RBE 577 - Nonlinear Navigation Filters Assignment

Everett Wenzlaff (ecwenzlaff@wpi.edu)

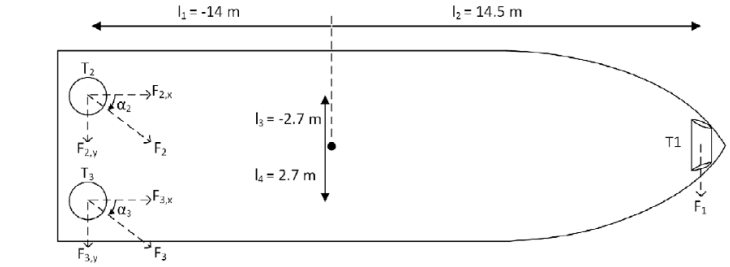
Due: May 7, 2025

1 Introduction

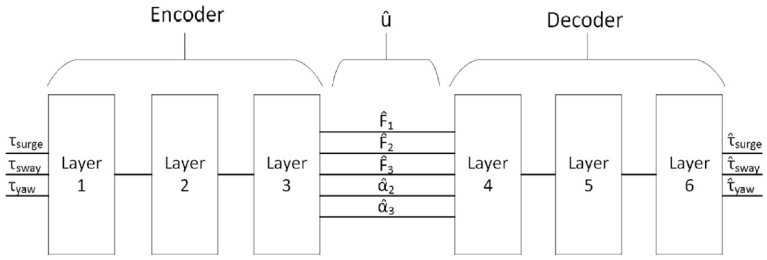
The purpose of this assignment is to implement a control allocator based on the findings from R. Skulstad et. al.'s "Constrained control allocation for dynamic ship positioning using deep neural network" [1]. The primary objective of the paper involved designing a deep neural network (DNN) to perform optimal control allocation for the torque commands received by a ship's motion controller. While we don't have access to all of the data or parameters used for testing the specific implementation, the paper described the basic architecture and sampling methods used to train their DNN. This high level description for architecture and sampling will be used as the starting point for this assignment.

2 Methodology

The basic architecture for the paper's DNN took the form of a symmetrical autoencoder. The autoencoder consisted of a symmetrical 3 layer encoder, and a 3 layer decoder, which each contained two layers of Long Short-Term Memory (LSTM) nodes. While autoencoders are typically used to reduce the dimensionality of the latent space compared to the inputs, the autoencoder from the paper converted 3 "generalized" torques, $[\tau_{surge}, \tau_{sway}, \tau_{yaw}]^T$, into separate forces and angle components within its latent space based on the configuration of the ship (see Figure 1 below). The forces, $[F_1, F_2, F_3]$, and angles, $[\alpha_2, \alpha_3]$, in the latent space represent the control allocator commands, $\hat{u}$, for the ship's thrusters.



(a)



(b)

Figure 1: The autoencoder in (b) converts the generalized torques requested from the ship's motion controller to direct commands for the ship's thrusters, $T1$, $T2$, and $T3$ in (b). Since $T1$ is a non-rotatable thruster, only thrusters $T2$ and $T3$ can receive angle commands (α_2, and α_3).

While the paper described testing the autoencoder using outputs from a high fidelity ship simulator (which cannot be reproduced for this assignment), the autoencoder was trained using randomly generated data. More specifically, the paper described generating data with a randomly initialized random walk for each of the thruster command components, and provided the associated upper and lower bounds. These randomly generated thruster commands were then converted into generalized torque inputs via the following transformation:

$$\begin{bmatrix} \tau_{surge} \\ \tau_{sway} \\ \tau_{yaw} \end{bmatrix} = \begin{bmatrix} 0 & \cos(\alpha_2) & \cos(\alpha_3) \\ 1 & \sin(\alpha_2) & \sin(\alpha_3) \\ l_2 & l_1 \sin(\alpha_2) - l_3 \cos(\alpha_2) & l_1 \sin(\alpha_3) - l_4 \cos(\alpha_3) \end{bmatrix} \cdot \begin{bmatrix} F_1 \\ F_2 \\ F_3 \end{bmatrix}$$

And as part of the autoencoder training, the paper also described a series of different loss functions that were used to drive the output behavior. The separate loss functions were then scaled (according to importance and magnitude) and combined to form a single loss value to facilitate optimization. The individual loss functions can be described as follows:

- $L0$ computed the mean square error (MSE) between the generalized input torques and the torques resulting from directly transforming the encoder's commanded outputs in order to align the inputs with the encoder outputs.
- $L1$ computed the MSE between the generalized input torques and the torques output from the decoder in order to align the inputs with the decoder outputs.
- $L2$ returned values for individual encoder components that exceeded respective predefined limits in order to minimize the magnitude of the encoder commands.
- $L3$ returned values which penalized large rate changes in the encoder commands.
- $L4$ computed an estimate of the power output from the encoder commands in an attempt to minimize power consumption.
- $L5$ returned values based on the encoder command angle outputs in order to minimize the time that the thrusters are commanded to operate within "inefficient azimuth sectors".

So in the spirit of using the paper as a starting point, I started implementing the random walk and autoencoder architecture in Python.

3 Results

4 Discussion

References

[1] Skulstad, R. (2023) (PDF) constrained control allocation for dynamic ship positioning using Deep Neural Network, ResearchGate. Available at: `https://www.researchgate.net/publication/369978600_Constrained_Control_Allocation_for_Dynamic_Ship_Positioning_using_Deep_Neural_Network`.