MARUM - Center for Marine Environmental Sciences

Submarine simulator with gazebo

Author:
Alexandre Colon

December 10, 2019

Abstract

This document will explain how I did a simulator for submarines with gazebo and a PID controller during my internship in Marum. I will explain how to use my work, how to change it (If you want to change some parameters or 3D models for example) and how I did so you can create a new submarine if necessary. I will also give some link to tutorials, it is important to do some of them if you are a beginner with ROS and Gazebo in general. If you already have some skills in ROS and Gazebo, it is not necessary for you the tutorials and you can directly go to the part where I explain how to make the submarine work and how to change it.

If you have any question, you can contact me at this email address : alex@colon.fr

Contents

Chapter 1

ROS and Gazebo

This part is dedicated to people who never, or don't really worked with ROS or Gazebo

1.1 Installation

If you are not using Ubuntu, or if you have a problem with the installation, a link is available in the bibliography

To install ROS melodic on Ubuntu :

```
sudo apt install ros-melodic-desktop-full
```

after you have to initialize rosdep :

```
sudo rosdep init
rosdep update
```

and finally :

```
source /opt/ros/melodic/setup.bash
```

To install Gazebo on Ubuntu, you just have to run this commands :

```
curl -sSL http://get.gazebosim.org | sh
```

If you are not using Ubuntu or if you have troubles during the installation, a link is available in the bibliographie.

1.2 Learn Gazebo and ROS

Two versions of the submarine are available on the github, one with .SDF files, and the other one with .URDF and .xacro files and UUV Simulator [1]

The first version :

- Very simple version based on the tutorials on the Gazebo website
- Easy to change
- Simple and effective plugins
- Easy to debug

[1] The Unmanned Underwater Vehicle Simulator is a set of packages that include plugins and ROS applications that allow simulation of underwater vehicles in Gazebo.

- Probably do not take enough physics parameters
- Not finished, it is necessary to enter the good parameters for the propellers

The second version :

- Use UUV simulator
- have a lot of physics parameters
- Hard to debug

So if you want to work on the first version, you should do these tutorials :

http://gazebosim.org/tutorials?cat=get_started,
http://gazebosim.org/tutorials?cat=build_robot,
http://gazebosim.org/tutorials?tut=hydrodynamicscat=physics

If you want to work on the second version, these tutorials can be good :

http://wiki.ros.org/ROS/Tutorials/UnderstandingTopics,
http://wiki.ros.org/fr/urdf/Tutorials

Chapter 2

Install and use the simulator

In this section, I will explain how to install UUV simulator, how to run the simulator, how to update the submarine, and how you can run the submarine.

2.1 Setup you environment

A quick tutorial of how install UUV simulator, setup you catkin package, and put your submarine inside.

2.1.1 Install UUV simulator

If you are using ROS melodic on Ubuntu :

```
sudo apt install ros-melodic-uuv-simulator
```

As usual, if you have a problem with this, a link if available with more explanations

2.1.2 Creating you workspace

If you don't have any workspace, here how to create one :

```
mkdir -p ~/catkin_ws/src
cd ~/catkin_ws
catkin init
```

2.1.3 Add UUV simulator and the submarine

This part is mandatory :

```
cd ~/catkin_ws/src
git clone https://github.com/uuvsimulator/uuv_simulator.git
git clone https://github.com/manateesubmarine/manatee_simulator
mv manatee_simulator submarine
cd ~/catkin_ws/
catkin build
source devel/setup.bash
```

2.2 Use the simulation

Here a simple tutorial of how you can use the simulator

```
roslaunch uuv_gazebo_worlds empty_underwater_world.launch
```

This command will open and load a world in gazebo [1]

[1] It is possible to create very easily a world with a custom seafloor in Gazebo, a link with explanations of how to do it is available in the bibliography

```
roslaunch submarine_description upload.launch
```

This command will load the submarine

And finally run

```
roslaunch submarine_control start_thruster_manager.launch reset_tam:=true
```

Of course, every command will have to be done in a different terminal

---

!! WARNING !! An error could appear, it can looks like this :

```
FILE:LINE: YAMLLoadWarning: calling yaml.load() without Loader=... is deprecated,
as the default Loader is unsafe.
Please read https://msg.pyyaml.org/load for full details.
```

If this error appears, don't worry, you just have to edit the file, change the function *yaml.load()* by *yaml.safe_ load()*, and it will work correctly.

Then, open a new terminal, and run the command

```
rostopic pub /submarine/thrusters/THRUSTER_ID/input uuv_gazebo_ros_plugins_msgs/FloatStamped "header:
    seq: 0
    stamp:
        secs: 0
        nsecs: 0
    frame_id: ''
data: VALUE"
```

Don't forget to change the VALUE. It has to be a float, you can put 5.0 for example

You also have to change THRUSTER_ID value. You can put 0 for the left thruster, and 1 for the right one.

If you want to control manually the fins, you can do the same :

```
rostopic pub /submarine/fins/FINS_ID/input uuv_gazebo_ros_plugins_msgs/FloatStamped "header:
    seq: 0
    stamp:
        secs: 0
        nsecs: 0
    frame_id: ''
data: VALUE"
```

Chapter 3

Change the submarine

In this part, I will explain how to change the submarine, and the things to do if you want that your submarine works properly.

3.1 Requirements

I advice you to install MeshLab. MeshLab is a software that can help to find most of the values to make work the simulator.
Here how to install MeshLab on Ubuntu

```
sudo apt-get install meshlab
```

3.2 3D models

All the 3D models have to be in the folder *submarine/meshes/*, 3 files are necessary: vehicle.stl, vehicle.dae and propeller.dae
To help yourself, try to have the center of gravity of the vehicle at the coordinates [0, 0, 0]
Actuators and vehicles are modeled separately in the robot description and should also have separate mesh files. The thruster plugin macros set the axis of rotation of the thruster as the X axis, so it is better to setup the mesh accordingly

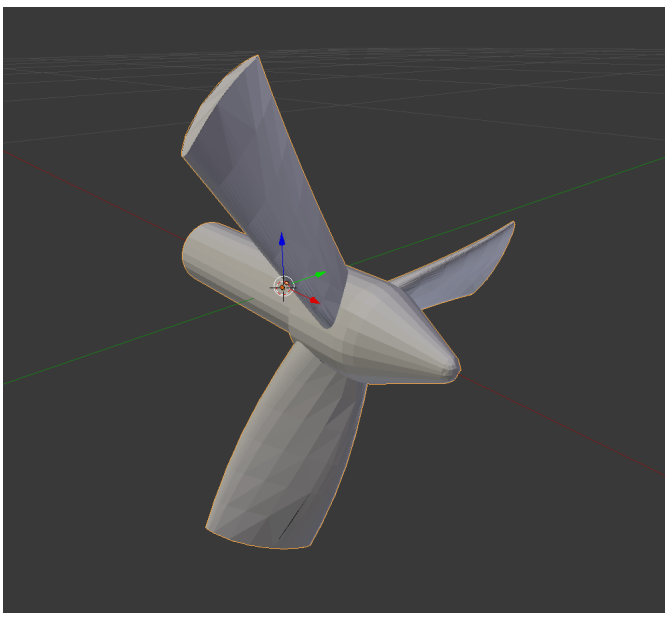

Figure 3.1: Example

3.3 Body of the submarine

I will explain you how to implement the body of the submarine in the simulator

To do so, yo have to edit to */submarine/urdf/base.xacro* file

Open meshLab, click on *View ->Show Layer Dialog*, then import the file vehicle.stl

Then, click on *Filters -> Quality Measure and Computations -> Compute Geometric Measures* and a lot of usefull informations will appear.

MeshLab will provide you some informations as the size, the volume, the center of mass of the body of the submarine.

The first value to change is the mass of the vehicle, to do so, you have to calculate FluidDensity [1] * Volume

Then, you have to change the center of gravity, in my case I took the center of mass given by MeshLab

Then, you have to change the inertia of the vehicle. For this, eshLab give you the inertia tensor of the model, and because it is a symmetric matrix, you just have to put 6 value

If you change the 3D model, don't forget to change the xyz of the origin of the visual and collision part of the vehicle macro

3.4 Thrusters

If you want to change the propellers properties, you will have to change two files, */submarine/urdf/snippets.xacro* and */submarine/urdf/actuators.xacro*

In the snippets file, a few values can be changed, the gain, the clampMax, clampMin, the thrustMin, thrustMax, the thrust_efficiency, the propeller_efficiency, the time_constant and the rotorConstant

The the actuator file, you can instantiate as many objects as you want, but be careful, you have to increment the thruster_id of each thruster, and then change the xyz, and rpy values if necessary

3.5 Fins

To change the fins, you will have to change the same files : */submarine/urdf/snippets.xacro* and */submarine/urdf/actuators.xacro*

In the snippets file, a few values can be changed, the min and max joint_limit, the lift and drag constant

3.6 Sensors

You can easily add a sensor such as an IMU, pressure sensor or a camera on the submarine. For this you have to update the */submarine/urdf/sensors.xacro* file

All sensors are already included, you can comment the ones you don't want, and then you can change their position and orientation on the submarine

Be careful that the file is well included in the *base.xacro* file. You can find all the documentation for sensors you need in this link : *http://gazebosim.org/tutorials?cat=sensors*

[1] By default 1028, correspond to salt water density

3.7 Worlds

The worlds are available in the directory uuv_simulator/uuv_gazebo_worlds/launch/*

To change a world, I advice you to choose one of them, and custom it the way you want

You can change the 3d mesh of the world to change the seafloor for example. To do so, you have to change the mesh with a .dea file

3.8 Submarine main file

The main file is the */submarine/urdf/gazebo.xacro*

If you change the submarine model, be careful, a few values have to changed

You first have to change the submarine_cob, that correspond to the center of buoyancy of the submarine

Then you have to change the submarine_volume value you already get with MeshLab

After, you will have to change the submarine_length, submarine_width and submarine_height values. To do it, you have to take the mesh bounding Box Size values you got with MeshLab

Then, you will have to change the hydrodynamics values of the submarine

Chapter 4

PID Controller

In this part, I will explain you how to use the PID controller

4.1 Install

To use the PID controller, you need a workspace. If you don't have any, please follow the explanations in the part 2.1.2

Then, run these commands

```
cd ~/catkin_ws/src
git clone https://github.com/manateesubmarine/manatee_PID
cd ~/catkin_ws
catkin_make
source /devel/setup.bash
```

4.1.1 Usage

To run the PID controller, you have to run this command :

```
roslaunch manatee_PID pid.launch
```

Of course, run the simulation before doing this !

In the file pid.launch, you can create differents ROS nodes, so you can run differents PID controller

Here an example for the depth control with the fins of manatee:

```
<node pkg="manatee_PID" name="pid" type="pid" output="screen"
      args="248.4207
      -0.6 0.6
      ../catkin/src/manatee_PID/pid.conf
      ../catkin/src/manatee_PID/depth.csv
      TIME,DEPTH
      /submarine/pressure
      /submarine/fins/0/input
      /submarine/fins/1/input">
    </node>
```

In our case, we are interested by the arguments.

- The first value is the desired value for the PID controller, in the example it is the pressure (15meters)
- The two next values are the min and max values you want, if this case, the min and max angle of the fins
- The pid.conf correspond to a file with the pid values so you just have to change this file and see change the values. Don't forget the ../PACKAGE/src/manatee_PID/ or it won't find the file
- This file will be necessary to get a graph of your vehicule movement

- This parameter will be usefull to have the good headers for your graph
- This is the topic you will subscribe, in our case the pressure we want
- Now, all the next parameters will be the topic you wil publish on, you can put as much as you want, in our case we change the fins angle to make the submarine move up and down

4.1.1.1 Graph

You maybe have noticed there is a graph.py file !

In fact, you can create of graph of the submarine mouvement with the command

```
python graph.py
```

To do so, be carefull you have the same header in your PID parameters that in the file. TIME is always the first one, and then put the value you want to track

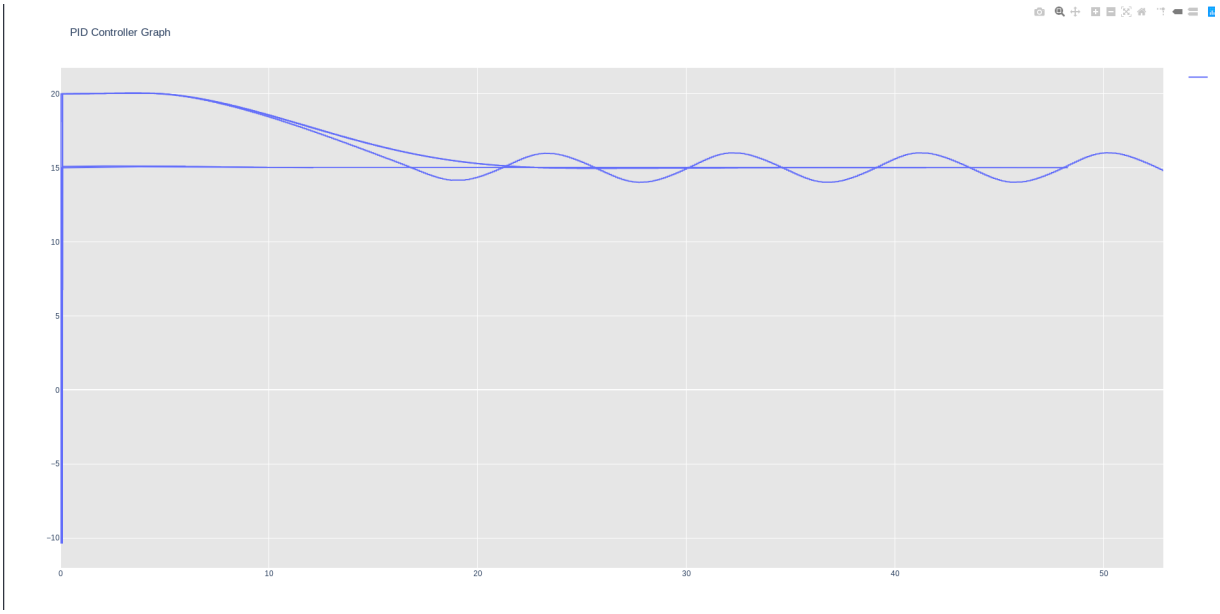

Figure 4.1: Example of graph

To create the graph, I choosed Plotly, Plotly provides online graphing, analytics, and statistics tools for individuals. You can do a lot of thing with to analys your PID controller and mouvements.

Chapter 5

Versions

Here some versions of the software I used to make this work

- ROS melodic
- Ubuntu 18.04.3 LTS
- Gazebo 9.11.0 7.4.0

Bibliography

[1] How to create a new world for the submarine. https://uuvsimulator.github.io/packages/uuv_simulator/docs/tutorials/seabed_world/.

[2] How to install gazebo. http://gazebosim.org/tutorials?cat=install.

[3] How to install ros. http://wiki.ros.org/melodic/Installation.

[4] How to install uuv simulator and setup you environment. https://uuvsimulator.github.io/installation/.