USER MANUAL

Project: RoboSim

Department of Computing and Information Sciences
College of Engineering

KANSAS STATE UNIVERSITY
Manhattan, Kansas

2005

TABLE OF CONTENTS

Chapter 1 Installation Manual

RoboSim is a distributed system consisting of the following modules `Environment`, `GeometryClient`, 3D/2D `Viewer`, `RemoteControlRobot`, and `RemoteControl`. These modules require the following configurations to be installed before the simulator is started.

1. Check out the RoboSim project from the CVS repository. The CVS is located at the server `fingolfin.user.cis.ksu.edu`[1]. A login is necessary to check out.
2. Download the setup file for Java3D API with runtime environment version 1.4 or 1.5 from http://java.sun.com and install it on your system.
3. Download the joystick API version 0.6 (Filename: Joystick-0-6.zip) from the project "Joystick Driver for Java" from http://sourceforge.net/. Perform the following steps for installing it.
    a. Unzip the file to location say <JOY_PATH> in your system.
    b. Open the RoboSim project in Eclipse. Select Preferences from the Window menu. Expand Java option and Build Path option. Select Classpath Variables in it and create a new variable JOYSTICK_PATH pointing to the jar file <JOY_PATH>/joystick/Joystick.jar.
    c. Select the Properties option in the Project menu and select the Java Build Path option. In the Libraries tabbed pane, select the Add Variable to add the JOYSTICK_PATH variable created. Close it and rebuild the whole project.
4. There are two .bat files in <RoboSim installation dir>/RoboSim/scripts/windows/ namely Remote.bat and Maze2.bat. The Remote.bat starts the Remote Control application using the OpHo2_1.xml as the EnvironmentConfigFIle and Maze2.bat starts the Maze application using the Maze2.xml as the EnvironmentConfigFile. These xml files are present in the <RoboSim Workspace>/RoboSim/TestLoadFiles/Environment directory.

[1] Server will be moved to `macr.user.cis.ksu.edu` in future

Chapter 2 Starting and Running Application

After installing the system, the following can be used as a step-by-step procedure for starting the simulator. The simulator consists of a minimum of four modules that need to be started. They are the `Environment`, `GeometryClient`, 3D/2D `Viewer`, `RemoteControlRobot`, and `RemoteControl` started in the given order.

1. The `RoboSim\scripts\` directory contains the executable files for Windows and Linux based systems. They are written to load pre-defined environment configurations. Samples of the `EnvironmentConfigFile` are located in the `RoboSim\TestLoadFiles\Environment` directory (e.g. `OpHo1.xml`). The `RobotConfigFile` given to the `RemoteControl` is also located in this directory (e.g. `RobotInfo.xml`). Using these files as templates, you can create new files based on the applications specifications.

   The `Environment` is the first module that should be started. The file given as input is the Environment configuration file. The command is:

   ```
   java edu.ksu.cis.cooprobot.simulator.environment.Environment
                                                                <EnvironmentConfigFile>
   ```

2. The next two modules to be started are the 2D/3D `Viewer` and the `GeometryClient`. They can be started in any order using the following commands:

   ```
   java edu.ksu.cis.cooprobot.simulator.viewer.Viewer2D <EnvHostname>
                                                               <EnvWaitPort = 3000>

   java edu.ksu.cis.cooprobot.simulator.viewer.Viewer3D <EnvHostname>
                                                               <EnvWaitPort = 3000>

   java edu.ksu.cis.cooprobot.simulator.geometry.GeometryClient
                                                 <EnvHostname> <EnvWaitPort = 10000>
   ```

   In the previous commands shown, the `EnvHostname` is the hostname or IP address of the system in which `Environment` module is running and `EnvWaitPort` is the port at which the `Environment` is waiting for a connection with the corresponding module.

3. The `RemoteControlRobot` module is the next one to be started. It can be done using the following command:

   ```
   java edu.ksu.cis.cooprobot.simulator.applications.maze.RemoteControlRobot 0
           <EnvHostname> <EnvWaitPort = 8000> <No. of sonars> <x-pos> <y-pos> <RCPort>
   ```

In the previous command, the EnvHostname and EnvWaitPort represent the same as mentioned in step 4. The next input is the size of sonar sensor array present in the robot. The x-pos and y-pos are the starting x and y coordinates of the robot. They can be used in future for any mapping applications. The RCPort is the port at which the RemoteControlRobot will be waiting for a socket connection from the RemoteControl.

4. Finally the RemoteControl is started using the following command:

```
java edu.ksu.cis.cooprobot.simulator.applications.maze.RemoteControl
                                                          <RobotConfigFile> <EnvHostname>
```

In the above command, the RobotConfigFile is the configuration file containing the information about the robots that will be controlled by the RemoteControl. The EnvHostname is the hostname of the system in which the Environment is running.

Chapter 1 Configuration File Formats

```
<?xml version="1.0"?>
<RobotInfo>
        <robot>
                <hostname><IP Address></hostname>
                <port><RCPort></port>
                <name><Robot Name></name>
                <primary><1/0></primary>
        </robot>
        …
</RobotInfo>
```

Figure 1. RobotConfigFile Format

Figure 1 shows the general format of a RobotConfigFile given as input to the RemoteControl. The information that should be specified in the file are the Hostname/IP Address of the RemoteControlRobot system, the Port at which the RemoteControl connects to it, the Robot - Name and a ‘1’ to indicate this robot is primary else a ‘0’. Only one RemoteControlRobot specified in the file can have a ‘1’ value in the primary tag.

```xml
<?xml version="1.0"?>
<world>
        <surface>
                <type>smooth</type>
                <size>
                        <xmin>-500</xmin>
                        <xmax>500</xmax>
                        <ymin>-500</ymin>
                        <ymax>500</ymax>
                </size>
                <object>
                        <shape>
                                <robot>
                                        <cylinder>
                                                <direction>0.0</direction>
                                                <name>Gold1</name>
                                                <x>30.0</x>
                                                <y>0.0</y>
                                                <z>0.0</z>
                                                <radius>10.0</radius>
                                                <height>1.0</height>
                                                <color>1.0 1.0 1.0</color>
                                                <hot>0</hot>
                                                <stationary>0</stationary>
                                        </cylinder>
                                        <sensor>
                                                <id>1</id>
                                                <type>heat</type>
                                                <range>15</range>
                                        </sensor>
                                        <sensor>
                                                <id>2</id>
                                                <type>sonar</type>
                                                <mode>2</mode>
                                                <position>
                                                        <x-relative>1.5</x-relative>
                                                        <y-relative>0.0</y-relative>
                                                        <z-relative>3.0</z-relative>
                                                        <range>15</range>
                                                </position>
                                        </sensor>
                                        <sensor>
                                                <id>3</id>
                                                <type>laser</type>
                                                <position>
                                                        <x-relative>1.5</x-relative>
                                                        <y-relative>0.0</y-relative>
                                                        <z-relative>3.0</z-relative>
                                                        <dir-relative>0.0</dir-relative>
                                                        <range>20</range>
                                                </position>
                                        </sensor>
                                </robot>
                        </shape>
                </object>
        </surface>
</world>
```

Figure 2. EnvironmentConfigFile Example

Figure 2 shows an example for the EnvironmentConfigFile. This particular example contains the information about a robot named Gold1, facing an initial angle of 0.0radians at location (10,-90,5) with a radius of 3units, a height of 10 units, temperature of 0 degrees and is not stationary. It has a laser range finder with `ID` = 3, `RANGE` = 20, the position relative to the robot is (1.5, 0, 3). The parameters for the stationary objects are also specified in a similar manner, but sensor section will not be present and the tag <`robot`> will be replaced by <`primative`> and a value of 1 for the <`stationary`> tag.

Chapter 2 Controlling Buttons on Joystick

The Joystick buttons that are used for controlling the robot motions and their corresponding actions are:

BUTTON 2	-	Send CheckWin message to `Environment`
BUTTON 3	-	Accelerate the robot
BUTTON 4	-	Decelerate the robot
Joystick Handle moved left	-	Turn the robot left
Joystick Handle moved right	-	Turn the robot right