GETTING STARTED WITH USARSIM

Amirreza Kabiri
Fatemeh Pahlevan
Autumn 2017
OUTLINE

- Intro
- Requirements
- Installation
- Installation problems
- How to setup
- How to use
- Robots
INTRO

- USARSim was the first server used as a simulation infrastructure for a virtual robot competition
- In 2016, Robocup Virtual Robot Committees decided to change the server to Gazebo/ROS
- USARSim fans and previous Competitors could use the USARSim code to connect to the Gazebo environments using the provided USARGazebo plugin.
- In This Lecture, USARGazebo will be explained in some details according to the RobocupRescuePackage.
REQUIREMENTS

- **Hardware**:
  a. core i-7 (CLK 3GHz)
  b. Mem 8GB
  c. nVidia Geforce GTX780Ti

- **Software**:
  a. Recommend:
     i. Ubuntu 14
     ii. Gazebo 5
     iii. Indigo

Amirreza Kabiri & Fatemeh Pahlevan @ Autumn 2017
In Robocup 2017, Software requirements was as followed:

- Ubuntu 16.04
- Kinetic
- Gazebo7

As in Robocup 2017, all the teams decided to participate using ROS/Gazebo manner, this structure remained unchanged and there might be some changes in the following instructions.
First the Gazebo package should be installed.

It is worth mentioning that RobocupRescuePackage does not require ROS installation. So if any ROS version has not been installed, you only have to use following commands:

```
$ sudo apt-get install libgazebo5 libgazebo5-dev gazebo5
$ sudo apt-get install protobuf-compiler
```
If you already have install ROS indigo with desktop-full command, it may installed Gazebo2. In the case, at first, you have to remove Gazebo 2, and then, you can install Gazebo 5.

```
$ sudo apt-get purge gazebo2*
$ sudo apt-get autoremove
$ sudo apt-get install libgazebo5 libgazebo5-dev gazebo5
$ sudo apt-get install protobuf-compiler
```
In Gazebo5 installation, you will often encounter "broken dependency" error. So don’t worry and just follow these 3 steps.

1: Start with a clean installed Ubuntu 14.
   (no tips, just do it by an ordinarily method)

2: ROS indigo installation, if you need ROS indigo.

   $ sudo apt-get install ros-indigo-desktop
3: Gazebo5 installation with running over "broken dependency" error

$ sudo dpkg --configure -a
$ sudo apt-get install -f
$ sudo sh -c 'echo "deb http://packages.osrfoundation.org/gazebo/ubuntu \`lsb_release -cs` main" > /etc/apt/sources.list.d/gazebo-latest.list'
$ wget http://packages.osrfoundation.org/gazebo.key -O - | sudo apt-key add -
$ sudo apt-get update
$ sudo apt-get install libgazebo5 libgazebo5-dev
$ sudo apt-get install gazebo5
$ sudo apt-get install protobuf-compiler
If you are already using Gazebo4 and ROS hydro by installing drcsim-hydro package, you may need to install “libgazebo4-dev” package instead of “libgazebo5-dev”.

$ sudo apt-get install libgazebo4-dev
You can install “git” or download the package from the [github repository](https://github.com/m-shimizu/RoboCupRescuePackage/).

```bash
sudo apt-get update
sudo apt-get upgrade
sudo apt-get install git
$ cd ~
$ git clone https://github.com/m-shimizu/RoboCupRescuePackage/
```
After downloading or cloning the RoboCupRescuePackage, you have to “make” it using the following Commands:

```bash
$ cd ~/RoboCupRescuePackage
$ source ./bashrc.USARGazebo
$ mkdir build
$ cd build
$ cmake ../
$ make
```
By this tutorial, you need to open four Terminal pages for running the package.

- The following commands should be written in corresponding terminals.

1st Terminal:

```bash
$ cd ~/RoboCupRescuePackage
$ source ./bashrc.USARGazebo
$ gazebo USARGazebo.world
```

- Gazebo will be opened and show the world file which is in the USARGazebo.world
HOW TO USE

- 2nd Terminal:

  ```
  $ telnet localhost 3000
  ```

  Now, These 3 Commands( GETSTARTPOSES, INIT, DRIVE) could be used in this terminal.

  Here is some examples:

  (1) GETSTARTPOSES
      Getting Usable Start Location

  (2) INIT {ClassName pioneer3at_with_sensors} {Name Robot1} {Location 5,10,0.3} {Rotation 0,0,0}
      a. Spawn a pioneer3at robot named “robot1” at the coordinate (5, 10, 0.3) with no rotation
      b. Spawning multi robots requires 1 sec interval between each robot spawning.
How to use

- **2nd Terminal (Cont’):**

  DRIVE {RIGHT 1.0}{LEFT 1.0}
  Drive the robot with speed 1 in each wheel
  MULTIDRIVE {FRFlipper 1.57}{FLFlipper 1.57}{RRFlipper 1.57}{RLFlipper 1.57}
  Move flipper arms of a crawler robot with "MULTIDRIVE" command

- **3rd Terminal:**

  $ telnet localhost 5003
  And then type “OK”
  This command returns a frame of camera image in USARSim style.
3rd Terminal (Cont’):

- With this commands, you can get a long landscaped jpeg image consisted from 4 robot cameras.
- Each robot’s camera image has width=640 and height=480.
- Camera images are located side-by-side and first spawned robot's camera’s image is located at left end of a jpeg image.
- If a robot has two cameras, currently you can see only right camera's image.
How to Use

4th Terminal:

```sh
$ cd ~/RoboCupRescuePackage
$ source ./bashrc.USARGazebo
$ cd build
$ ./robot_teleop pioneer3at_with_sensors 2
```

Now, you can control the robot with keyboard.

"robot_teleop" is a program which have a plugin named"SkidSteerDrive" and control the robot through the "/vel_cmd" topic.

Amirreza Kabiri & Fatemeh Pahlevan @ Autumn 2017
How to use

4th terminal (Cont’):

In “robot_teleop”:

- The 1st argument is the robot model name. Please check it in the model name tree in the left sidebar of the gazebo window. If you see a different name like "pioneer3at_with_sensors_0", replace the 1st argument with it.
- The 2nd argument is a plugin type number defined in this program. You shouldn’t change this argument.
**HOW TO USE**

- **4th terminal (Cont’):**
  - In order to stop this program, push control key and C key on your keyboard at same time.
  - Shutdown process of Gazebo needs 15~20 sec. You should wait at least this much time to before starting the Gazebo again.
HOW TO USE

- XML tags for USARGazebo in a world file are as followed
  - Location and orientation of start point
    ```xml
    <StartPose_1>LOCATIONNAME1 X Y Z Roll Pitch Yaw</StartPose_1>
    <StartPose_2>LOCATIONNAME2 X Y Z Roll Pitch Yaw</StartPose_2>
    <StartPose_3>LOCATIONNAME3 X Y Z Roll Pitch Yaw</StartPose_3>
    <StartPose_4>LOCATIONNAME4 X Y Z Roll Pitch Yaw</StartPose_4>
  ```
  - Turning Ground Truth on
    ```xml
    <GroundTruth>true</GroundTruth>
    ```
Robots

- pioneer3at_with_sensors series
  a. Classname:
     i. pioneer3at_with_sensors
     ii. pioneer3at_with_sensors_r
     iii. pioneer3at_with_sensors_b
     iv. pioneer3at_with_sensors_g
     v. pioneer3at_with_sensors_y
  b. Equipments:
     i. 2 encoder sensors
     ii. 1 camera
     iii. 1 range sensor
ROBOTS

- pioneer2dx_with_sensors series
  a. Classname:
     i. pioneer2dx_with_sensors_r
     ii. pioneer2dx_with_sensors_b
     iii. pioneer2dx_with_sensors_g
     iv. pioneer2dx_with_sensors_y
  b. Equipments:
     i. 1 camera
     ii. 1 range sensor
Robots

- turtlebot_with_sensors series
  a. Classname:
     i. turtlebot_with_sensors_r
     ii. turtlebot_with_sensors_b
     iii. turtlebot_with_sensors_g
     iv. turtlebot_with_sensors_y
  b. Equipments:
     i. 1 camera
     ii. 1 range sensor
ROBOT

- crawler_robot series
  a. Classname:
     i. crawler_robot_r
     ii. crawler_robot_b
     iii. crawler_robot_g
     iv. crawler_robot_y
  b. Equipments:
     i. 1 camera
     ii. 1 range sensor
     iii. 4 flipper arms.
crawler_robot series

a. Flipper arms:
   i. You can control the angle of a flipper arm with MULTIDRIVE command.
   ii. MULTIDRIVE {FRFlipper 1.57}{FLFlipper 1.57}{RRFlipper 1.57}{RLFlipper 1.57}
A sample client software was prepared to show fundamental usage of the RoboCupRescuePackage.

See https://github.com/m-shimizu/USARSimSampleClient

In USARSimSampleClient:
- You can spawn 4 robots
- 4 robot's camera images transferred in jpeg format are shown on rviz
WORKABLE USARSI SIM COMMAND

- USARSim commands
  - INIT
  - GETCONF
  - GETGEO
  - DRIVE
  - MULTIDRIVE
    - only for controlling flipper arms
  - GETSTARTPOSES
    - It can return effective start point parameters
    - It can not read parameters from map but it can read those from world file as plugin options

- Image Server
  - OK (return Jpeg image)