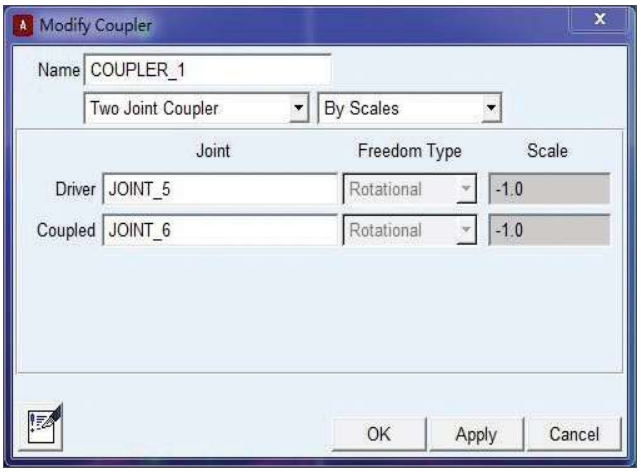
a. Select **Contact** icon 
b. Make sure **Solid to Solid** is selected in the **Contact Type** drop down menu
c. Right click in the **I Solids** and **J Solids** dialog box and use the Pick to select the geometries in the Main Window.
d. Repeat for the other three cubes.
e. Create **fixed joints** between each **platform** and **ground**.

Step 10. Couple the Motion of the Grippers

Now we will fix the platforms to ground, and create contact between the boxes and platforms so they do not fall into space.

a. Set the Working Grid back to **Global YZ** and switch to **Right View.**
b. Delete the motions acting on the gripper revolute joints.
c. Select the **Joint Coupler.** 
d. Choose the **gripper_right revolute joint**, then the left, to define a motion coupler.
e. Modify the coupler as shown. This constrains the motion of the joints to be equal in magnitude but opposite in direction.





Step 11. Add Forces to the Gripper

a. Select **Create a Rotational Spring-damper.**

b. Select **gripper_right** then **manipulator_base** for the bodies to define the spring, then the center of the revolute joint for location.

c. Modify the torsion spring to have a **Stiffness** of **10**, **Damping** of **5** and a **Preload** of **10**. The causes the grippers to spring slightly open by default.

d. Select **Create a Torque.**

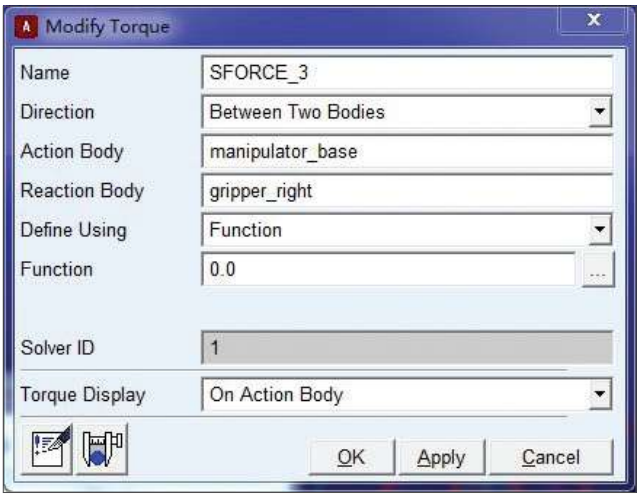
- Note: Although the torque should act between **gripper_right** and **manipulator_base**, use the default **Space Fixed** option (which reacts on ground) to take advantage of the default **Normal to Grid** for direction. The **SFORCE** will later be modified to react on **manipulator_base**.

e. Select **manipulator_left** for the body to define the **SFORCE** and the center of the **gripper_right** revolute joint for location.

f. Change the appearance of the SFORCE to have a color of **blue** and a size of **11** for visibility.

g. Rename the torque **SFORCE_grip_torque**.

h. Modify the SFORCE as shown.



Step 12. Test the Gripper Forces

a. Set the other motions in the model to be 0 and verify the operation of the manipulator.

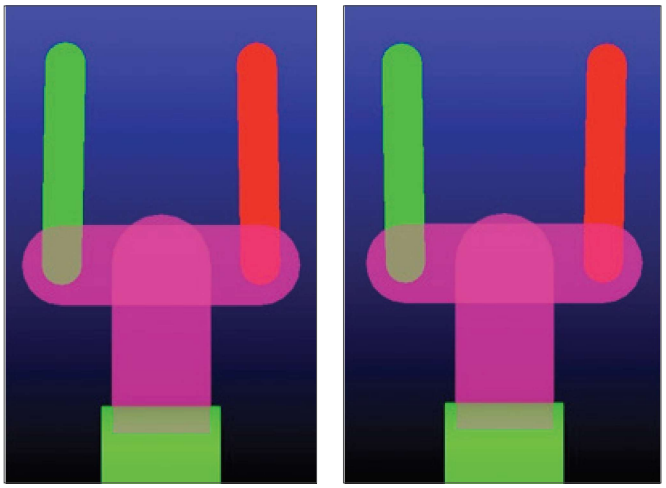
b. Set all 4 of the joint motion (**motor_1, motor_2, etc.**) function definitions to be **0**.

c. Modify **grip_torque's function** to be **20.**

d. Simulate for **3 seconds, 30 steps**, the grippers should now settle in a slightly closed position, as shown.

e. Change **grip_torque's function** to be **0.**

f. If the grippers do not behave as shown, refer to the Modify dialog boxes and images for the torsion spring and grip_torque on the previous slide to check their definitions

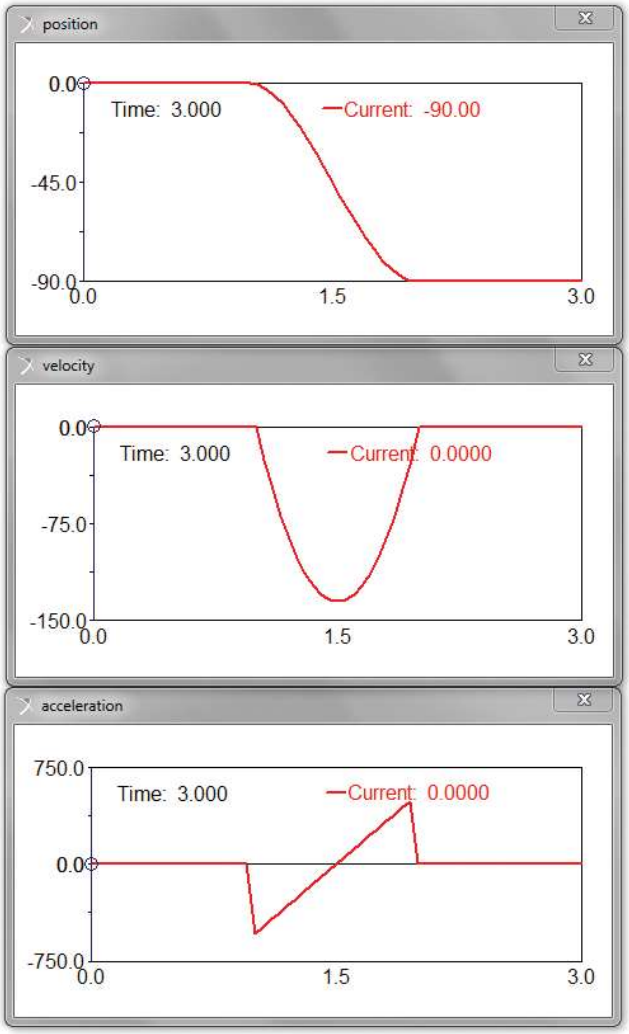


Step 13. The Step Function Introduction

A step function will be used to define the motion of the robot. The step function dependent variable has an initial valve h0, before the independent variable, x, reaches x0, a final value h1 after the x reaches x1, and a smooth step in between. “Smooth” mean the first derivative is continuous (i.e. no instantaneous change in acceleration). Position, Velocity, and Acceleration of a step defined motion are show below.

Step Function: **-90d*step(time,1,0,2,1)**

As you can see, detailed information on Adams functions can be found in the Adams help documentation



STEP

The STEP function approximates the Heaviside step function with a cubic polynomial. It has continuous first derivatives. Its second derivatives are discontinuous at x=x0 and x=x1.

Format

STEP (x, x0, h0, x1, h1)

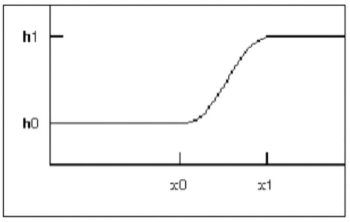
Arguments

x	The independent variable. It can be a function expression.
x0	A real variable that specifies the x value at which the STEP function begins.
x1	A real variable that specifies the x value at which the STEP function ends.
h0	The initial value of the step.
h1	The final value of the step.

Extended Definition

The STEP function approximates the Heaviside step function with a cubic polynomial. The figure below illustrates the STEP function.

Step Function



The equation defining the STEP function is:

$$a = h_1 - h_0$$

$$\Delta = (x - x_0)/(x_1 - x_0)$$

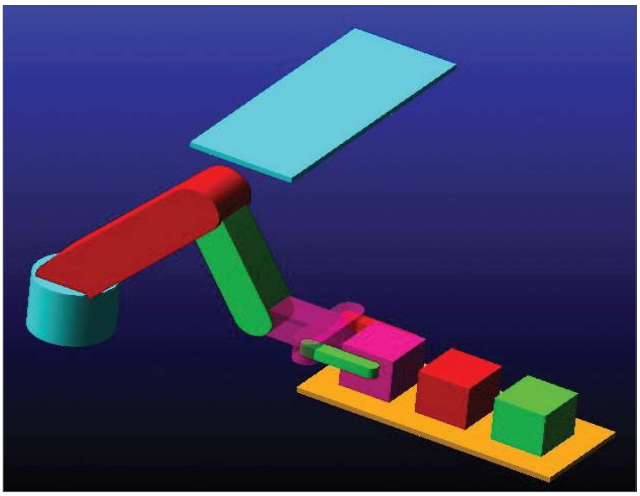
$$STEP = \begin{cases} h_0 & : x \le x_0 \\ h_0 + a \cdot \Delta^2(3 - 2\Delta) & : x_0 < x < x_1 \\ h_1 & : x \ge x_1 \end{cases}$$

Step 14. "Stepping" the Robot

Now, motions will be defined with step functions to bring into position to grip cube_1.

a. Define the following step functions. Be sure delete what is already in the function box and select apply in the motion modify window.
b. Run a simulation for **1 second, 20 steps.**
c. At the end of the simulation, the gripped should be positioned to grip the cube, as shown.

Motions	Function
Motor_1	step(time,0,0,1,-40d)
Motor_2	step(time,0,0,1,-110d)
Motor_3	step(time,0,0,1,60d)
Motor_4	0



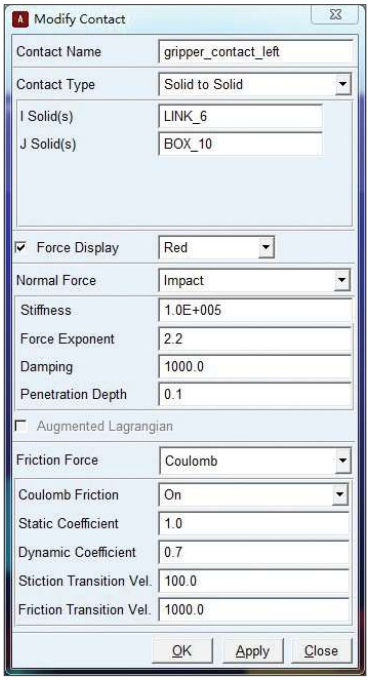
Step 15. Grip the Cube

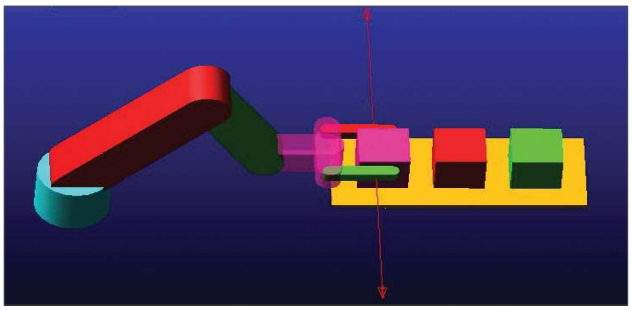
Next create a contact and friction force between the grippers and the cube.

a. Create a solid to solid contact between **gripper_left** and **cube_1.**
b. Change the **Static** and **Dynamic Coefficients** as shown.
c. Repeat for **gripper_right** and **cube_1.**
d. Rename each **gripper_contact_[left/right].**

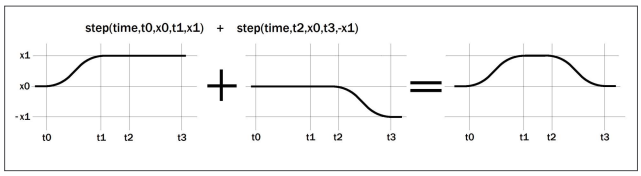
Use a step function to set torque equal to 0 from 0-1sec, allowing the spring the keep the grippers in the slightly open position, then apply 8000 N*m when in position.

e. Modify **grip_torque's function** to be **-step(time,1,0,1.1,-8000)**
f. Simulate for **1.1 seconds, 55 steps**. Confirm that the gripper makes contact with cube_1 as shown.





Step 16. Adding Steps

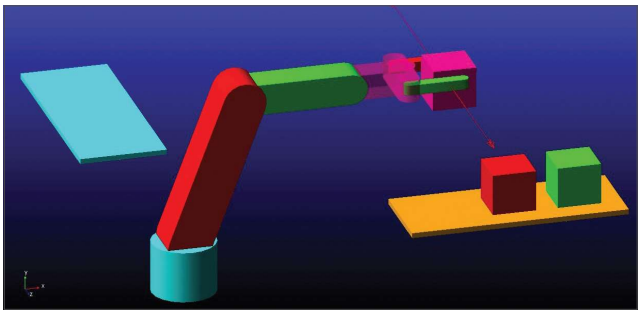


a. Add the following steps to the specified motion functions:

Motions	Step to Add
Motor_1	step(time,1.1,0,2,20d)
Motor_2	step(time,1.1,0,2,40d)
Motor_3	step(time,1.1,0,2,-60d)

b. Simulate for **2 seconds, 40 steps.**

c. Verify that your simulation matches with the image.



Step 17. Finalize Definition of Motions and Torques

The chart below describes the necessary step values required to complete the entire operation. Try to figure out the missing values on your own. Recall that x values are the independent value (time) and the h values are the dependent variables (position/torque) and the format for the step function (with time defined as X) is step(time,x0,h0,x1,h1)

	Already Done													
Movement Description	**Step1** Move into position		**Step2** Grip the cube		**Step 3** Lift		**Step4** Move to platform2		**Step5** Lower into place		**Step6** Release		**Step7** Raise manipulator	
Time Values	x0	x1	x0	x1	x0	x1	x0	x1	x0	x1	x0	x1	x0	x1
	0	1	1.1	2	2	3	2.5	3.5	3	4	4	4.1	4.1	5
Motion/ Torque Values	h0	h1	h0	h1	h0	h1	h0	h1	h0	h1	h0	h1	h0	h1
motor_1	0	-40d	N/A	N/A	0	20d	N/A	N/A			N/A	N/A		
motor_2	0	-110d	N/A	N/A	0	40d	N/A	N/A			N/A	N/A		
motor_3	0	60d	N/A	N/A	0	-60d	N/A	N/A			N/A	N/A		
motor_4	N/A	N/A	N/A	N/A	N/A	N/A			N/A	N/A	N/A	N/A	N/A	N/A
grip_torque	N/A	N/A	0	8000	N/A	N/A	N/A	N/A	N/A	N/A			N/A	N/A

Add to motion definitions one step at a time and simulating to verify the model behaves as expected.

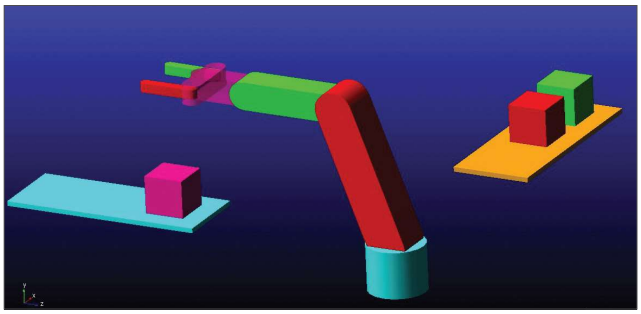
Note that all h0 values are zero and h1 always describes motion relative to the current position.

The necessary motion to place the cube on the platform and release it is essentially the reverse of picking it up. In other words, steps 3, 5, and 7 are very similar.

For example, the final function definition of motor_2 is: **step(time,0,0,1,-110d) + step(time,1.1,0,2,40d) + step(time,2,0,3,-40d) + step(time,4.1,0,5,40d)**

Step 4 is simple a rotation of the base to bring the cube over platform_2. Note that its x values overlap steps 2 & 5.

Try to simulate the whole operation (0-4 seconds) with your values. If you are having trouble, continue to the next step.



Step 18. Finalize Definition of Motions and Torques

	Already Done													
Movement Description	Step1 Move into position		Step2 Grip the cube		Step 3 Lift		Step4 Move to platform2		Step5 Lower into place		Step6 Release		Step7 Raise manipulator	
Time Values	x0	x1	x0	x1	x0	x1	x0	x1	x0	x1	x0	x1	x0	x1
	0	1	1.1	2	2	3	2.5	3.5	3	4	4	4.1	4.1	5
Motion/ Torque Values	h0	h1	h0	h1	h0	h1	h0	h1	h0	h1	h0	h1	h0	h1
motor_1	0	-40d	N/A	N/A	0	20d	N/A	N/A	0	-20d	N/A	N/A	0	20d
motor_2	0	-110d	N/A	N/A	0	40d	N/A	N/A	0	-40d	N/A	N/A	0	40d
motor_3	0	60d	N/A	N/A	0	-60d	N/A	N/A	0	60d	N/A	N/A	0	-60d
motor_4	N/A	N/A	N/A	N/A	N/A	N/A	0	90d	N/A	N/A	N/A	N/A	N/A	N/A
grip_torque	N/A	N/A	0	8000	N/A	N/A	N/A	N/A	N/A	N/A	0	-8000	N/A	N/A

Final list of motion/torque functions:

Motions	Function
Motor_1	step(time,0,0,1,-40d) + step(time,1.1,0,2,20d) + step(time,2,0,3,-20d) + step(time,3.1,0,4,20d)
Motor_2	step(time,0,0,1,-110d) + step(time,1.1,0,2,40d) + step(time,2,0,3,-40d) + step(time,3.1,0,4,40d)
Motor_3	step(time,0,0,1,60d) + step(time,1.1,0,2,-60d) + step(time,2,0,3,60d) + step(time,3.1,0,4,-60d)
Motor_4	step(time,1.5,0,2.5,90d)
gripper_torque	step(time,1,0,1.1,8000) + step(time,3,0,3.1,-8000)

Step 19. Optional Tasks

Torque Demand

a. Switch to **PostProcessor** and examine the results of the simulation.
b. Look at torque demands **(Source:Objects >> motor _x>> Element Torque >> Mag),** and gripper contact forces**(Source:Objects >> gripper_contact_[left/ right]>> Element Torque >> Mag).**
c. Note the sporadic spikes in torque required to maintain the smooth step motion.
d. Switch back to **View** and increase the **contact damping** to **100** and re-simulate.
e. How do the torque demands and contact forces look now?

Move the Remaining Cubes

For simplicity, the robot sits the block down in the same position on platform 2 as it was on platform 1.

a. Use sketch paper to derive the necessary angles to sit the block near the far edge of platform 2 , where cube 3 is on platform 1.
b. Try to create the additional steps necessary to move the remaining blocks. Derive the necessary angles by hand or use trial and error to determine the correct values.