RobotModeler User's Guide

Version 1.0

Steven J. Rodenbaugh
David E. Orin
Department of Electrical Engineering
The Ohio State University

May 30, 2003

Revision History

May 30, 2003	Updated for Version 1.0 (Lucas Frankart)

Background

RobotBuilder and RobotModeler were originally developed as part of Steven Rodenbaugh's M.S. thesis work. Defining and saving a user view and other camera control functions were implemented by Luke Frankart as a part of his Bachelor's Honors thesis. The applications are offered for free to encourage research in the field of robotics. While considerable effort has been dedicated to verifying the accuracy of all calculations, the results are not guaranteed to be valid. For more information about the applications, please consult the User's Guide or the RobotBuilder website at http://eewww.eng.ohio-state.edu/~orin/RobotBuilder/RobotBuilder.html.

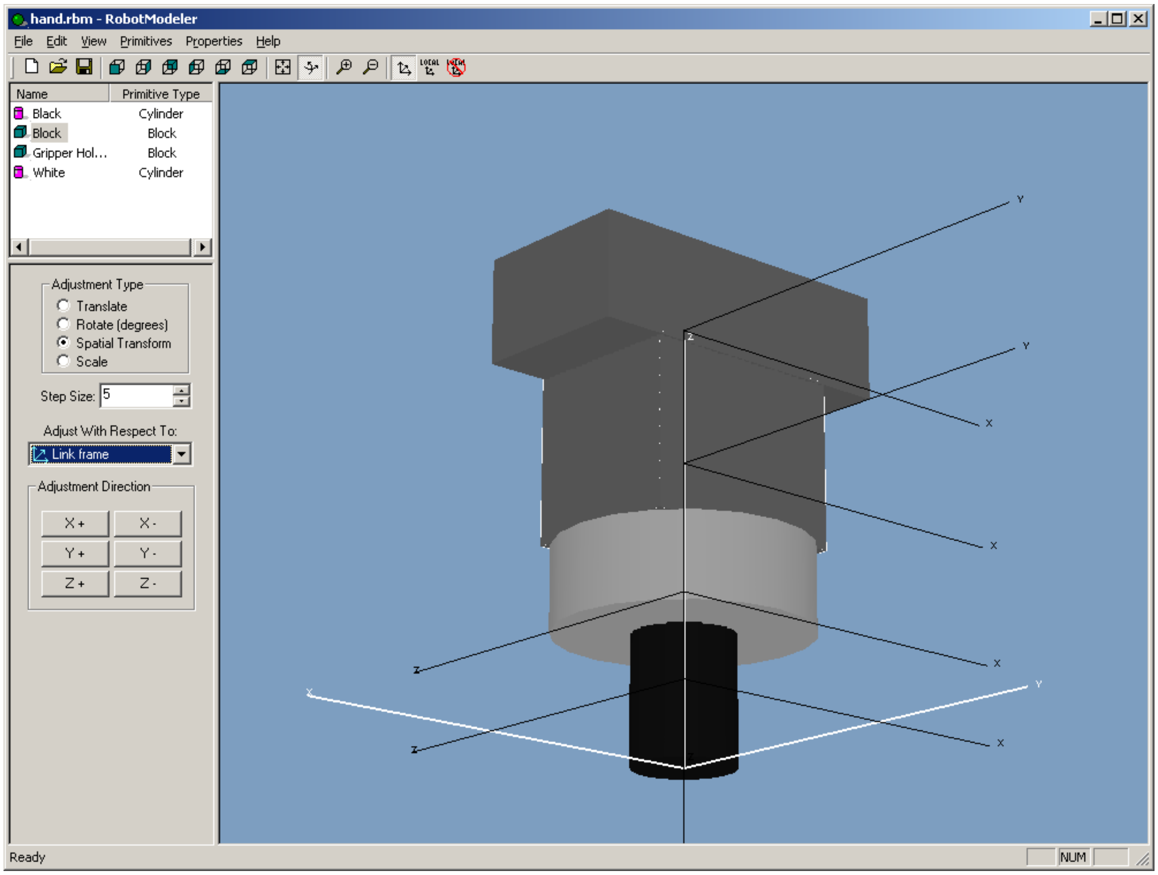
TABLE OF CONTENTS

1. Introduction

RobotModeler is an application that enables the researcher to quickly develop graphical models for use in RobotBuilder. It has a simple user interface for ease of use, but has enough features to create detailed models rapidly. The application will also automatically calculate the physical properties of the model for use in RobotBuilder.

Models are created by assembling simple 3D shapes. The 3D shapes available, also referred to as graphic primitives, are the block, cone, cylinder, hemisphere, sphere, and truncated cone. The user has the ability to set the parameters that define these shapes, and then position them as desired by translating and rotating the shape.

A screenshot of the application can be seen in Figure 1. At the top of the main window, there is a menu bar and a toolbar. Below that there are three main windows. The window in the top left is the Primitive List View where the primitives that currently make up the model are enumerated. Below that window is the Modification Pallet. This window enables the user to easily scale, translate, rotate, and spatially transform any primitive or primitives selected. The window on the right is the 3D Model View of the model, using the *WorldToolKit* graphics library.



hand.rbm - RobotModeler

Name	Primitive Type
Black	Cylinder
Block	Block
Gripper Hol...	Block
White	Cylinder

Adjustment Type

○ Translate
○ Rotate (degrees)
● Spatial Transform
○ Scale

Step Size: 5

Adjust With Respect To: Link frame

Ready

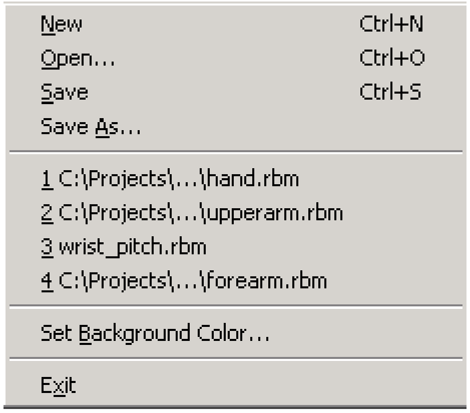
Figure A1: RobotModeler application screenshot.

2. Menu

The menu bar provides access to many functions offered by RobotModeler. The main menu items are:

2.1 File Menu

The “File” menu provides access to file related functions.



New

“New” will clear the current model and start with an empty project.

Open...

“Open...” will open the Open dialog which will let the user open an already existing model created with RobotModeler. By default, the file filter will be set to show only `.rbm` files, the extension used for RobotModeler files, but it can be changed to show all files.

Save

Saves the current model to a file. If a filename has not been assigned, a Save file dialog will be automatically opened to enable the user to specify the file name and path.

Note: The application does not warn if the current model is not saved when the user opens a new one or exits the application.

Save As...

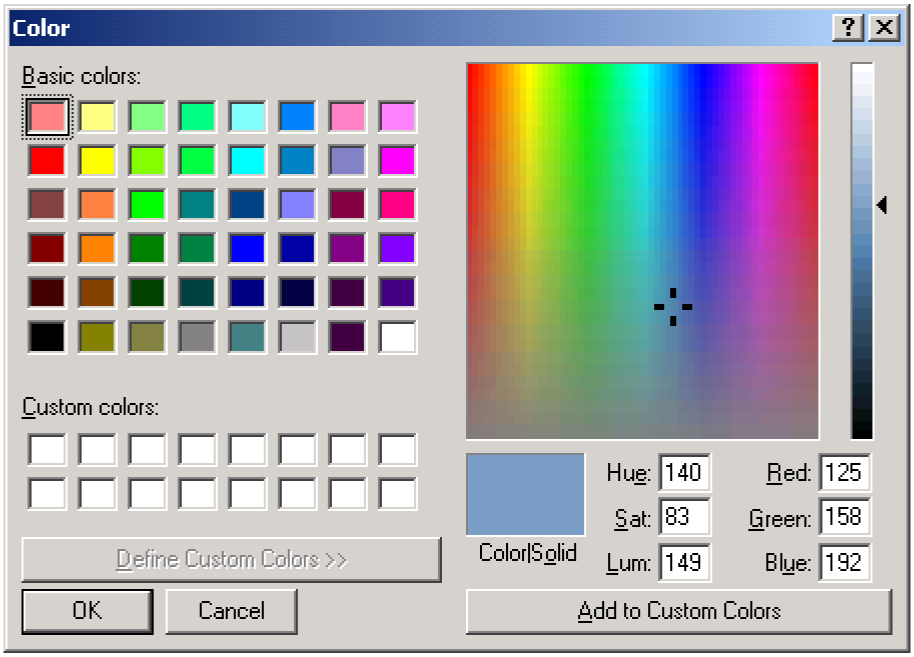
Opens the Save dialog to enable the user to save the current model to a file.

Recently Used File List

After the “Save As...” option, there will be up to four items which list recently used files. Clicking on one of these will open the file if it exists.

Set Background Color...

Opens a dialog that enables the user to specify the background color. The dialog will be initialized with the current background color. This data is saved with the model, so the next time the model is opened, the background color will be changed to the saved background color.



Hue: 140
Sat: 83
Lum: 149
Red: 125
Green: 158
Blue: 192

Note: The color dialog lets the user specify “Custom Colors” to save a color for the current instance of the application. To specify a “Custom Color”, choose the custom color square to set (otherwise the default will overwrite the first square), adjust the RGB or HSL as desired, and click “Add to Custom Colors” button.

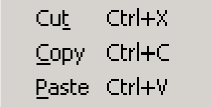
Exit

Closes the application.

Note: Be sure to save your work before you exit. The application will NOT prompt you to save your work if it has been modified.

2.2 Edit Menu

The "Edit" menu provides the user with common editing functionality.



Note: This editing capability does not utilize the Windows clipboard. This means that it won't overwrite what is currently there, but also implies that the buffer is only available to the current instance of RobotModeler.

Cut

Copies the currently selected primitive or primitives from the Primitive List View to the buffer, then removes the primitives from the list. This is only enabled when one or more primitives are selected.

Copy

Copies the currently selected primitive or primitives from the Primitive List View to the buffer. This is only enabled when one or more primitives are selected.

Paste

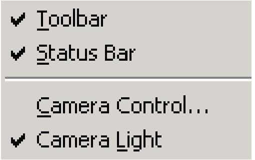
Adds one or more primitives in the buffer to the current model. This is only enabled when one or more primitives have been cut or copied to the buffer.

2.3 View Menu

The "View" menu enables the user to affect the visibility of some components of the application, and change how the model is viewed in the 3D Model View.

Toolbar

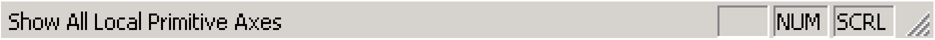
Toggles the visibility of the application's toolbar, located below the menu bar.



Note: See Section 3 for more information about the toolbar.

Status Bar

Toggles the visibility of the application's status bar. The status bar is located at the bottom of the main window. It shows help information on the left related to the current menu item or toolbar button and the state of the 'Caps Lock', 'Num Lock', and 'Scroll Lock' keys on the right.



Show All Local Primitive Axes

Camera Control...

Opens the `Camera Control` dialog box. The `Camera Control` dialog box enables the user to change settings that affect how the camera (viewpoint) is positioned and oriented. Before analyzing the camera control settings, the camera coordinate frame must be understood.

The coordinate frame of the camera is defined as the right-hand system with positive **z** pointing in the direction the camera is looking, and **x** pointing to the right, parallel to the window and in the plane of the viewpoint.

The camera is controlled by the mouse in the 3D Model View window. The application monitors for the left or right mouse button to be pressed down and released. This implies that the mouse can be clicked for an incremental change, or the mouse button can be held down to repeatedly apply the change

The effect of the change in the camera position is proportional to the distance between the mouse pointer and the center of the window. Consequently, clicking the mouse at the edge of the window will cause the greatest change and clicking near the

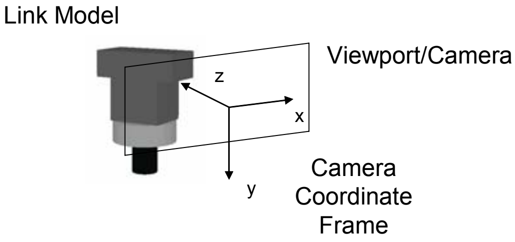


Figure 2: Camera coordinate frame.

center of the window will cause a smaller change.

The camera control mode will affect how the application interprets mouse commands made in the 3D Model View window. There are two main modes available for camera control: `Pan View` and `Examine View`.

In `Pan View`, the camera merely translates in its current plane, or zooms in or out. The mouse affects the camera in the `Pan View` in the following ways:

- Left button in the right half of the window will translate the camera in the camera's **x** direction.
- Left button in the left half of the window will translate the camera in the camera's –**x** direction.
- Left button in the top half of the window will translate the camera in the –**y** direction.
- Left button in the bottom half of the window will translate the camera in the **y** direction.
- Right button in the top half of the window will zoom in (translate in **z** direction).
- Right button in the bottom half of the window will zoom out (translate in the –**z** direction).

Note: The effects will be combined in some cases. For example, pressing the left mouse button in the top-right part of the window will translate in the ***x*** *<u>and</u>* ***–y*** *directions.*

In Examine View mode, the camera always points toward a chosen object, referred to as a center of interest, and the camera's movement is constrained to a sphere centered at the center of interest. There are two variations of Examine View. In the default mode, the changes are made relative to the link coordinate system. When there is a clear default orientation for the object, this mode is sometimes less difficult to control, because it is easy to rotate the camera to obtain an intuitive view. The other variation makes the changes relative to the camera (viewport) coordinate frame. This mode is convenient to work in because the changes are consistent no matter how the object is viewed.

When in Examine View relative to the link coordinate frame, the camera control can be thought of as an angle above or below the **x-y** plane of the link coordinate frame translated to the center of interest, and a rotation about the link coordinate frame's **z** axis translated to the center of interest.

- Left button in the right half of the window will rotate the camera about the **z** axis. *Note that the apparent direction of the change will switch when the model is upside down.*
- Left button in the left half of the window will rotate the camera about the negative **z** axis.
- Left button in the top half of the window will increase the camera's angle above the **x-y** plane. Increasing to greater than 90º will result in a decrease in the angle with the camera being upside down. This implies that the object can be orbited with the camera in the same plane.

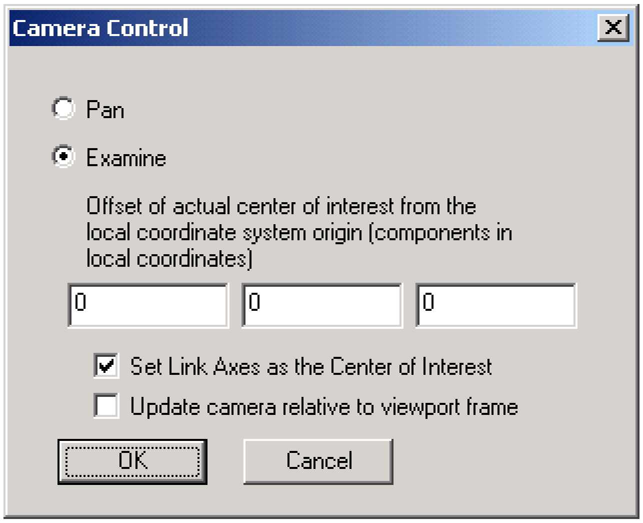
- Left button in the bottom half of the window will decrease the camera's angle above the **x-y** plane.
- Right button in the top half of the window will zoom in (shrink the sphere centered at the center of interest, that the camera operates on).
- Right button in the bottom half of the window will zoom out (increase the size of the sphere that the camera operates on).

Note: To be flexible, RobotModeler allows the user to zoom in within 1 unit of the coordinate frame. If the camera is zoomed inside a primitive, the model may look invisible because the backfaces of primitives may not be rendered.

When the `Examine View` is relative to the camera coordinate frame, the camera also operates in a spherical shell centered at the center of interest, but now the changes are relative to the camera coordinate frame:

- Left button in the right half of the window will rotate the camera in a negative angle about the **y** camera frame axis.
- Left button in the left half of the window will rotate the camera in a positive angle about the **y** camera frame axis.
- Left button in the top half of the window will rotate the camera in a negative angle about the **x** axis
- Left button in the bottom half of the window will rotate the camera in a positive angle about the **x** axis.
- Right button in the top half of the window will zoom in (shrink the sphere centered at the center of interest, that the camera operates on).

- Right button in the bottom half of the window will zoom out (increase the size of the sphere that the camera operates on).



Camera Control

○ Pan

● Examine

Offset of actual center of interest from the local coordinate system origin (components in local coordinates)

0 | 0 | 0

☑ Set Link Axes as the Center of Interest

☐ Update camera relative to viewport frame

The settings on the Camera Control dialog box can now be easily understood. The top two radio buttons switch between the Pan and Examine Views discussed above. The user has the option to specify an offset from the object that is the center of interest for the camera to be focused on. This offset is in terms of the coordinate system that is specified as the center of interest. Note that when in Pan View, the center of interest offset is continuously adjusted so that the center of interest remains in the center of the viewport frame. Next is a checkbox that enables the user to set the inertial axes as the center of interest. Finally, there is a checkbox that sets whether the changes are made relative to the viewport (also referred to as camera) coordinate frame. If that is not checked, the changes will be relative to the link coordinate frame.

Note: Whenever a model is opened, the default center of interest is the link coordinate frame. If the center of interest is ever changed, the only way to make the link coordinate frame the center of interest again is to choose the setting in the Camera Control dialog box.

Note: When RobotModeler *starts, the default camera control is set to* `Examine View` *relative to the link coordinate frame with the link coordinate frame being the center of interest.*

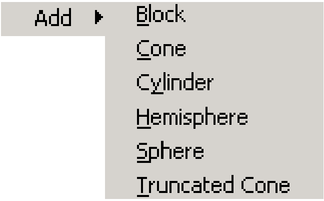
Camera Light

Toggles the light that points in the direction of the camera, on or off.

Note: See Section 6's discussion on lighting.

2.4 Primitives Menu

The "Primitives" menu provides the user with the ability to add new primitives to the current link model. This menu only has one menu item named "Add". When selected, this opens a submenu with a list of primitives.



Block

Adds a block primitive.

Note: For details about the block primitive, see the Block discussion in Section 4.2.

Cone

Adds a cone primitive.

Note: For details about the cone primitive, see the Cone discussion in 4.2.

Cylinder

Adds a cylinder primitive.

Note: For details about the cylinder primitive, see the Cylinder discussion in 4.2.

Hemisphere

Adds a hemisphere primitive.

Note: For details about the hemisphere primitive, see the Hemisphere discussion in 4.2.

Sphere

Adds a sphere primitive.

Note: For details about the sphere primitive, see the Sphere discussion in 4.2.

Truncated Cone

Adds a truncated cone primitive.

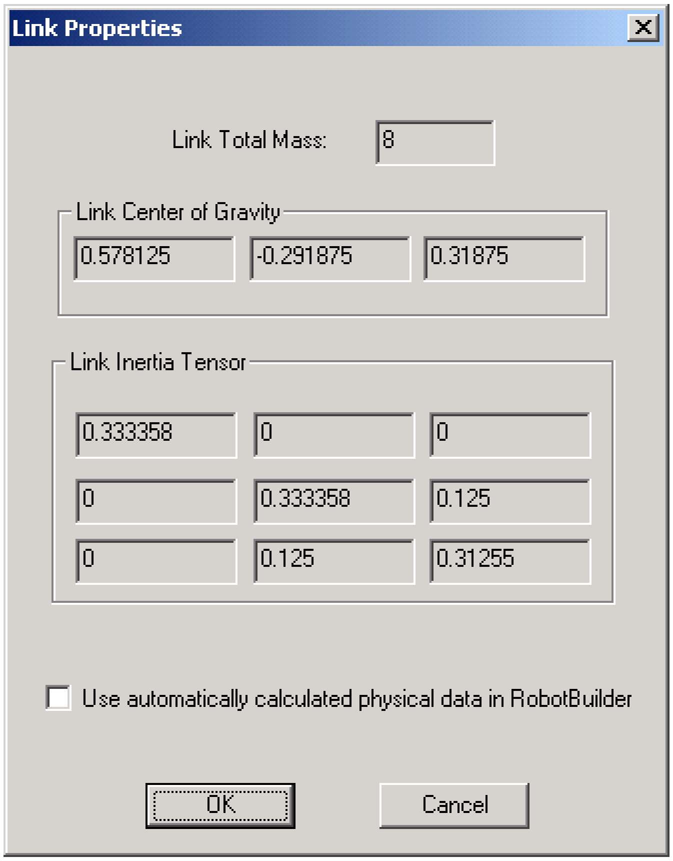
Note: For details about the truncated cone primitive, see the Truncated Cone discussion in 4.2.

2.5 Properties Menu

The "Properties" menu enables the user to see the physical properties that are automatically calculated for the link.

Physical Properties...

Physical Properties...



Link Properties

Link Total Mass: 8

Link Center of Gravity

0.578125	-0.291875	0.31875

Link Inertia Tensor

0.333358	0	0
0	0.333358	0.125
0	0.125	0.31255

☐ Use automatically calculated physical data in RobotBuilder

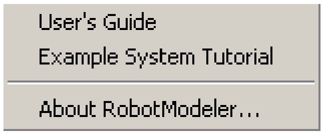
Opens the `Link Properties` dialog box that displays the approximate physical properties automatically calculated by RobotModeler. The total mass, center of gravity, and link inertia tensor are read-only. The link mass is the sum of the masses of the individual primitives. The center of gravity is the weighted average of the primitive mass centers. The inertia tensor is the sum of the translated inertias of each primitive. The checkbox at the bottom sets whether the automatically calculated physical parameters should be used in RobotBuilder or not. If it is checked and the model is used as a model for a RobotBuilder link with rigid body parameters, the fields in RobotBuilder will be read-only, and it will extract the mass, center of gravity, and inertia tensor from the RobotModeler model.

Note: The calculated inertia does not take into account overlapping primitives. If two primitives overlap, the overlap region will be considered twice as dense. One way to work around this is to set the mass very low (but not 0 because the validation procedure *prevents this) and the effect of the inertia of the primitive with low mass will be negligible.*

Note: The setting for whether to use the automatically calculated values in RobotBuilder or not, is in RobotModeler to make the consistency ramifications more intuitive. For example, if it was in RobotBuilder, it is not clear if the setting should be applied to just the use of the model in the current articulation or in all articulations.

2.6 Help Menu

The "Help" menu enables the user to get help and see general information about the application.



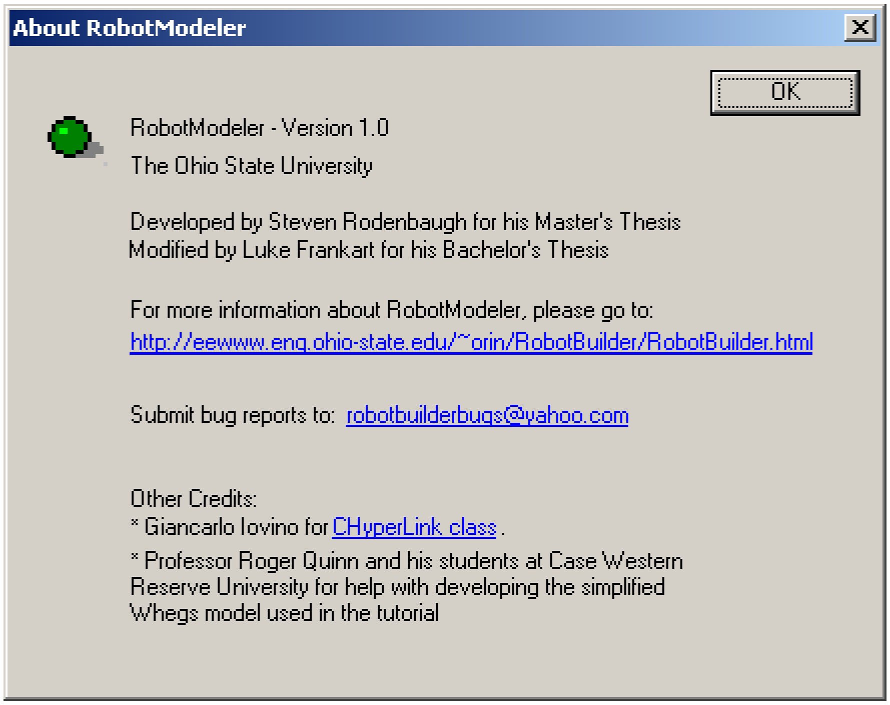
User's Guide

Opens the RobotModeler User's Guide (this document) with the PC's `.pdf` viewer (probably *Adobe Reader*). The document is located in the `Docs` subdirectory.

Example System Tutorial

Opens the RobotBuilder Example System Tutorial with the PC's `.pdf` viewer (probably *Adobe Reader*). The document is located in the `Docs` subdirectory.

About RobotModeler...


About RobotModeler

RobotModeler - Version 1.0

The Ohio State University

Developed by Steven Rodenbaugh for his Master's Thesis
Modified by Luke Frankart for his Bachelor's Thesis

For more information about RobotModeler, please go to:
http://eewww.eng.ohio-state.edu/~orin/RobotBuilder/RobotBuilder.html

Submit bug reports to: robotbuilderbugs@yahoo.com

Other Credits:
* Giancarlo Iovino for CHyperLink class.
* Professor Roger Quinn and his students at Case Western Reserve University for help with developing the simplified Whegs model used in the tutorial

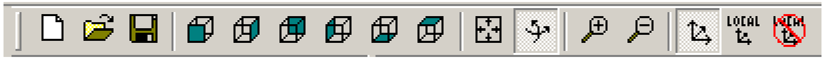
Opens the About RobotModeler dialog box. This dialog displays the version number, and a website to get more information.

Note: Although the version is listed as 1.0, the application has not been extensively tested in different environments. Please send a bug report to the email address in the dialog if a problem is encountered.

Note: When submitting a bug report, please give as much information as possible. Please send the application version number, your OS version, your video card and driver version, the problem that is being experienced, and the steps to reproduce that problem.

3. Toolbar

The toolbar provides one-click access to many commonly used functions in RobotModeler. The toolbar is located below the menu bar, and can be seen below:



3.1 File Functions

These three buttons provide shortcuts for menu items on the "File" menu.

New

Clears the current model and starts with an empty project.

Open 📂

Opens a dialog to find a `.rbm` file to open. See Section 2.1 – “Open” for details.

Save 💾

Saves the current model to a file. See Section 2.1 – “Save” for details.

3.2 View Functions

These buttons enable the user to easily change the position and orientation of the camera to predefined points. These buttons will orient the camera to be perpendicular to a particular plane indicated by the button and position the camera on the side of the model indicated by the button. The camera is then zoomed in or out. The zoom function will place the camera so that the model is wide enough to fit in the window, and the window will be centered on the center of interest (if applicable).

Note: To clarify, the model might not completely appear in the window, because the camera will place the center of interest at the center.

Note: The zoom setting will likely change to fit the model in the window after pressing one of these buttons.

Front View

Moves the camera to look at the "front" of the model (camera coordinate **x-y** plane is parallel to link coordinate **x-z** plane and camera looks toward the link positive **y** direction).

Right View

Moves the camera to look at the "right" side of the model (camera coordinate **x-y** plane is parallel to link coordinate **y-z** plane and camera looks toward the link negative **x** direction).

Back View

Moves the camera to look at the "back" side of the model (camera coordinate **x-y** plane is parallel to link coordinate **x-z** plane and camera looks toward the link negative **y** direction).

Left View

Moves the camera to look at the "left" side of the model (camera coordinate **x-y** plane is parallel to link coordinate **y-z** plane and camera looks toward the link positive **x** direction).

Bottom View

Moves the camera to look at the “bottom” side of the model (camera coordinate **x-y** plane is parallel to link coordinate **x-y** plane and camera looks toward the link positive **z** direction).

Top View

Moves the camera to look at the “top” side of the model (camera coordinate **x-y** plane is parallel to link coordinate **x-y** plane and camera looks toward the link negative **z** direction).

3.3 Camera Control Functions

These buttons enable the user to easily switch between the `Pan View` and `Examine View`.

Pan View

Switches the camera control to Pan View. Clicking this button is the same as selecting the Pan View radio button in the Camera Control dialog box.

Examine View

Switches the camera control to Examine View. Clicking this button is the same as selecting the Examine View radio button in the Camera Control dialog box.

3.4 Zoom Functions

These buttons enable the user to easily zoom the camera in or out.

Zoom In

Moves the camera closer to the center of interest.

Zoom Out

Moves the camera further away from the center of interest.

3.5 Axes Visibility Functions

RobotModeler has the ability to show the direction of the axes of the link and primitive coordinate frames by using a 2D overlay. This implies that the axes will always appear on top of the models, but this enables the user to easily see the position and orientation of the axes. The axes are also labeled with **x**, **y**, and **z** for clarity. These buttons show or hide the link coordinate frame and the axes of the individual primitive frames.

Note: The axis labels are by default displayed with an Microsoft Sans Serif (size 9) font, specified in the `font.wtk` *file found in the same directory as* `RobotModeler.exe`*. If the labels appear bulky, this file may be missing, the font may not be installed on the system, or the working directory may not be properly specified when the application is started.*

Note: The axes of the primitives can be toggled on or off individually by the context menu in the Primitive List View (see Section 4.2).

Toggle Link Axes

This button is stateful. When in the down position, the link axes will be drawn. When in the up position, the link axes will not be drawn.

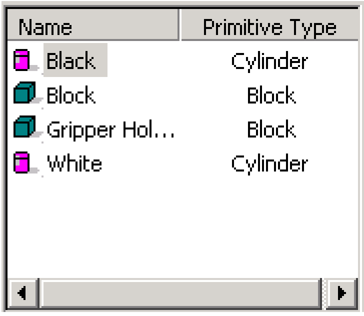
Show All Local Primitive Axes

When pressed, the axes of all primitives will be shown.

Hide All Local Primitive Axes

When pressed, the axes of all primitives will be hidden.

4. Primitive List View



Name	Primitive Type
Black	Cylinder
Block	Block
Gripper Hol...	Block
White	Cylinder

The Primitive List View shows a list of primitives that are in the current model. There are two columns in the list. The first column is labeled “Name” and includes the user-settable name given to the primitive, preceded by a small icon indicating the type of primitive. The second column, labeled “Primitive Type”, explicitly states the type of primitive. The two columns can be resized. The primitive list is always kept in alphabetical order by primitive name. If the list gets too long, scrollbars will be automatically added. In addition, more space can be given to the window by adjusting the splitter window that contains the list view.

Note: As with most lists in Windows, if the column separator in the label line is double clicked, the column to the left will expand in size so that all text is completely visible (within the size of the window).

When an item is selected by left clicking on the name, the primitive in the 3D Model View window will be denoted by a white bounding box. This box can be cleared by selecting another primitive or clicking in the list view in an area below the names of the primitives.

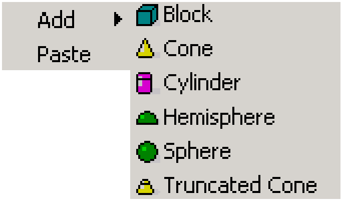
Note: See Section 6 for more information about the 3D Model View window.

4.1 Context Menu

The context menu is accessed by right clicking a selected item, a group of items, or no items. The context menu varies depending on what it selected.

4.1.1 No Items Selected Context Menu

When no items are selected, the context menu has two options.



Add

When “Add” is selected, a submenu opens up with all of the available primitives. Select a primitive to add it to the model.

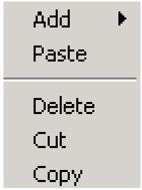
Paste

Pastes one or more primitives from the buffer. This is only enabled if there is one or more primitives in the buffer.

Note: See Section 2.2 for more information about the buffer.

4.1.2 Multiple Items Selected Context Menu

When more than one item is selected, the context menu will have options that can be applied to multiple items.



Note: A number of the menu items are related to the buffer. See Section 2.2 for more information about the buffer.

Add

When selected, a submenu will open that will list the primitives that can be added.

Paste

Enabled when one or more items are in the buffer, and when selected, will add them to the primitive list.

Delete

Deletes the selected primitives.

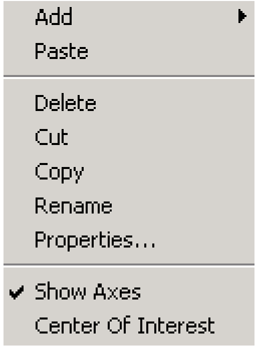
Cut

Removes the selected primitives and places them in the buffer.

Copy

Places a copy of the selected primitives in the buffer.

4.1.3 One Item Selected Context Menu



When one item is selected, the context menu will have the following options:

Note: A number of the menu items are related to the buffer. See Section 2.2 for more information about the buffer.

Add

When selected, "Add" will open a submenu that will allow the user to choose a new primitive to add.

Paste

Enabled when one or more items are in the buffer, and when selected, will add them to the primitive list.

Delete

Deletes the selected primitive.

Cut

Removes the selected primitive and places it in the buffer.

Copy

Places a copy of the selected primitive in the buffer.

Rename

Choose to rename the primitive. When selected, the primitive name in the Primitive List View will become editable. After typing a new name, press ‘Enter’ or click anywhere in the application outside of the edit area. The list will automatically be resorted into alphabetical order.

Show Axes

Toggles the visibility of the axes of the current primitive on or off. This item will be preceded with a check mark if the primitive axes are currently displayed.

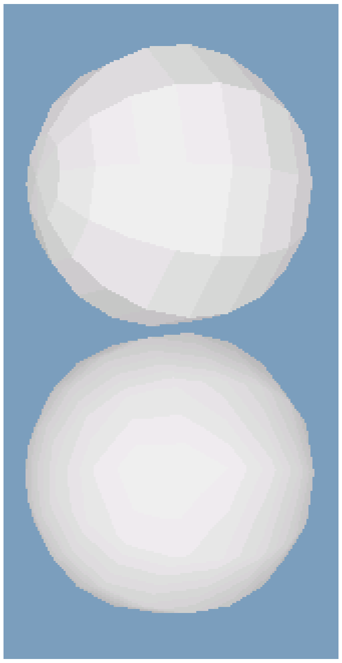
Note: See Section 3.5 for more information about axes.

Center of Interest

Sets the current primitive as the center of interest. This item will be preceded with a check mark if the primitive is already the center of interest. *Note: See Section 2.3 for more information about camera control.*

Properties...

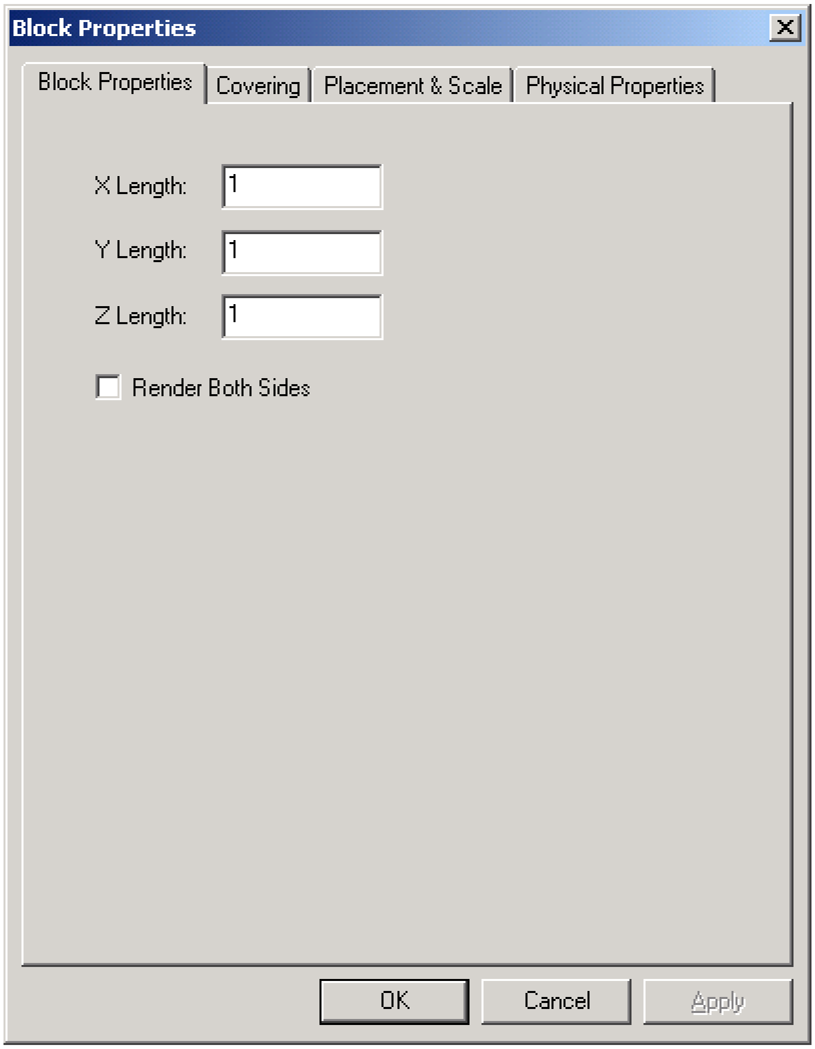
Opens a property sheet for the selected primitive. The property sheet enables the user to view and edit settings for the primitive. The first page of the property sheet always contains data about the specific type of primitive.



Some settings are common to several primitives. Most primitives have an option whether to render the backfaces. The backfaces are inside the primitive so they are normally not rendered for more efficient graphics. The user may want to enable them so that when the camera is zoomed inside a primitive, its faces can be visible from the inside too. Primitives with curved surfaces have a setting for enabling Gouraud shading. Gouraud shading attempts to visibly smooth the polygonal surfaces by computing the resultant color at the vertices of the faces. This method is more computationally intensive, but it makes curved features appear more smooth. In the picture to the right, the top sphere was drawn without Gouraud shading, and the bottom sphere was drawn with it.

4.2 Primitives

Block



X Length: 1

Y Length: 1

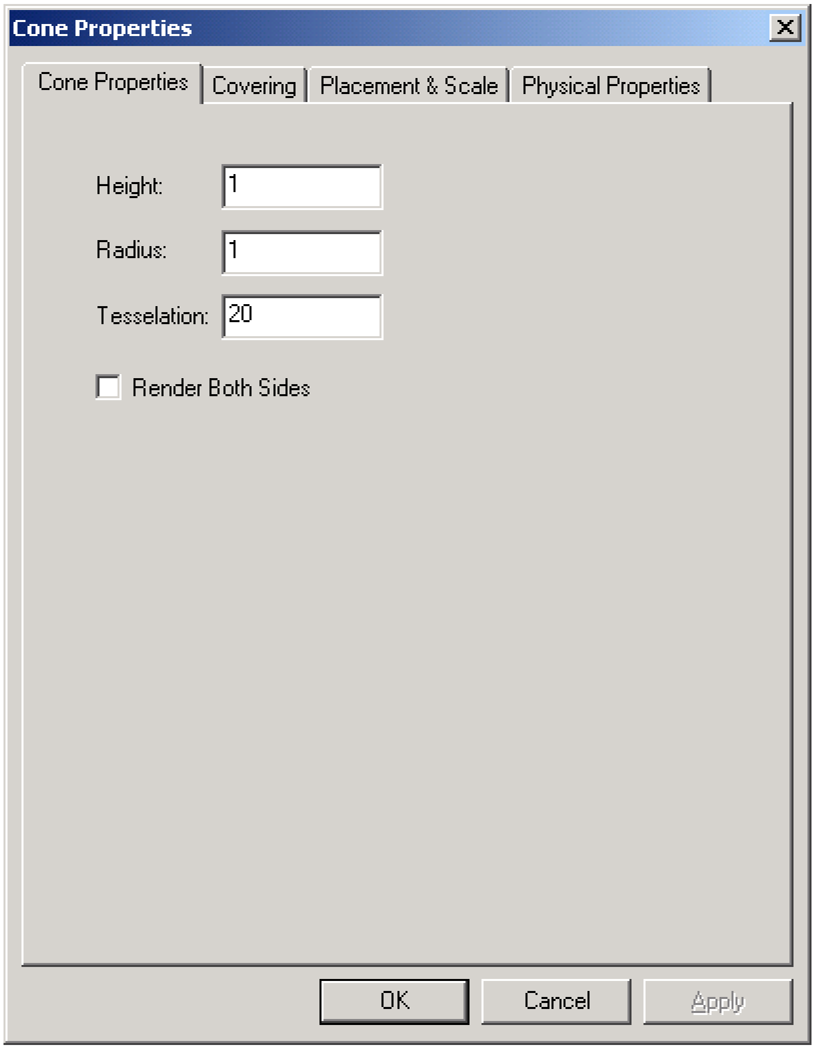
Z Length: 1

□ Render Both Sides

The Block Properties sheet displays properties of the block primitive. The user can set the length in the **x**, **y**, and **z** directions. The user can also choose to render both sides. The default values are shown.

The coordinate frame is at the center of the block.

Cone



Height: 1

Radius: 1

Tesselation: 20

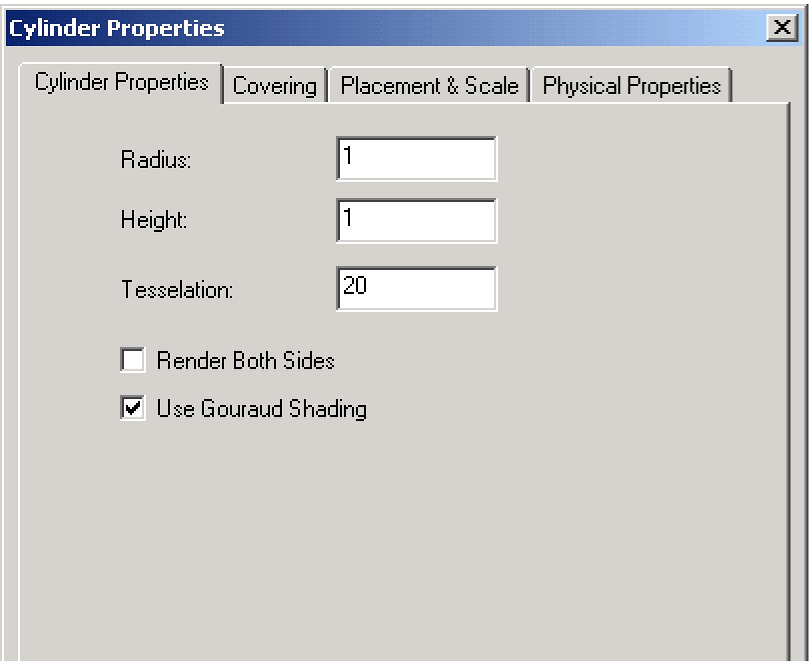
□ Render Both Sides

The Cone Properties sheet displays properties of the cone primitive. The user can set the height (along the **y** axis) and the radius of the base. The tesselation setting specifies how many triangles to use to create the surface (not counting the base). Consequently, a tesselation of three would create a triangular pyramid. The user can also choose to render both sides. The default values are shown.

The coordinate frame is at the midpoint between the base and the tip, with the cone coming to a point along the –**y** axis.

Note: A tesselation greater than 256 will cause the bottom face not to render.

Cylinder



Radius: 1

Height: 1

Tesselation: 20

☐ Render Both Sides

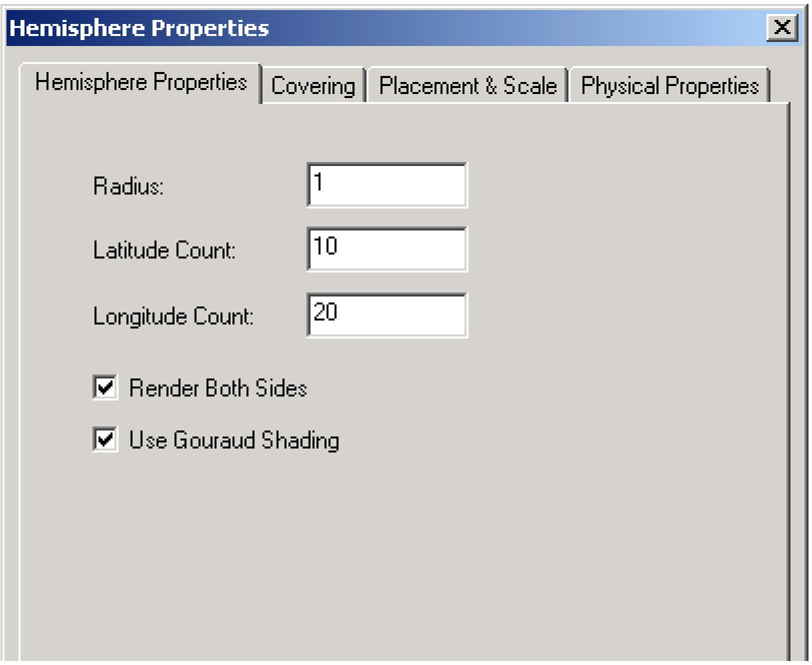
☑ Use Gouraud Shading

The Cylinder Properties sheet displays properties of the cylinder primitive. The user can set the height, (along the **y** axis) and the radius. The tesselation setting specifies how many polygons to use to create the surface (not counting the bases). The user can also choose to render both sides and enable Gouraud shading. The default values are shown.

The coordinate frame is at the midpoint of the cylinder.

Note: A tesselation greater than 256 will cause the bottom face not to render.

Hemisphere



Hemisphere Properties

Radius: 1

Latitude Count: 10

Longitude Count: 20

☑ Render Both Sides

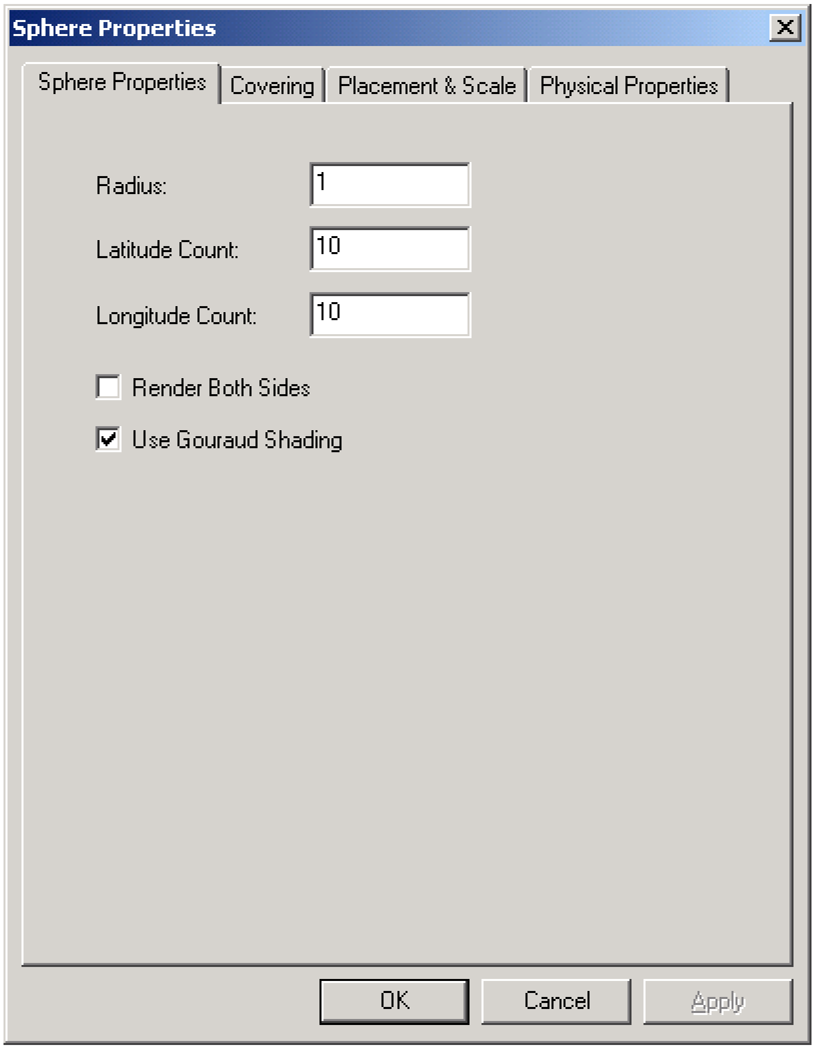
☑ Use Gouraud Shading

The Hemisphere Properties sheet displays the properties of the hemisphere primitive. The user can set the radius, the latitude count (number of partitions parallel to the **x-z** plane), and the longitude count (number of equal sized partitions along the **y** axis, and therefore the number of sides on the base). The user can also choose to render both sides and enable Gouraud shading. The default values are shown.

The coordinate frame is centered at the base of the hemisphere (**x-z** plane) when not scaled.

Note: The hemisphere renders as a shell with no base. Consequently, if "Render Both Sides" is disabled, it will be invisible when viewed from the base. Even though it is drawn as a shell, the physical parameter calculations assume it is a uniformly dense hemispherical volume.

Sphere



Radius: 1

Latitude Count: 10

Longitude Count: 10

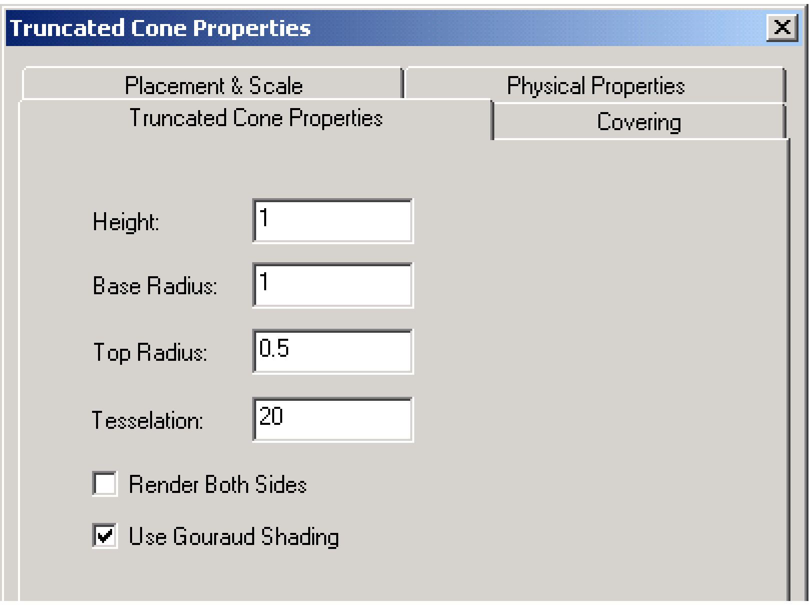
□ Render Both Sides

☑ Use Gouraud Shading

The Sphere Properties sheet displays properties of the sphere primitive. The user can set the radius, the latitude count (number of partitions parallel to the **x-z** plane), and the longitude count (number of equal sized partitions along the **y** axis). The user can also choose to render both sides and enable Gouraud shading. The default values are shown.

The coordinate frame is located at the center of sphere with the poles located along the **y** axis.

Truncated Cone


Truncated Cone Properties

Height: 1

Base Radius: 1

Top Radius: 0.5

Tesselation: 20

□ Render Both Sides

☑ Use Gouraud Shading

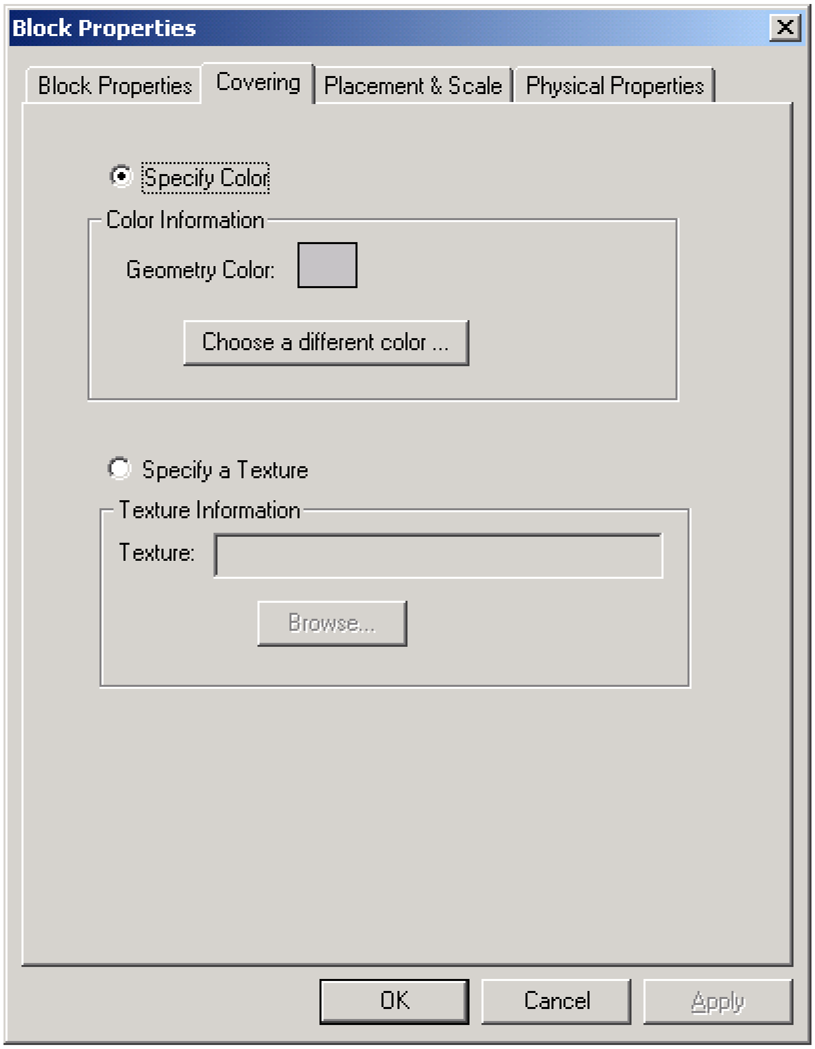
The Truncated Cone Properties sheet displays properties of the truncated cone primitive. A truncated cone is a regular cone with the point removed. The user can set the height, the radius of the top, the radius of the base, and the tesselation count. The user can also choose to render both sides and enable Gouraud shading. The default values are shown.

The coordinate frame is centered in the truncated cone and is located at the midpoint between the top and the base, which is along the **y** axis. The top intersects the negative **y** axis and the base intersects the **y** axis.

Note: There is no requirement that the top radius be smaller than the bottom.

Note: A tesselation greater than 256 will cause the base faces to not render.

Covering Property Page



● Specify Color

Color Information

Geometry Color: ▨

○ Specify a Texture

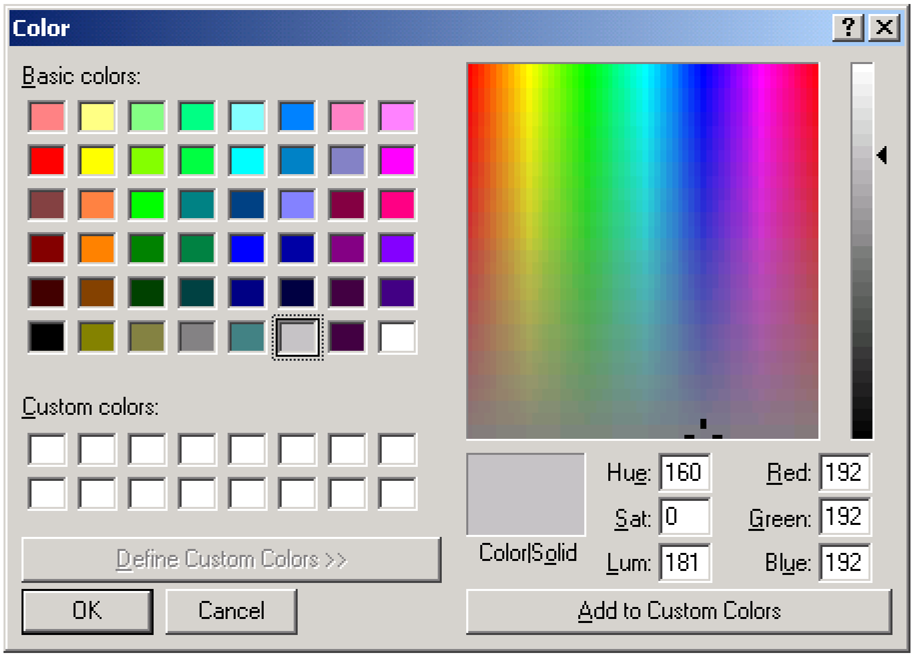
Texture Information

Texture:

A covering can be specified for each primitive. The covering is either a solid color, or a texture. When the “Choose a different color …” button is selected, the color dialog below appears. To specify a texture, choose a `.jpg` file after pressing the “Browse” button.

Note: The color specifies the diffuse and ambient reflective properties for the material. The actual resulting color will depend on the lighting conditions.

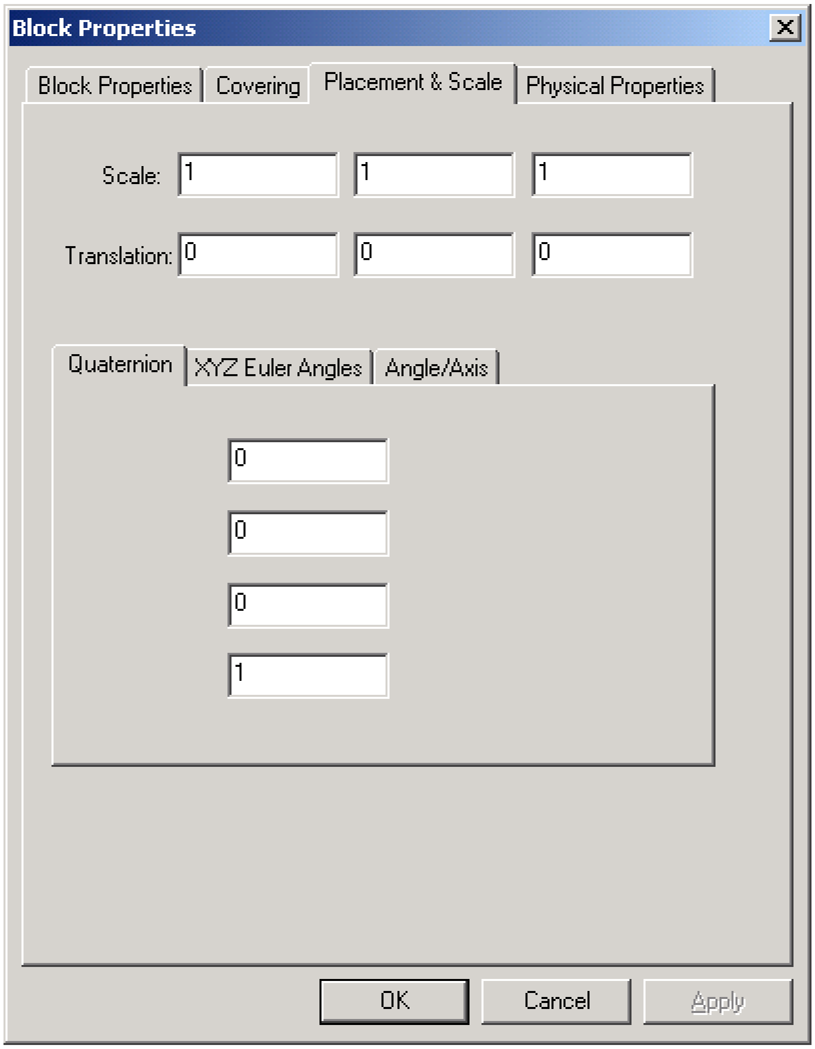
Note: RobotModeler *locates texture files by following a search method. If an error message indicates that the specified texture cannot be found, please see Section 7.2 for more information.*



Hue: 160
Sat: 0
Lum: 181
Red: 192
Green: 192
Blue: 192

Note: The color dialog lets the user specify "Custom Colors" to save a color for the current instance of the application. To specify a "Custom Color", choose the custom color square to set (otherwise the default will overwrite the first square), adjust the RGB or HSL as desired, and click "Add to Custom Colors" button.

Placement & Scale Property Page



Scale: 1 1 1

Translation: 0 0 0

0

0

0

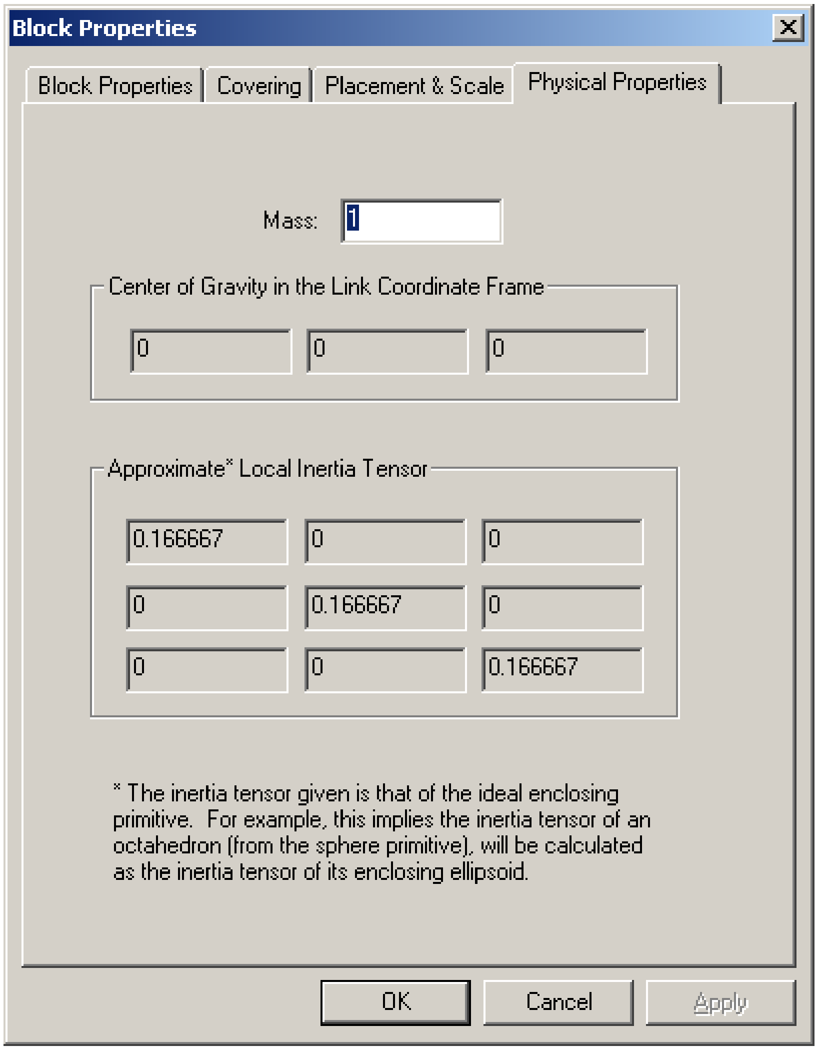
1

The scale, position, and orientation can be specified for every primitive with the Placement & Scale property page. The scale is the absolute scale to be applied along the local primitive axes, the translation is the position of the primitive's coordinate frame in terms of the link coordinate frame, and the rotation follows the right-handed convention from the link coordinate frame to the primitive coordinate frame. The rotation is presented and can be edited in three different formats: quaternion, XYZ Euler angles, and angle/axis format [1].

Note: RobotModeler internally stores rotation in a three by three matrix. When switching between the different rotation representations, the current representation is derived from the stored three by three rotation matrix. Consequently, small round-off errors may appear when switching between rotation representations.

Physical Properties Property Page

This property page displays the physical properties of the selected primitive. The mass of the primitive is editable by the user and is assumed to be uniformly distributed in the primitive. Using the mass, the type of primitive, and the position of the primitive, RobotModeler can approximate the center of gravity and the local inertia tensor. This data is combined for each primitive to find values for the complete link model to be used in RobotBuilder. Note that the inertia is approximated by calculating the ideal enclosing primitive. The default mass for every primitive is one.



Mass: 1

Center of Gravity in the Link Coordinate Frame

0	0	0

Approximate* Local Inertia Tensor

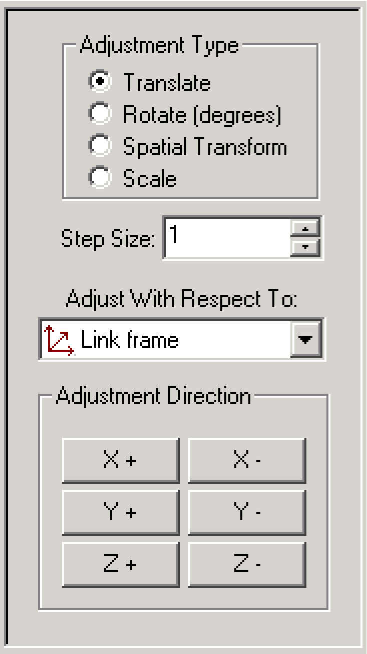
0.166667	0	0
0	0.166667	0
0	0	0.166667

* The inertia tensor given is that of the ideal enclosing primitive. For example, this implies the inertia tensor of an octahedron (from the sphere primitive), will be calculated as the inertia tensor of its enclosing ellipsoid.

Note: See Section 2.5 to learn how to override the automatically calculated data.

Note: See Section 7.1 for a discussion on the units of the data.

5. Modification Pallet


Adjustment Type
● Translate
○ Rotate (degrees)
○ Spatial Transform
○ Scale

Step Size: 1

Adjust With Respect To: Link frame

The modification pallet enables the user to modify one or more primitives easily and immediately see the result in the 3D Model View window. The user can incrementally change the position, rotation, and scale.

First, the user chooses one or more primitives from the Primitive List View to apply to the modification. Next the adjustment type is selected. The step size of the modification, which must be positive, is entered next. Then the frame that the modification is relative to is selected from the combo box. Scaling is an exception and can only be applied relative to the local primitive frame. Finally, the relative axis is selected. Since there are both positive and negative buttons, a negative translation is achieved by entering a positive step size, then selecting the negative axis.

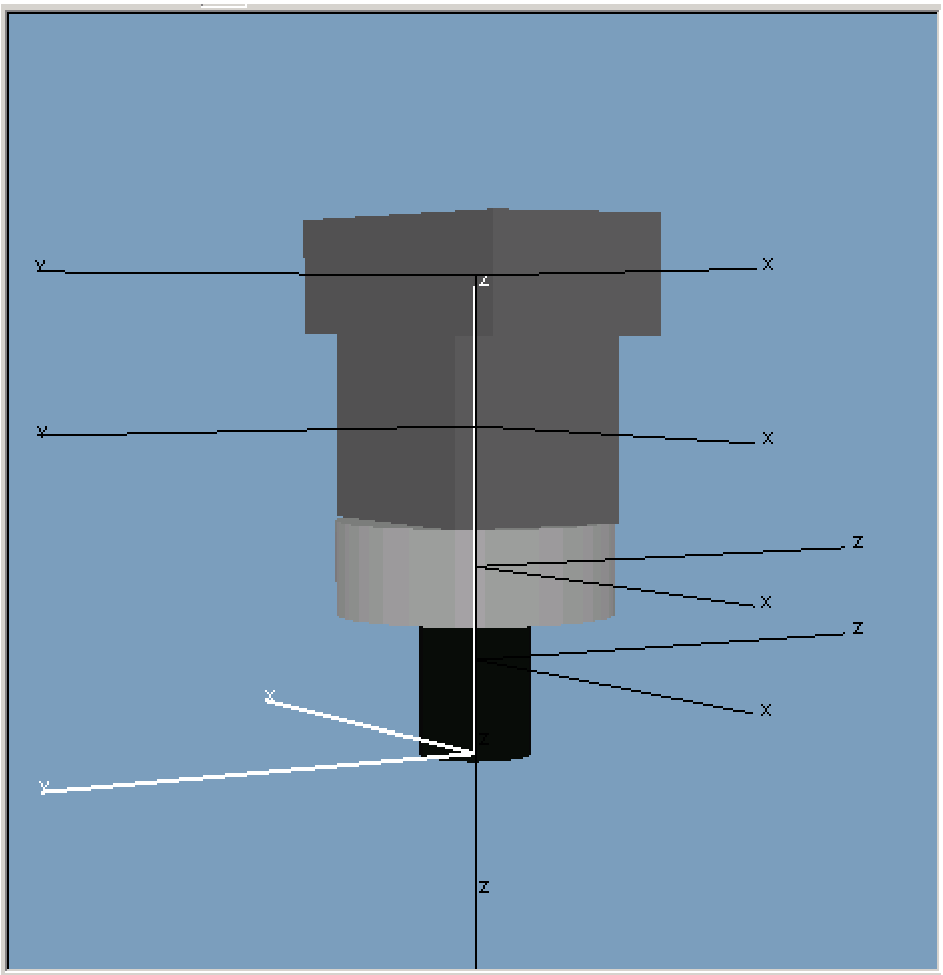
Note: If desired, the absolute values for the scale, position, and rotation can be seen and edited in the `Placement & Scale` *property page accessed by selecting a primitive in the Primitive List View and selecting "Properties..." from the context menu.*

Note: A spatial transformation relative to the local frame is the same as a rotation relative to the local frame.

*Note: When a negative axis button is clicked for a scale modification, the step size will be reciprocated. Therefore, a step size of two applied to the negative **x** axis will shrink the primitive in half along the **x** axis.*

6. 3D Model View

The 3D Model View displays a 3D representation of the current link model.



The colors that result in the 3D Model View are due to the material properties and the lighting used. By default, there is a directed light that points in the direction of the camera. This light is a white light at 30% intensity. In addition, the scene has an ambient white light at 95% intensity that illuminates all faces evenly.

Note: For information about navigation in the window, see Section 2.3.

Note: For information about toggling the visibility of the link and primitive axes, see Section 3.5.

7. General Topics

7.1. Units

RobotBuilder, and by association, RobotModeler, are written so that the units are at the discretion of the user. There is not a specific place to state the units, but instead they are implied by the data entered. It is up to the user to verify that the entered data uses consistent units.

7.2. File Search Method

When RobotModeler needs to locate a file, it follows a specific search pattern to find it. This search method will be used when textures have to be loaded to apply to models. It is important to understand the method, in order to understand if the file specified will be able to be found in the future.

There are two ways a reference to a texture can be saved: the filename or the full path and filename. The first way is useful because it allows projects to be portable to different directories and machines with different configurations. The difficulty is successfully locating the file because it may exist in a number of places.

When RobotModeler needs to locate a texture by just a filename, it first checks the directory where the `.rbm` file of the current link is located. If the texture is not found, the working directory of the application is checked. The working directory is specified by the operating system and should be the directory where the RobotModeler executable is located. It is not recommended to place files here, but it is searched for completeness. Finally, a subdirectory named `Graphics` under the working directory is searched.

If the texture cannot be found using the search method, and the reference is not a full path, RobotModeler will not be able to find the file. Upon failure, an error message will be presented to the user.

8. Conclusion

Hopefully, RobotModeler will be helpful in your research. If you discover any bugs, please send an email to robotbuilderbugs@yahoo.com.

BIBLIOGRAPHY

[1] John J. Craig, *Introduction to Robotics: Mechanics and Control, 2nd Edition*, Addison-Wesley Publishing Company, Reading, Massachusetts, 1989.