

Autonoship Simulation User Guide

1. Introduction

The development of autonoship simulation package aims at building a testing environment for ship collision avoidance and autonoship system. The simulation can support a controllable own ship with sensors including radar, camera, GPS, and IMU, and multiple controllable target ships.

This project includes the autonoship gazebo simulation package and usable ship dynamics and radar plugins. An installation guidance is contained.

2. System Requirements

This simulation is built on Ubuntu 16.04 (<http://releases.ubuntu.com/16.04/>) and ROS kinetic (<http://wiki.ros.org/kinetic/Installation/Ubuntu>). For installation and usage, please follow the instructions in the file *autonoship_guidance.html*

3. Simulation environment

3.1. Number of target ships

The demo of the autonoship simulation comes with three target ships. While finding that three target ships are too heavy for the game machine, please try to reduce the number of target ships.

All launch files are in the directory: `<workspace_name>/src/autonoship_simulation/launch`

The target ships are spawned into the simulation by the launch file: *spawn_ship.launch*

Examples of including this launch file can be found in *autonoship_gazebo.launch*:

```
<include file="$(find autonoship_simulation)/launch/spawn_ship.launch" >
  <arg name="namespace" value="targetship2" />
  <arg name="wind" value="false"/>
  <arg name="radar" value="false"/>
  <arg name="ownship" value="false" />

  <arg name="x" value="3000.0" />
  <arg name="y" value="0.0" />
  <arg name="z" value="$(arg z)" />
  <arg name="P" value="$(arg P)" />
  <arg name="R" value="$(arg R)" />
  <arg name="Y" value="$(arg Y)" />

  <arg name="urdf" value="$(arg urdf)"/>
</include>
```

Set each target ship a unique namespace to control different ships separately.

The argument “*ownship*” indicates whether the spawning ship is an own ship or a target ship. The own ship is equipped with radar model, while target ship is not. When radar is required on the target ship, turn the argument “*ownship*” to *true* to load the model with radar.

The argument “*wind*” should always be *false*. The wind module comes from the RobotX

simulator, and has not been tested on the autonoship simulator.

If the argument “*ownship*” is set *false*, keep the “*radar*” as *false* too. Otherwise, switch the “*radar*” according to your needs.

3.2. Capability of target ships

When you need your target ship to have the same sensing capability as your own ship, spawn your target ship as:

```
<include file="$(find autonoship_simulation)/launch/spawn_ship.launch" >
  <arg name="namespace" value="targetship" />
  <arg name="wind" value="false"/>
  <arg name="radar" value="$(arg radar)"/>
  <arg name="front_camera" value="true"/>
  <arg name="gps" value="true"/>
  <arg name="imu" value="true"/>
  <arg name="p3d" value="false"/>
  <arg name="ownship" value="true" />

  <arg name="x" value="1000.0" />
  <arg name="y" value="-1000.0" />
  <arg name="z" value="$(arg z)" />
  <arg name="P" value="$(arg P)" />
  <arg name="R" value="$(arg R)" />
  <arg name="Y" value="$(arg Y)" />

  <arg name="urdf" value="$(arg urdf)"/>
</include>
```

Give your target ship a different namespace to your own ship. The radar, camera, GPS, and IMU can be turned on by setting their arguments to *true*.

Notice! If you are doing so to create a target ship, remember to comment out the *autonoship_keys2rudder.launch* in the *spawn_ship.launch* (line: 44-46). The nodes are used to control the own ship from keyboard inputs. Launching multiple keyboard subscribers may result in weird behavior in the controlling.

3.3. Adjustment of time flow

The time flow can be adjusted in the file:

```
<workspace_name>/src/autonoship_simulation/worlds/autonoship.world
```

The actual time flow in the simulation can be calculated by:

$$\begin{aligned} &< real_time_update_rate > * < max_step_size > \\ &= 5000 * 0.001 \\ &= 5.0 \text{ (secs in simulation/ sec in real world)} \end{aligned}$$

4. Parameters of sensors

Parameters of all plugins can be found in the files:

<workspace_name>/src/autonoship_simulation/autonoship_gazebo/urdf/autonoship.launch*

5. Running the simulation

- 1) To control the ownship with keyboard ("w, a, s, d, q, e"), in a terminal, run:

```
roslaunch autonoship_simulation autonoship_gazebo.launch
```

- 2) To control targetships, publish message in the terminal:

```
rostopic pub targetship1/u2 std_msgs/Float64 10000000
```

- 3) To test the radar module, echo the radar feedback in the terminal:

```
rostopic echo own_ship/logical_camera
```

- 4) To change the RPM of the radar to 80, run in a terminal:

```
rostopic pub own_ship/rpm std_msgs/Float64 80
```