Gazebo USV Plugins, Theory of Operation

Brian Bingham, February 26, 2018

1 Overview

The rigid body dynamics implemented in Gazebo are augmented with environmental forces to simulate an unmanned surface vessel via a set of Gazebo plugins. These plugins simulate the effects of

- Dynamics
  - Maneuvering - added mass, drag, etc.
  - Wave field - motion of the water surface
- Thrust - vehicle propulsion
- Wind - windage.

This document is a high-level description of the implementation, but not a full documentation of the models and techniques used. The code itself is the ultimate documentation; comments have been included in the source to allow for users to adapt and extend these simple techniques for their own purposes.

Our goal with these tools is to create a simulation environment that provides sufficient fidelity so that techniques developed in simuation can be transitioned to the physical world, but that are also as simple as possible to minimize compuational requirements. The design has attempted to provide an environment consistent with the following rules:

- Autonomy that does not work in simulated world will not work in the physical world.
- Autonomy that does work in the physical world will work in the simulated world.

2 Maneuvering Model and Waves

The influence of both maneuvering and wave forces on the motion of the USV is implemented as a single model plugin for Gazebo. Currently there is only one implementation, but other implementations, with varying fidelity, could be developed in the future.

2.1 usv_gazebo_dynamics_plugin

2.1.1 Maneuvering Model

The plugin implements a portion of the nonlinear maneuvering equations from [1],

$$\underbrace{\boldsymbol{M}_{RB}\dot{\boldsymbol{\nu}} + \boldsymbol{C}_{RB}(\boldsymbol{\nu})\boldsymbol{\nu}}_{\text{rigid-body forces}} + \underbrace{\boldsymbol{M}_A\dot{\boldsymbol{\nu}}_r + \boldsymbol{C}_A(\boldsymbol{\nu}_r)\boldsymbol{\nu}_r + \boldsymbol{D}(\boldsymbol{\nu}_r)\boldsymbol{\nu}_r}_{\text{hydrodynamic forces}} = \boldsymbol{\tau} + \boldsymbol{\tau}_{wind} + \boldsymbol{\tau}_{waves} \tag{1}$$

where the state vector $\boldsymbol{\nu} = [u, v, r]^T$ includes the velocities u, v and r are in the surge, sway and yaw directions respectively and $\boldsymbol{\nu}_r$ is the velocity vector relative to an irrotational water current $\boldsymbol{\nu}_c$, i.e., $\boldsymbol{\nu} = \boldsymbol{\nu}_r + \boldsymbol{\nu}_c$. The *rigid-body forces* components are simulated via the Gazebo physics engine, while the *hydrodynamic forces* are determined via the plugin.

The six DOF maneuvering model is specified by the following matrices. The added mass matrix expressed as

$$\boldsymbol{M}_A = \begin{bmatrix} -X_{\dot{u}} & 0 & 0 & 0 & 0 & 0 \\ 0 & -Y_{\dot{v}} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.1 & 0 \\ 0 & 0 & 0 & 0 & 0 & -N_{\dot{r}} \end{bmatrix}. \tag{2}$$

The Coriolis-centripetal matrix for the added mass expressed as

$$\boldsymbol{C}_A(\boldsymbol{\nu}_r) = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & Y_{\dot{v}}v_r + Y_{\dot{r}}r \\ 0 & 0 & 0 & 0 & 0 & -X_{\dot{u}}u_r \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -Y_{\dot{v}}v_r - Y_{\dot{r}}r & X_{\dot{u}}u_r & 0 \end{bmatrix}. \tag{3}$$

. It is worth noting that $\boldsymbol{C}_A$ includes the nonlinear Munk moment (see [1] p.121). Following [1] the SNAME notation for the hydrodynamic derivatives. Currently these terms are neglected in the model for simplicity and because of the challenges in estimating and verifying the pertinent parameters.

The linear and quadratic drag terms

$$\boldsymbol{D}(\boldsymbol{\nu}_r) = \begin{bmatrix} X_u + X_{u|u|}|u| & 0 & 0 & 0 & 0 & 0 \\ 0 & Y_v + Y_{v|v|}|v| & 0 & 0 & 0 & 0 \\ 0 & 0 & Z_w & 0 & 0 & 0 \\ 0 & 0 & 0 & K_p & 0 & 0 \\ 0 & 0 & 0 & 0 & M_q & 0 \\ 0 & 0 & 0 & 0 & 0 & N_r + N_{r|r|}|r| \end{bmatrix}. \tag{4}$$

which neglects coupleing between sway and yaw.

Each of the hydrodynamic coefficients listed is specified as an SDF[1] configuration file read by the plugin at runtime.

2.1.2 Wave Forcing

2.1.2.1 Wave Field The wave field is generated using simply Gerstner waves[2] with three independent, superimposed wave functions. Each of the component waves is user-specified by an amplitude, period and direction. Deep water dispersion is used in simulating the wave behaviors.

2.1.2.2 Vessel Response To determine the influence of the wave field on the vessel, the vessel footprint is decomposed into a rectilinear grid, with points at each corner of the vessel. The water surface displacement at each grid point is determined based on summation of the component waves. The location of the vessel grid point is determined based on the current pose (position and attitude) of the vessel. The resulting buoyancy force is then determined based on the location of the vessel grid point relative the the water surface and the force is applied directly to the corresponding grid point.

[1] `http://sdformat.org/`

3 Thrust

The external force and torque from the vessel propulsion is implemented in a standalone plugin to allow for independent extension to higher-fidelity thrust configurations and models.

3.1 usv_gazebo_thrust_plugin

The plugin subscribes the `cmd_drive` topic to receive messages of type UsvDrive (defined in the `robotx_gazebo` package). These message specify left and right thrust commands where the commands are scaled from $\{-1.0 - 1.0\}$.

To emulate the thruster behavior, the commands are mapped to a thrust force applied to the model. Two possible mappings are currently available. Users select which mapping using the `mappingType` SDF tag.

The axial force is applied at a point vertically separated from the CG as specified by the `thrustOffsetZ` parameter. This results in coupling between the forward thrust and the vehicle pitch.

3.1.1 0: Linear thruster map

The command values (-1.0 to 1.0) are scaled linearly to the `maxForceFwd` and `maxForceRev` SDF parameters. A total forward thrust is caculated as the sum. A total torque is calculated assuming the two thrust forces are applied at opposite ends of the `boatWidth` parameter.

3.1.2 1: GLF thruster map

In this mode two generalized logistic functions (GLFs) are used to convert commands to thrust force. One set of GLF parameters are used for positive commands (0 to 1.0) and a second set of parameters are used for negative commands (-1 to 0). The form of the GLF used is

$$T = A + \frac{K - A}{(C + \exp(-B(x - M)))^{1/\nu}} \tag{5}$$

where T is the thrust force in Newtons, x is the command and the remaining variables are the GLF parameters. To identify the GLF parameters, the data from [3] was used—Figure1. The Python script for accomplishing this is include in the `usv_gazebo_plugins` repository in the `thrust_curve_fit` directory.

The data was fit with two separate GLF functions using the `scipy.optimize.fmin` optimization routine to minimize the squared error. The results are shown in Figure2.

4 Wind

The influence of wind on the motion of the USV is implemented as a standalone model plugin for Gazebo. Currently there is only one implementation, but other implementations, with varying fidelity, could be developed in the future.

4.1 usv_gazebo_wind_plugin

The wind forces (x and y) and moment (yaw) are predicted following the models presented by Fossen [4].

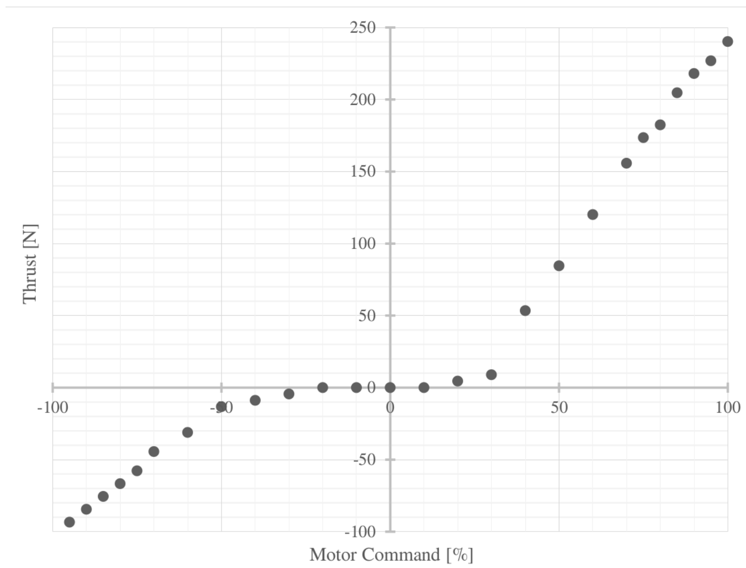

Figure 1: Empirical thrust performance data from [3].

The wind velocity on the vessel ($\mathbf{V}_w$) is considered to be a constant velocity and direction. If desired, this could be extended to include a parameterized wind spectrum the distribution of wind velocities over time, e.g., average wind velocity, gusts, etc. For the current implementation the constant wind velocity is specified as a three element vector which specifies the wind speed the world-frame x, y and z coordinates with units of $^m/_s$. The z component is ignored.

The resulting forces and moments on the vessel are determined based on the user-specified force/moment coefficients and the relative wind velocity. Within the plugin, the relative (or apparent) wind velocity vector $\mathbf{V}_R$. The forces/moment are calculated as

$$\begin{aligned} X_{wind} &= (C_X)V_{R_x}|V_{R_x}| && (6) \\ Y_{wind} &= (C_Y)V_{R_y}|V_{R_y}| && (7) \\ N_{wind} &= -2.0(C_N)V_{R_x}V_{R_y} && (8) \\ & && (9) \end{aligned}$$

where C_X, C_Y and C_N are specified as the three element `wind_coeff_vector`. Approximate values for these coefficients are given in [3] which can then be further tuned to give reasonable response.

References

[1] T. I. Fossen, *Handbook of Marine Craft Hydrodynamics and Motion Control*. Wiley, 2011.

[2] J. Tessendorf, C. C, and J. Tessendorf, “Simulating ocean water,” 1999.

[3] E. I. Sarda, H. Qu, I. R. Bertaska, and K. D. von Ellenrieder, “Station-keeping control of an unmanned surface vehicle exposed to current and wind disturbances,” *Ocean Engineering*, vol. 127, pp. 305 – 324, 2016.

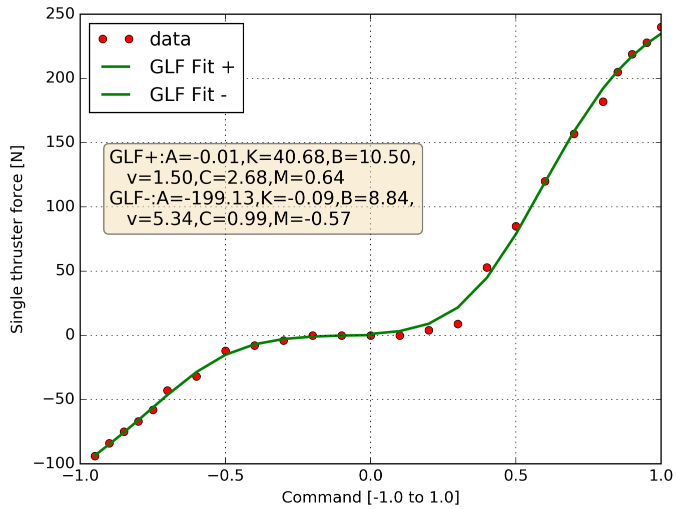

Figure 2: GLF fit of empirical data.

[4] T. I. Fossen, *Guidance and Control of Ocean Vehicles*. Wiley, 1994.