# RoboSim User's Guide

**Version 0.5.0**

# Contents

# 1 Introduction

`RoboSim` is a robot simulation environment, developed by the UC Davis C-STEM Center, for programming Barobo Mobots and Linkbots. Almost any program that can control hardware Barobo robots, except for programs using a controller robot, can be used to run virtual robots in RoboSim without any modification. Also any program that can control virtual robots in RoboSim, except for the coordinate system programs in Appendix B, can be run on hardware Barobo robots without any changes.

# 2 RoboSim GUI

The RoboSim graphical user interface (GUI), shown in Figure 1, allows the user to change between hardware and virtual robots when a Ch robot program is executed. There is no save button within the GUI, all changes are automatically saved on each click.

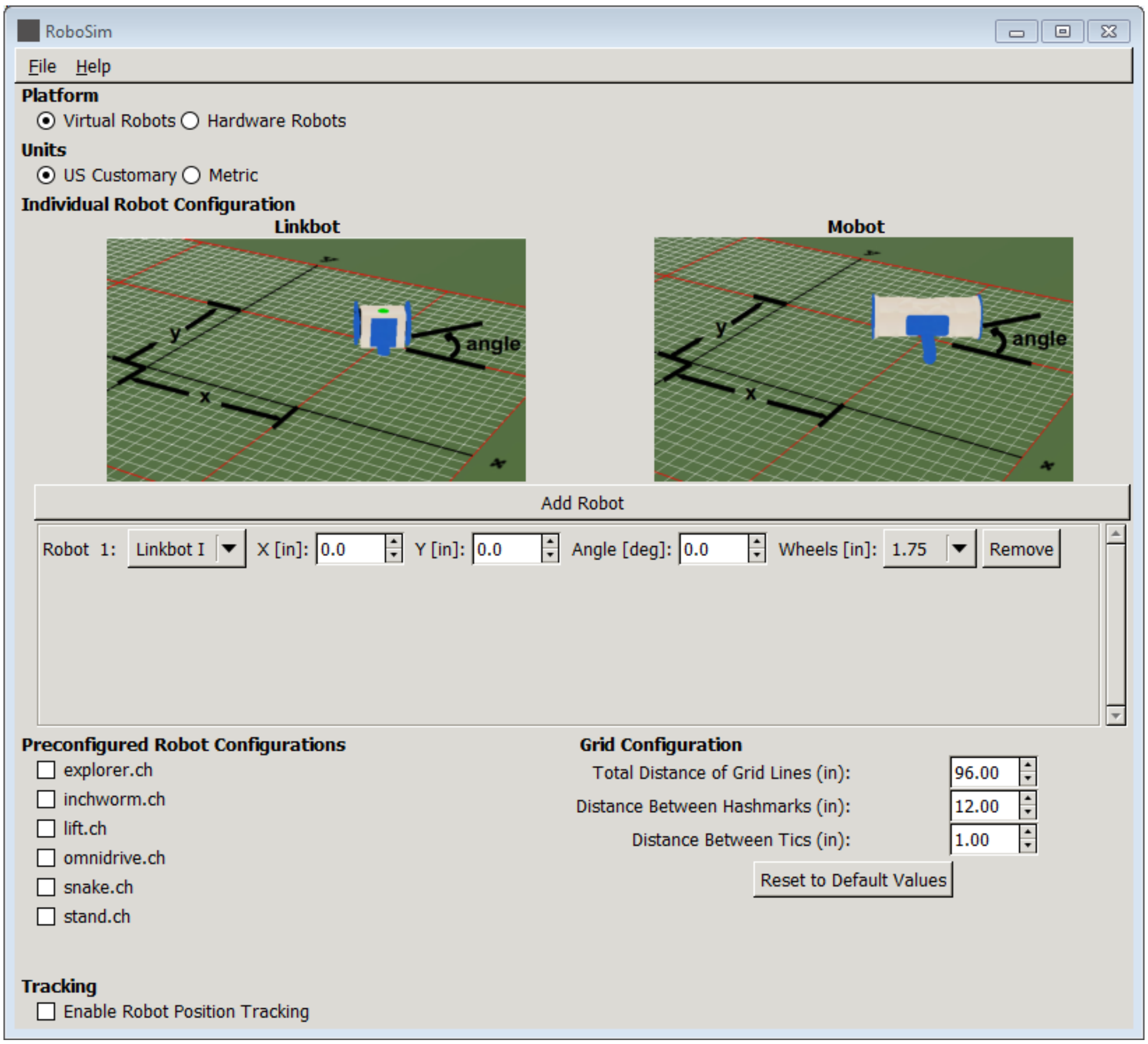


Figure 1: The RoboSim GUI.

## 2.1 Platform

This lets a user pick whether the output is sent to the virtual or hardware robots. Each time a new Ch code is started, it will check this variable. Therefore to run subsequent trials on simulated and hardware robots, run the Ch code with the 'Virtual Robots' button checked. Close the code, change to 'Hardware Robots' and then rerun the same code. It will now output to the physical robots.

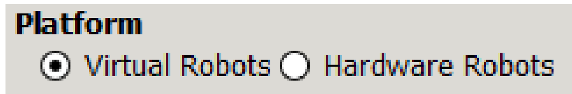


Figure 2: Initial robot configuration dialog.

## 2.2 Units

Simulations within RoboSim can be run either in US Customary units consisting of inches, degrees, and seconds or Metric units with centimeters, degrees, and seconds. Changing units will effect the grid spacing drawn beneath the robots and the spacing between robots. Changing between these two options will change the labels within the GUI to indicate the units being used.

## 2.3 Individual Robot Configuration

Initial robot configurations can either be done through the individual robot section or the preconfigured options. The individual robot section has options to allow robots to be positioned within the simulated scene either with or without wheels but not attached to each other.

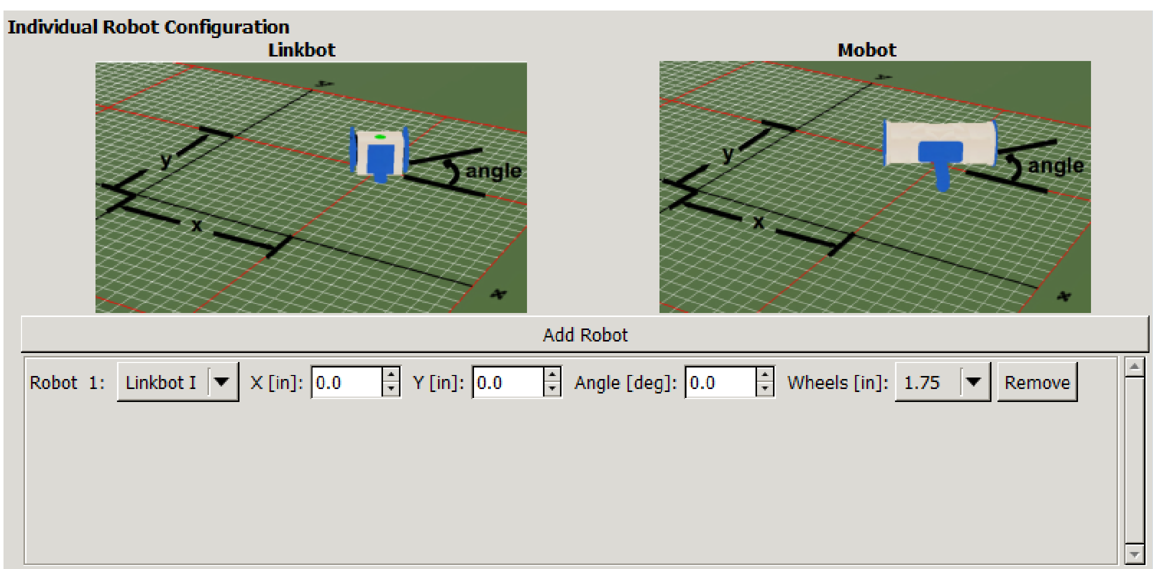


Figure 3: Individual robot configuration dialog.

Images for the Linkbot and Mobot showing the meaning of each of the options is shown above the configuration box. These are screenshots of the simulation program with the robots positioned at 1 foot in both the X and Y dimensions and at an angle of 30 degrees. These are the dimensions available to initially position robots within the simulation.

Initially the individual robot list is empty but can be populated by the large 'Add Robot' button below the configuration images. Clicking this button will add more and more robots into the scene each offset from the previous in the x-direction by 6 inches or 15 centimeters depending upon the units selected. The order within the robot list will be the order in which the robots will be read into the simulation program similar to how BaroboLink reads the list of available hardware robots.

### 2.3.1 Robot Type

There are three options for robot type available. Linkbot-I, Linkbot-L, and Mobot. The options are presented in a drop down menu as shown in Figure 4.

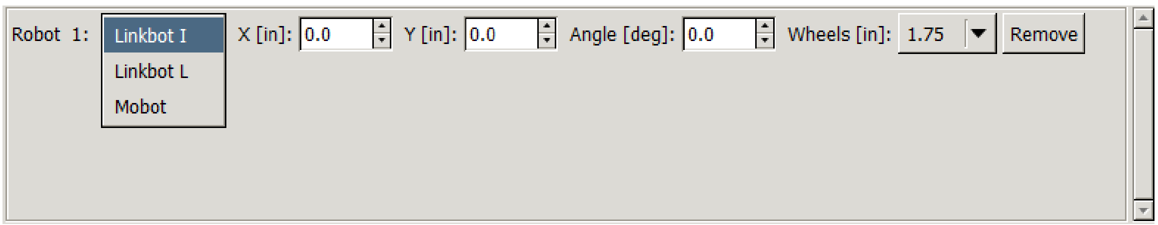


Figure 4: Picking a robot type.

### 2.3.2 Robot Position

Both X and Y positions can be chosen independently for each robot.

### 2.3.3 Robot Angle

The rotation angle from the x-axis can be used for changing the orientation of the two robots respective to each other.

### 2.3.4 Wheels

Since so many times the robots are run with wheels and a caster connected, a drop down menu is provided to select different wheel sizes. The options listed are the radii of the wheels provided with Linkbots when purchased from Barobo. Each wheel is drawn with a series of dots along the one radius to easily show the rotation of the wheel. The correlation between wheel radius and number of dots is given in Table 1.

| Number of Dots | Wheel Radius |
|---|---|
| 3 | 1.625 inch |
| 4 | 1.75 inch |
| 5 | 2.00 inch |

Table 1: Wheel sizes and number of dots.

### 2.3.5 Remove

Each robot can be removed from the scene by clicking the remove button. All remaining robots will remain within the simulation and be at the same positions.

## 2.4 Preconfigured Robot Configurations

In addition to positioning robots independently within the scene, six preconfigured options are available to users which represent common Linkbot configurations. Selecting one of the options will display a picture of the configuration built with the hardware Linkbots and corresponding to code within the C-STEM Learn Linkbot book. When one of these options is selected, the specific configuration for this setup is passed into Ch and any robots in the individual robot list are ignored. To switch back to the individual configuration, just unselect the selected robot configuration.

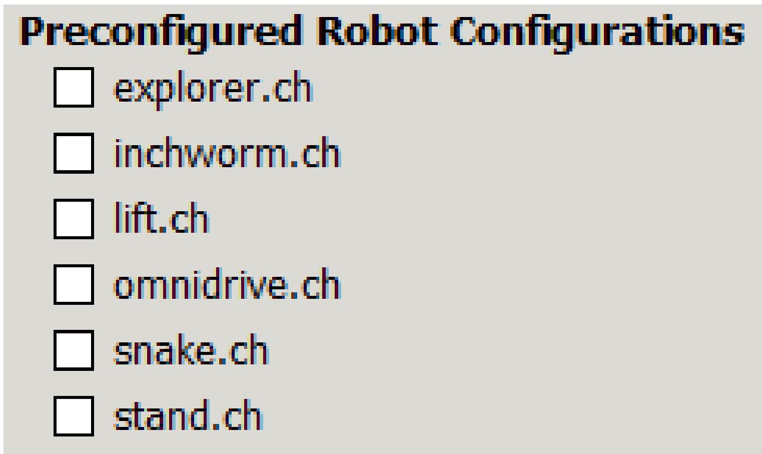


Figure 5: Preconfigured Linkbots.

## 2.5 Grid Configuration

To be able to see how far robots have moved, a grid is enabled under the robots. There are three options to alter the layout of the grid lines. Total distance is the entire distance between -x and x for which grid lines will be displayed. Hashmarks are the red lines drawn within the configuration images. By default they are every foot. Tics are the most frequent lines drawn in a light gray and by default every inch. Switching between US Customary and Metric units will change these default values to logical starting points for the metric system. At any point the defaults can be restored using the 'Restore to Default Values' button. It will restore to the defaults of the units selected. The defaults for US and Metric units are shown in Figures 6 and 7, respectively.

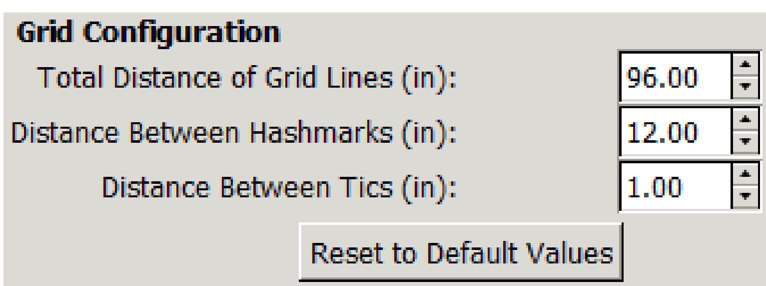


Figure 6: Default US Customary Grid Spacing.

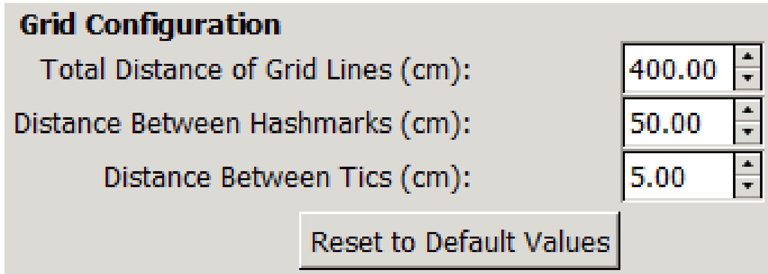


Figure 7: Default Metric Grid Spacing.

### 2.6 Tracking

Tracking where robots have been can be enabled by default by selecting the check box. This will draw green lines which draw out everywhere that each robot has been.

## 3 Running a Simulation

Once the simulation environment has been configured with the RoboSim GUI in Section 2, users can now run code written to control the hardware Barobo robots within simulation. The RoboSim GUI should remain open while simulating robots, once it is closed, it will revert to hardware mode. The graphics for each simulation are created upon running the code.

## 4 Interacting with a Simulation

The simulation window responds to mouse inputs to allow users to move about the scene. The mouse wheel or right mouse button allows zooming in and out from the robots. Holding the left button rotates the camera about the viewpoint. Clicking, holding, and dragging both mouse buttons pans around the scene to change the camera location.

The ground plane is for reference only. It is designed to disappear when viewing the robots from below to be able to inspect the movement from all angles.

### 4.1 Keyboard Input

The simulation will respond to keyboard input as outlined in Table 2.

Once a simulation is running, the tracking line can be enabled or disabled by pressing the 't' key. Since the robot can get in the way of viewing the tracked locations, the 'r' key enables the tacking line as well as makes the robot disappear. The motion robot is still moving around and completing its motion, however it is just not drawn onto the screen. Any other key on the keyboard will either pause or unpause the simulation depending upon the current state.

| key | action |
|---|---|
| r | toggle robot visibility and enable tracking |
| t | toggle robot tracking |
| any other key | Pause and unpause simulation |

Table 2: Keyboard input for RoboSim

### 4.2 Mouse Input

The mouse can be used to move the camera around the scene. Left clicking and dragging the mouse spins the camera around the fixed camera position. Right clicking and dragging enables scaling of the view. Clicking both left and right mouse buttons and dragging changes the location of the camera within the scene.

| **Button** | **Action** |
|---|---|
| Left | Spin around fixed camera point |
| Right | Scale simulation view |
| Both | Pan around scene |

Table 3: Keyboard input for RoboSim

Clicking on any robot will enable a pop up which displays the robot name and the current position of the robot. Clicking again will disable the display. The display will update automatically as the robot is moving.

# A Manual Configuration File Generation

## A.1 Robot Attributes

Each robot element is required to have one attribute titled **id** which is an unique identifier for the simulation to reference. A second optional attribute is **orientation** which orients the face of a second robot when it is being attached to a first robot.

| | |
|---|---|
| `<linkboti id="0"/>` | one linkbot I with id = 0 |
| `<linkboti id="0" orientation="3"/>` | Linkbot I is 'upside-down' |

Table 4: Examples

| **attribute** | **values** | **description** |
|---|---|---|
| id | unique integer | a unique integer to identify each robot |
| orientation | 1 | robot face number is at 12 o'clock |
| | 2 | robot face number is at 3 o'clock |
| | 3 | robot face number is at 6 o'clock |
| | 4 | robot face number is at 9 o'clock |

Table 5: Robot Attributes