CS460 Open Project Report

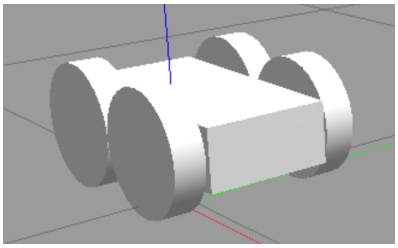
Karun Kanda (kk951)

December 6th, 2021

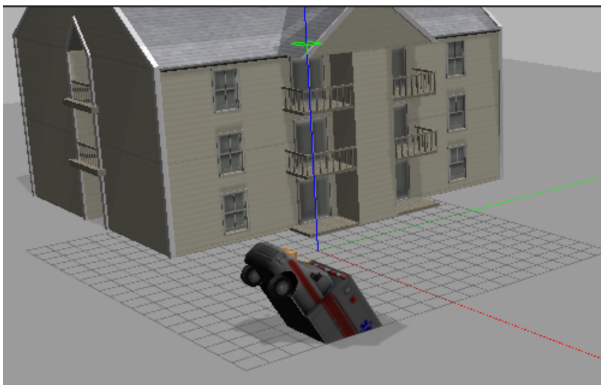
1 Bonus Project 1

1.1 Setting up the Demonstration

To setup this project first I had to download ros, gazebo and mobaxterm because I was running this on Windows Subsystem for Linux. To download those three items there are many websites that show you step by step tutorials on how to set them up. Afterwards, I followed a tutorial on the ros website on how to setup a catkin workspace and a ros package. When the ros workspace was setup I had to setup additional directories such as src (for the ros python programs), launch (to launch the environment), urdf (for the robot's model), and world (for the world's model).
After those were setup, I first setup the model that I also used the gazebo website to help me make the mobile robot which I made with 4 wheels and a rectangular base. Then I had to edit the sdf file to add a plugin to make sure the robot actually moves using the skid steer controller plugin found on the gazebo website.



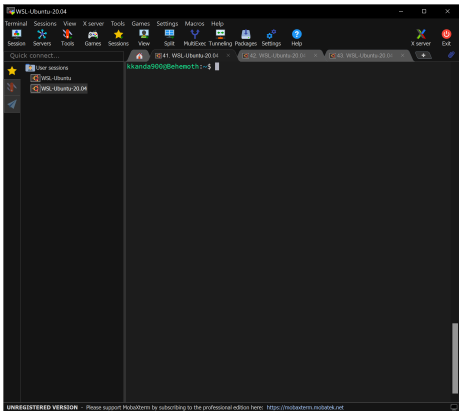
Then to setup the environment I made that through the gazebo software where I added a building and a few other objects in the scene.



After putting those models in there respective place I made a .msg file in the new msg directory called Position.msg. In Position.msg it holds a string direction that we fill in using the **publish_values.py**. There is only one that that says *string direction* Direction will be either forward or turn which aligns with making the robot go forward or turn. Once that .msg file is setup we can use **publish_values.py** to assign the direction then use **publisher_node.py** to make the robot go the direction that was assigned in Position.msg.
After making the .msg file I made two python files to publish the values into the message and one file to load the files and make the robot move. **publish_values.py** initializes one ros node that makes direction in Position.msg a value. Then **publisher_node.py** first initializes a node that subscribes the direction into the direction in Position.msg and then make the robot move based on speed and a time.

1.2 How to use the Program

To use the program first setup three Linux terminals where one will be for the demonstration, one will be for assigning the direction and one will actually make the robot move.



Before going into the catkin workspace we need to setup ros. So type in the command **source /opt/ros/noetic/setup.bash** in all three terminals.

```
kkanda900@Behemoth:~$ source /opt/ros/noetic/setup.bash
```

Then we want to change the directory to the catkin workspace which in this case I set to be **cs460_ec**, once your in the catkin workspace do **catkin_make** to make the workspace has the latest compiled dependencies. Once that is done, the final setup is to use **source devel/setup.bash** to make sure you can launch the environment.

```
kkanda900@Behemoth:~/cs460_ec$ catkin_make && source devel/setup.bash
```

Now make sure all the directories are changed to src/robot_cs460 to get into the ros package. Now making sure that in the first terminal use **roslaunch robot_cs460 environment.launch** to start the environment.

```
kkanda900@Behemoth:~/cs460_ec/src/robot_cs460$ roslaunch robot_cs460 environment
.launch
```

Once gazebo fully launches and you can see the robot, go to the second terminal and use the command **rosrun robot_cs460 publish_values.py <direction>**.

```
kkanda900@Behemoth:~/cs460_ec/src/robot_cs460$ rosrun robot_cs460 publish_values
.py forward
```

Once you see that the direction is published then go to the third terminal and use the command **rosrun robot_cs460 publisher_node.py <speed> <time>** which will take the speed and make it run at a reasonable speed so you can see it.

```
kkanda900@Behemoth:~/cs460_ec/src/robot_cs460$ rosrun robot_cs460 publisher_node
.py 10 10
```

Make sure too ctrl+Z all of the terminals after use and close gazebo so it doesn't eat your CPU! :)

1.3 Demonstration Link

Click the picture to be directed to the demonstration.