MCsim Python library
Rev 0

Mathias Marley, Department of Marine Technology, NTNU

January 2022

1 Introduction

This note serves as documentation for users and developers of the Marine Cybernetics simulation (MCsim Python) toolbox in Python. The note is not intended as an exhaustive reference, but shall provide:

- Introduction to the toolbox, for users and developers.
- Document the main theory used in the toolbox.
- Provide overview of the models, including source/justification for vessel-specific data and similar.

2 Overview

The toolbox is currently separated into two main libraries:

- **MClib**: library of general functions.
- **MCmod**: library of model-specific functions.

Additional libraries, such as **MCcon** (control systems) are added as needed.

Model and simulation parameters are identified by dictionary objects. Currently there are three main dictionaries:

- **parV:** vessel parameters, e.g. rigid body inertia, fluid inertia, viscous load parameters.
- **parA:** actuator model parameters, e.g. placement of azimuth thrusters, rudder force coefficients.
- **parS:** simulation parameters, e.g. time step, current speed and direction. (Environmental parameters are included in **parS**, since they are considered simulation specific.)

Additional dictionary objects, such as **parC** (control system parameters), are added as needed.

2.1 Model functions

Consider a differential equation

$$\dot{x}(t) = f(x(t), u(t), w(t), p) \tag{1}$$

with state x, control input u, disturbance input w and fixed parameters p. Model functions, contained in **MCmod**, solve a single time step integration of (1), i.e. the solve the equation

$$x(t_{i+1}) = \int_{t_i}^{t_{i+1}} f(x(\tau), u(t_i), w(t_i), p)d\tau, \tag{2}$$

where t_i is current time step, and t_{i+1} is next time step. Note that the inputs u and w are assumed constant over the time interval.

2.2 Using the toolbox

We illustrate the use of the toolbox by an example. The function

x_next=int_RVGMan3_lq(x,u,w,parV,parA,parS),

performs single time step integration of a 3DOF manuevering model of Research Vessel Gunnerus (RVG) using a “linear+quadratic” damping formulation, with commanded thruster states as control input. Here, **x** is the state vector, u is control input, w is disturbance, **parV** and **parA** are model parameters and **parS** are simulation parameters.

To simulate the response of RVG using **int_RVGMan3_lq**, create an initialization script consisting of the following main steps:

1. Load model parameters **parV** and **parA** from relevant pickle file.
2. Specify simulation parameters (including environmental parameters) and store in **parS**.
3. Simulate the response using a for loop that calculates control inputs and calls the function **int_RVGMan3_lq** at each time step.
4. Plot the results.

Model parameters (**parV** and **parA**) are generated by separate scripts and stored using the pickle package. Note that **int_RVGMan3_lq** can model any vessel with two azimuth thrusters as actuators, simply by changing the model parameters.

2.3 Programming tips

As far as possible, abide by the PEP 8 style rules [3]. If you are coming from MATLAB, useful tips are given in [2]. Highlighted tips are listed below:

- Do not use numpy.matrix objects or associated functions.
- Avoid nested numpy arrays. Ensure that functions do not unintentionally return nested numpy arrays.
- For vectors: be aware of the difference between 1D arrays (vectors without orientation), and 2D vertical or horizontal vectors, i.e. 2D arrays with shape (n,1) or (1,n). Extracting a column from a 2D array returns a 1D array. For this reason, aim at using 1D arrays to represent vectors.

3 Mathematical preliminaries

The following definitions are used in the remainder of this document. For a vector $x = [x_1\ x_2\ x_3]^\top \in \mathbb{R}^3$ we define the skew-symmetric matrix cross-product operator $S : \mathbb{R}^3 \to \mathbb{R}^{3\times 3}$ as

$$S(x) := \begin{bmatrix} 0 & -x_3 & x_2 \\ x_3 & 0 & -x_1 \\ -x_2 & x_1 & 0 \end{bmatrix}. \tag{3}$$

We define the selection vectors

$$\varepsilon_1 := \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \quad \varepsilon_2 := \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \quad \varepsilon_3 := \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}. \tag{4}$$

4 General 3DOF maneuvering models

The dynamics of 3DOF manuevering models are calculated using:

- State:
  - **eta**: $\eta := [x\ y\ \psi]^\top \in \mathbb{R}^2 \times [-\pi, \pi]$: Ship pose (position and heading),
  - **nu**: $\nu := [u\ v\ r]^\top \in \mathbb{R}^3$: Body-fixed velocities.
- Input: **F**: $F \in \mathbb{R}^3$ external force vector, including acutator loads
- Parameters: **parV, parS**: vessel and simulation parameters.

4.1 Kinematics

Define the 3DOF rotation matrix about the vertical axis as

$$R_z(\psi) := \begin{bmatrix} \cos(\psi) & -\sin(\psi) & 0 \\ \sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{5}$$

The kinematic equation relating ship pose with body-fixed velocities is given by

$$\dot{\eta} = R_z(\psi)\nu. \tag{6}$$

A steady (time-invariant) and uniform (in space) current is described by

- Current speed: $U_c \in \mathbb{R}$.
- Flow direction: $\beta_c \in [-\pi, \pi]$.

The inertial frame current velocity vector $\nu_c^n \in \mathbb{R}^3$ is given by

$$\nu_c^n := U_c \begin{bmatrix}\cos(\beta_c) & \sin(\beta_c) & 0\end{bmatrix}^\top. \tag{7}$$

The body-fixed current velocity $\nu_c \in \mathbb{R}^3$ becomes

$$\nu_c := [R_z(\psi)]^\top \nu_c^n. \tag{8}$$

The derivative of the body-fixed current velocity is given by

$$\dot{\nu}_c = \frac{d[R_z(\psi)]^\top}{dt}\nu_c^n = r[S(\varepsilon_3)R_z(\psi)]^\top \nu_c^n. \tag{9}$$

4.2 Kinetics

Define the fluid relative velocity $\nu_r \in \mathbb{R}^3$ as

$$\nu_r := \nu - \nu_c. \tag{10}$$

The nonlinear equation of motion of a ship is commonly stated as

$$M_{rb}\dot{\nu} + M_{rb}\dot{\nu}_r + C_{rb}(\nu)\nu + C_a(\nu_r) + D(\nu_r)\nu_r = F, \tag{11}$$

where $M_{rb}, M_a \in \mathbb{R}^{3\times3}$ are the rigid body and added mass matrices, respectively, $C_{rb}, C_a : \mathbb{R}^3 \to \mathbb{R}^{3\times3}$ are the rigid body and added mass Coriolis and centripetal force matrices, and $D : \mathbb{R}^3 \to \mathbb{R}^{3\times3}$ is the hydrodynamic damping matrix. F collects all external forces, e.g. actuator forces, wave and wind loads.

Eq. (11) can be solved using either ν or ν_r as the state. Choosing ν we obtain

$$\dot{\nu} = [M_{rb} + M_a]^{-1}(F - C_{rb}(\nu)\nu - C_a(\nu_r) - D(\nu_r)\nu_r + M_a\dot{\nu}_c), \tag{12}$$

where $\dot{\nu}_c$ is given by (9).

4.2.1 Coriolis and centripetal forces

We assume ships with port-starboard symmetry, and reference point along the center line. This gives inertia matrices of the form

$$M = \begin{bmatrix} m_{11} & 0 & \\ 0 & m_{22} & m_{23} \\ 0 & m_{32} & m_{33} \end{bmatrix}. \tag{13}$$

The rigid body mass matrix is symmetric, while the added mass matrix is in general non-symmetric. The Coriolis and centripetal force matrices are parametrized as

$$C(\nu) := \begin{bmatrix} 0 & 0 & -m_{22}v - 0.5(m_{23} + m_{32})r \\ 0 & 0 & m_{11}u \\ m_{22}u + 0.5(m_{23} + m_{32})r & -m_{11}u & 0 \end{bmatrix}. \tag{14}$$

4.3 Hydrodynamic viscous loads

The function **dot_nu3_man_lq** calculates the derivative of $\nu \in \mathbb{R}^3$, using maneuvering theory, and a "linear+quadratic" damping formulation given by

$$D(\nu_r) := D_l + D_u|u_r| + D_v|v_r| + D_r|r|, \tag{15}$$

where $D_l \in \mathbb{R}^{3\times3}$ is the linear damping matrix, and $D_u, D_v, D_r \in \mathbb{R}^{3\times3}$ are matrices of quadratic damping coefficients. The relation to second-order modulus functions using hydrodynamic derivatives is illustrated by an example. The viscous damping formulation proposed by Blanke (see [1, Section 7.1.2]),

$$D(\nu_r) = \begin{bmatrix} -X_{|u|u}|u_r| & 0 & 0 \\ 0 & -Y_{|v|v}|v_r| & -Y_{|v|r}|v_r| \\ 0 & -N_{|v|v}|v_r| & -N_{|v|r}|v_r| \end{bmatrix} \tag{16}$$

is represented by the matrices $D_l = D_r = 0$, and

$$D_u = \begin{bmatrix} -X_{|u|u} & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad D_v = \begin{bmatrix} 0 & 0 & 0 \\ 0 & -Y_{|v|v} & -Y_{|v|r} \\ 0 & -N_{|v|v} & -N_{|v|r} \end{bmatrix}. \tag{17}$$

5 Actuator models

5.1 First order actuator models

By first order actuator models, we mean actuator dynamics modelled as

$$\dot{x} = -T(x - x_d), \tag{18}$$

where x is the actuator state (e.g. rudder angle), x_d is the desired actuator state, and T is a diagonal matrix of time constants.

5.2 Azimuth thruster loads

Consider an azimuth thruster located at a position $r_t := [x_t \ y_t \ z_t]^\top \in \mathbb{R}^3$ in the body-fixed coordinate system, with thrust force $\tau \in \mathbb{R}$ and azimuth angle $\alpha \in [-\pi, \pi]$. The body-fixed actuator force vector $F : \mathbb{R}^2 \to \mathbb{R}^6$ is given by

$$F(\tau, \alpha) = \tau \begin{bmatrix} R_z(\alpha)\varepsilon_1 \\ S(r_t)R_z(\alpha)\varepsilon_1 \end{bmatrix}, \quad \varepsilon_1 := \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}^\top, \tag{19}$$

which implicitly defines the convention that $\tau > 0$ and $\alpha = 0$ represents a pure surge force in the forward direction. Expanding we obtain

$$F(\tau, \alpha) = \tau \begin{bmatrix} \cos(\alpha) \\ \sin(\alpha) \\ 0 \\ -z_t \sin(\alpha) \\ z_t \cos(\alpha) \\ -y_t \cos(\alpha) + x_t \sin(\alpha) \end{bmatrix}. \tag{20}$$

Note that (19) does not contain “rudder” loads, that are important for foil-shaped azimuth thrusters at forward speeds.

6 RVG maneuvering model

The function **MCmod.int_RVGMan3_lq** models the 3DOF maneuvering equations of RVG, combined with a first order model of the two azimuth thrusters. Thruster forces and azimuth angles are considered control inputs.

Main parameters of RVG are provided in Table 1.

Table 1: Main parameters of RVG

Parameter	Value
Length between perpendiculars	$L_{pp} = 33.9$m
Breadth (moulded)	$B_m = 9.6$m
Draught (moulded)	$d_m = 2.7$ m

6.1 Vessel data

HOLD

6.2 Thruster data

The location of the thrusters, crudely estimated from drawings, are provided in Table 2.

Table 2: Thruster locations of Gunnerus

Parameter	Value	Comment
Longitudinal	0m	Relative to aft perpendicular
Vertical	0.3m	Above baseline
Lateral	± 2.7 m	Relative to centerline

References

[1] Thor I Fossen. *Handbook of marine craft hydrodynamics and motion control.* John Wiley & Sons, 2011.

[2] *numpy for MATLAB users.* `https://numpy.org/doc/stable/user/numpy-for-matlab-users.html`.

[3] *PEP 8 Python style guide.* `https://www.python.org/dev/peps/pep-0008/`.