MCsim Python library
Rev 0

Mathias Marley, Department of Marine Technology, NTNU

January 2022

EXAMPLE DOCUMENTATION, WILL BE UPDATED ONCE TOOLBOX MATURES

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 contains several libraries (modules in Python terminology):

- **kinematics.py**: kinematics functions

3 Models

Model and simulation parameters are identified by dictionary objects. E.g.

- **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.

3.1 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.

4 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.

5 Programming tips

As far as possible, abide by the PEP 8 style rules [2]. If you are coming from MATLAB, useful tips are given in [1]. 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.

6 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

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

References

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

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