Stage in ROS

Stage is a simulation tool part of the Player/Stage project [1]. The Stage stack contained in ROS wraps this tool and contains the node stageros that exposes a subset of Stage’s functionalities via ROS. Stageros node simulates a world defined in a .world file, which contains details for sensors, robots and obstacles in the simulated world.

Running Stage in ROS

Before all we have to build Stage stack with the following command

```
$rosmake stage
```

After this, start an instance of roscore with:

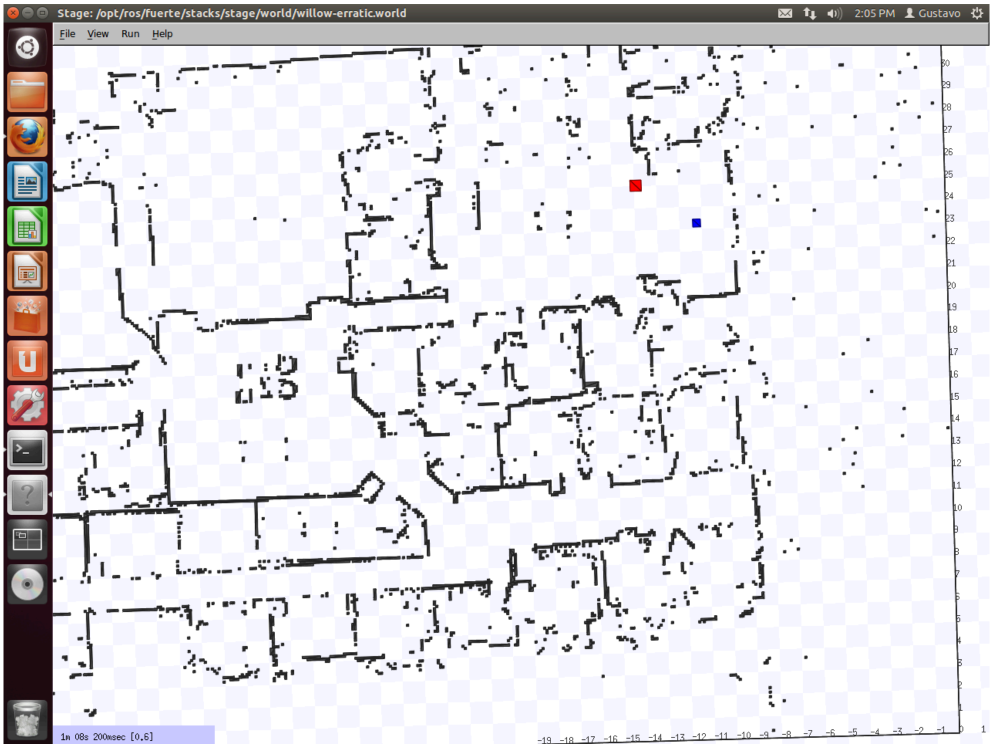
```
$roscore
```

And run stageros node from stage stack with an example world in other terminal, and then a stage window must appear:

```
$rosrun stage stageros `rospack find stage`/world/willow-erratic.world
```

Or

```
rosrun stage_ros stageros `rospack find  stage_ros`/world/willow-erratic.world
```


Stage: /opt/ros/fuerte/stacks/stage/world/willow-erratic.world
2:05 PM
Gustavo
File
View
Run
Help
1m 08s 200msec [0.6]

Now that we run stage, we can make our own simulation.

Creating a .world file

As said before, this file contains definition of the world and what is in it. From obstacles, walls, through sensors and robot. All entities are defined as models and are made using the following syntax:

```
define model_name model
(
      # parameters
)
```

Now let start creating a map of the world from a image file.

Map

The floorplan model is used to define the basic environment of simulation (like walls). This model is included in map.inc file (Located at ROS_PATH/stacks/stage/share/stage/worlds)

```
define floorplan model
(
        # sombre, sensible, artistic
        color "gray30"
        # most maps will need a bounding box
        boundary 1
        gui_nose 0
        gui_grid 0
        gui_move 0
        gui_outline 0
        gripper_return 0
        fiducial_return 0
        laser_return 1
)
```

To make use of this model, we have to write following code in our .world file (hall1.world), assuming that map.inc is in the same directory:

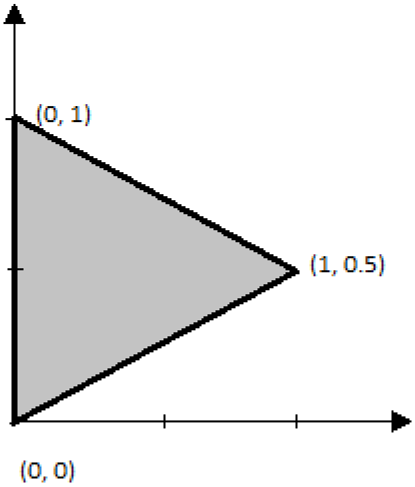
```
include “map.inc”
```

Within floorplan model we can find many parameters, for example “color” that is the color used for render the environment from image file, “boundary” is to activate a bounding box to prevent robot of wander out from the map. Other parameters explanation can be found in Stage manual at [2].

Robot

To simulate a robot in stage we have to include an existing model or, for this tutorial, we have to create one. A robot is model of type position and is made of its shape, characteristics and sensors, so in this order we will proceed

Now we will define a simple triangular-shaped robot called robcar, using the block model. In this we define the corner of the shape as points. This is the shape for the robot and code for create it:



```
define robcar position
(
    #Shape of robot
    block
    (
        points 3
        point[0] [1 0.5]
        point[1] [0 1]
        point[2] [0 0]
        z [0 1]
    )
    #fit shape to this size in meters
    size [0.5 0.5 0.5]
)
```

Now we have to add a sensor to our robot. In this case we use the ranger model in which we can define a range sensor and its features like range, field of view and number of samples. Also we can define a shape for attaching on the robot. This is the implementation of our sensor that we called robcar_laser.

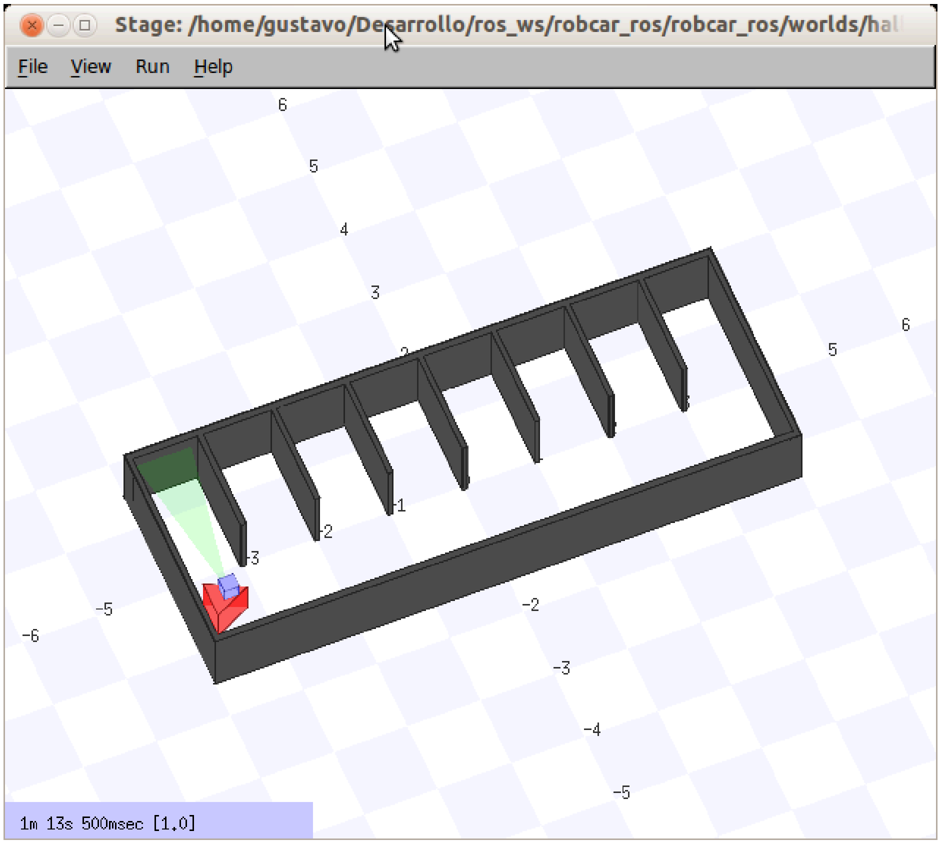
```
define robcar_laser ranger
(
    sensor(
        range [ 0.0  3.0 ]
        fov 20
        samples 1081
    )

    block(
            points 4
            point[0] [0 1]
            point[1] [1 1]
            point[2] [1 0]
            point[3] [0 0]
            z [0 1]
      )
      
  # generic model properties
  color_rgba [ 0 0 1 0.3 ]
  size [ 0.2 0.2 0.2 ]
  pose [0.0 0.0 0.3 90]
)
```

…

TO DO

…



Stage: /home/gustavo/Desarrollo/ros_ws/robcar_ros/robcar_ros/worlds/hal

```
1m 13s 500msec [1.0]
```

References

[1] http://playerstage.sourceforge.net/

[2] http://playerstage.sourceforge.net/doc/Stage-3.2.1/

[3] Owen, Jennifer. “How to use Player/Stage”. 2nd edition. Available at http://www-users.cs.york.ac.uk/~jowen/player/playerstage-tutorial-manual.pdf. 2010.

[4] http://www.ros.org/wiki/stage/Tutorials/SimulatingOneRobot