Akshitha Pothamshetty
116399326
ENPM690 HW#3

Github Repository Link:

https://github.com/AkshithaPothamshetty/ENPM690-AutonomousRobot

Instructions to run the program:

1. Create a Catkin Workspace

```
mkdir -p ~/catkin_ws/src
```

2. Initialize the Workspace

```
cd ~/catkin_ws/src
catkin_init_workspace
```

3. Extract the submission folder content into the src directory

4. Build the packages and source the setup file

```
cd ~/catkin_ws
catkin_make
```

5. Execute the launch files

```
cd ~/catkin_ws
```

Open two terminals(T1 and T2) and run the following commands:

```
T1: source ~/catkin_ws/devel/setup.bash
T1: roslaunch my_robot robot_description.launch

T2: source ~/catkin_ws/devel/setup.bash
T2: roslaunch ball_chaser ball_chaser.launch
```

6. Drag the white ball in view cone of the robot (*in Gazebo)*

Note: - ROS needs to be installed on the system to run this project.

Akshitha Pothamshetty
116399326
ENPM690 HW#3

Project Report

In this project, I have demonstrated a robot equipped with laser sensor(Hokuyo) and Camera to follow a white ball when it comes in its view cone.

Project Structure

```
.Project2                          # ENPM690 HW#3
    ├── my_robot                       # my_robot package
    │   ├── launch                     # launch folder for launch files
    │   │   ├── robot_description.launch
    │   │   ├── world.launch
    │   ├── meshes                     # meshes folder for sensors
    │   │   ├── hokuyo.dae
    │   ├── urdf                       # urdf folder for xarco files
    │   │   ├── my_robot.gazebo
    │   │   ├── my_robot.xacro
    │   ├── world                      # world folder for world files
    │   │   ├── RobotWorld.world
    │   ├── CMakeLists.txt             # compiler instructions
    │   ├── package.xml                # package info
    ├── ball_chaser                    # ball_chaser package
    │   ├── launch                     # launch folder for launch files
    │   │   ├── ball_chaser.launch
    │   ├── src                        # source folder for C++ scripts
    │   │   ├── drive_bot.cpp
    │   │   ├── process_images.cpp
    │   ├── srv                        # service folder for ROS services
    │   │   ├── DriveToTarget.srv
    │   ├── CMakeLists.txt             # compiler instructions
    │   ├── package.xml                # package info
    └──
```

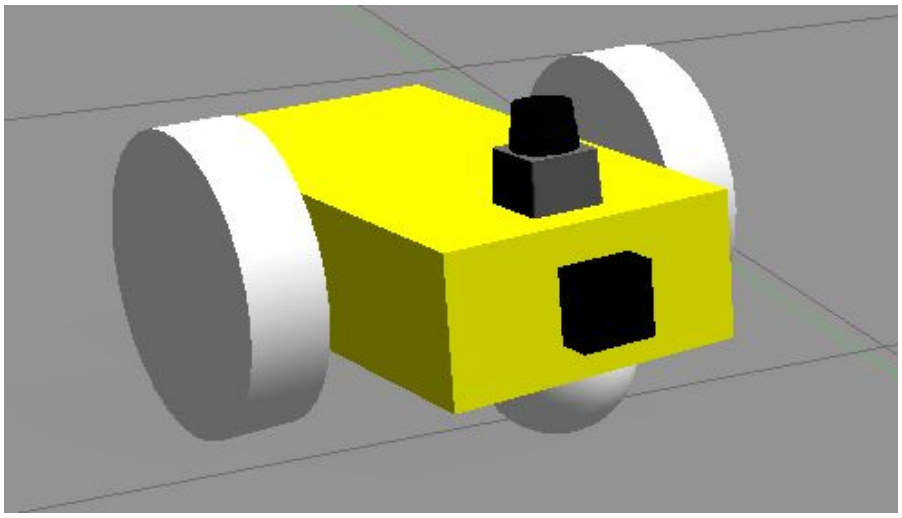
Project can be splitted into two sub-tasks:

1. Building Robot Package

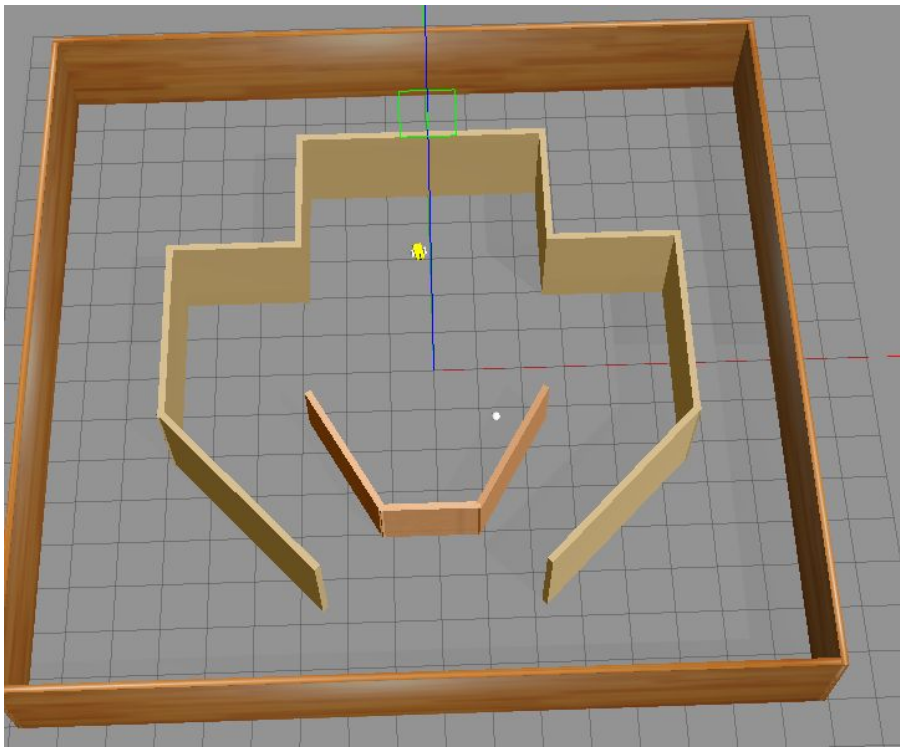
In this Package, I built a differential drive robot from scratch in URDF Scripts. For the Housing of the Robot, I used Gazebo Model Editor. Building a robot in URDF format was challenging at first, but I have kept my Robot features as pretty basic. It contains Cuboid Chassis, Two Cylindrical Wheels, HemiSpherical Castor Wheel, Hokuyo Laser Sensor on top and a Camera in the front. Added Gazebo plugins for robot's differential drive, lidar, and camera.

Akshitha Pothamshetty
116399326
ENPM690 HW#3

Robot Model:



Then I build a House for the Robot using the Model Editor:



I added a white ball inside the world. A launch file(world.launch) was created to launch the world.

Akshitha Pothamshetty
116399326
ENPM690 HW#3

2. Building Ball Chaser Package

Created a ball_chaser package which consists of following nodes, services and launch files:

1. drive_bot.cpp (Node)
2. process_image.cpp (Node)
3. ball_chaser.launch (Launch Files)
4. DriveToTarget (service)

The ***drive_bot*** C++ node provides a ball_chaser/command_robot service to drive the robot by controlling its linear x and angular z velocities. The service publishes to the wheel joints and returns back the requested velocities.

The ***process_image*** C++ node reads the robot's camera image, analyzes it to determine the presence and position of a white ball. If a white ball exists in the image, node requests a service via a client to drive the robot towards it.

The ***ball_chaser.launch*** runs both the drive_bot and process_image node.

Summary

In this project, I designed an autonomous Robot and its Home. I created another package(ball_chaser) to move the robot when it sees the white ball. I got a better understanding of ROS nodes, Services and Launch files through this project. Later on, I would like to make this robot map the environment on its own using the AMCL package available in ROS. Overall, this was a good project for getting started with ROS.

References

1. http://wiki.ros.org/ROS/Tutorials