UNIVERSITY OF TOULON

January 12, 2024

Depth Control of an 'A18M' Autonomous Underwater Vehicle

By:

Abubakar Aliyu Badawi: 22303696

Krittapat Onthuam: 22303709

Tayyab Tahir: 22303713

Ishfaq Bhat: 22303701

Contents

List of Figures

1 Introduction

1.1 Aim

This report intends to outline the process of developing an optimal control command law on the vertical plane to follow a depth command without having too many pitch oscillations of an Autonomous Underwater Vehicle (AUV).

1.2 Objective

To control the depth on the vertical plane, we design a Linear Quadratic Regulator (LQR) controller to replace the existing proportional integral & derivative (PID) controller. The procedures for implementing the LQR controller are outlined below:

1. Identification of missing hydrodynamic parameters using provided data.
2. Linearization of the simplified non-linear AUV model around a fixed point.
3. Designing an LQR controller.
4. Tuning the LQR controller to follow a depth command without having too many pitch oscillations.

1.1 Background

ECARobotics has pioneered the development of Autonomous Underwater Vehicles (AUVs), known as underwater drones, designed to autonomously survey specific areas. These AUVs, exemplified by the "A18M," are equipped with advanced sonar technology, imparting them with the critical function of detecting underwater mines. Given the cost-effectiveness and efficiency of mines in posing threats to military ships and submarines, the need for effective mine detection mechanisms is paramount.

However, the inherent challenge lies in the stability requirements of the AUVs, particularly concerning the vertical and horizontal planes as well as roll stability. The sonar sensor, a pivotal component in mine detection, demands precise stability across all these dimensions for accurate and reliable performance. In response to this challenge, a simplified simulator of an AUV has been developed for this guide.

The report focuses on the AUV A18-M by ECARobotics, designed for detecting underwater mines. The AUV employs sonar technology, requiring high stability at all levels for effective mine detection. Overall, the report addresses the significance of advanced control strategies in optimizing AUV performance in complex underwater environments, particularly for critical applications like mine detection.

2 Parameter Identification

The characterization of hydrodynamic forces is crucial for modeling the behavior of a submerged vehicle experiencing oscillations in undisturbed water. Among these forces are significant factors such as added mass and damping, both playing pivotal roles in the overall hydrodynamic interaction. These forces primarily arise from environmental factors like currents and waves influencing a body in motion

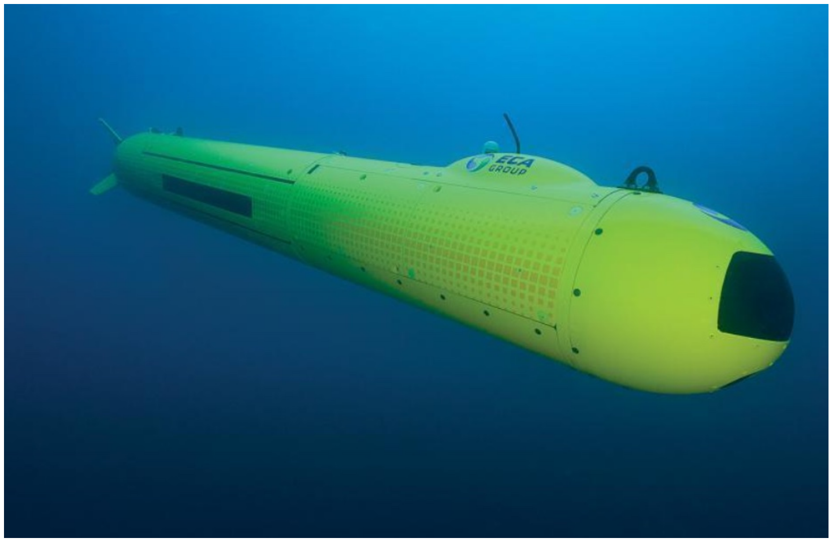

Figure 1: A18M Autonomous Underwater Vehicles

within a water medium. Hydrodynamic damping specifically entails the dissipation of energy as a vibrating body encounters resistance within the system.

In the context of a six-degree-of-freedom Autonomous Underwater Vehicle (AUV), the challenge lies in determining damping parameters, particularly in the horizontal and vertical planes. These parameters are essential for an accurate representation of the hydrodynamic forces acting on the AUV. Various estimation methods have been developed to assess hydrodynamic forces and damping, including slender body theory and parametric identification. It is noteworthy, however, that these methods rely on assumptions, introducing potential limitations to their accuracy. Despite these limitations, they remain valuable tools for approximating the forces influencing the dynamic model of the underwater vehicle. The continuous exploration and refinement of estimation techniques contribute to enhancing our understanding and predictive capabilities in modeling hydrodynamic interactions.

In this instance, our approach involves the experimental estimation of damping parameters by leveraging sensor data acquisition. The data collected from various sensors play a pivotal role in the identification of these damping parameters. The sensor data encompass time series measurements related to:

- Vehicle State: This includes parameters such as position, orientation, as well as the linear and angular velocity of the vehicle.
- Total Force Measurement: The total force acting on the vehicle is deconstructed into components like actuator torques, Coriolis forces, restoring forces, and hydrodynamic forces. Notably, our focus is solely on the hydrodynamic forces within this context.

To facilitate this data-driven estimation process, three distinct missions are conducted to systematically gather the necessary information. During these missions, the sensors capture the dynamic behavior of the vehicle and the associated forces, enabling a comprehensive dataset for subsequent analysis and identification of the elusive damping parameters. This empirical approach ensures a more grounded and realistic representation of the hydrodynamic interactions influencing the vehicle's motion.

- Surge Maneuver: In this mission, the AUV executes pure surge maneuvers, denoted as "u," while maintaining all other velocities negligible. This specific dataset proves instrumental in the identification of the parameter K_{sh} governing surge hydrodynamic forces.

- Pitch Maneuver: During this mission, the AUV performs pitch maneuvers, represented by " θ ." These maneuvers induce speed variations in the form of "w" and "q," while keeping all other velocities negligible. The acquired data play a crucial role in discerning parameters such as **CZ** and **CM** associated with pitch dynamics.

- Yaw Maneuver: The Dieudonnée mission involves virtual helm movements, denoted as "A," leading to speed variations in "v" and " r ." Similar to the previous missions, all other velocities are considered negligible. The dataset obtained from this mission is particularly valuable for identifying parameters such as **CY** and **CN** governing yaw dynamics.

The parameters below are computed in computing_params.m matlab script.

2.1 Obtaining K_{sh}

2.1.1 Dynamic Equation in the Forward Plane

K_{sh} of the AUV is calculated by modeling the drag force F_X as a function of the squared velocity u, the water density ρ, and the AUV's reference surface area S_{ref}. The term C_{XF}, derived from the Reynolds number, adjusts F_X to account for the flow dynamics around the AUV. Employing the least squares method, K_{sh} is determined as the factor that best fits the modeled force to the observed data.

$$F_X = \tfrac{1}{2} \cdot \rho \cdot S_{ref} \cdot C_{X_0} \cdot u \cdot |u|$$

$$C_{X_0} = K_{sh} \cdot C_{XF}$$
$$C_{XF} = \frac{0.075}{(\log_{10}(Re) - 2)^2}$$
$$Re = \frac{u \cdot L_{ref}}{v}$$

$$F_x = K_{sh} \cdot \frac{1}{2} \cdot \rho \cdot S_{ref} \cdot U \cdot |U| \cdot \frac{0.075}{\left(\log_{10}\left(|U| \cdot \frac{L_{ref}}{v}\right) - 2\right)^2}$$

Then $F_x = K_{sh} \cdot x$, where

$$x = \frac{1}{2} \cdot \rho \cdot S_{ref} \cdot U \cdot |U| \cdot \frac{0.075}{\left(\log_{10}\left(|U| \cdot \frac{L_{ref}}{v}\right) - 2\right)^2}$$

We use the Lease Square Method (lsqr) method to get K_{sh} as other parameters and variables are known. K_{sh} is the slope of the line that fit onto (F_x, x) data.

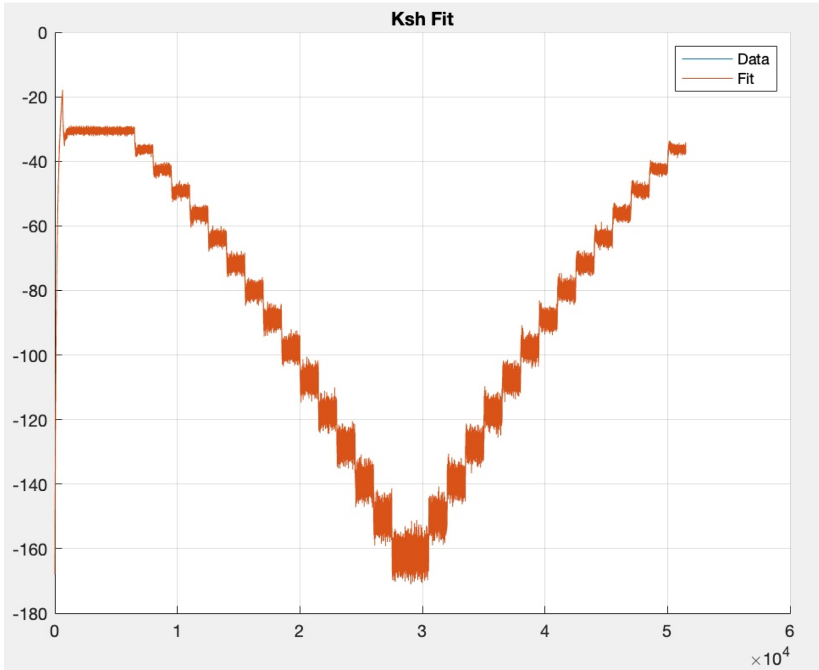

Figure 2: Ksh Fit

The graph shows the comparison between the measured force data and the estimated values from the model. The Ksh Fit title indicates the fitting of the Ksh coefficient over the dataset.

2.2 Obtaining $CZ_{uw}, CZ_{uq}, CM_{uw}, CM_{uq}$

We make use of the data from pitch maneuvers to find CZ (coefficient of heave) and CM (pitching moment) using the lsqr (least squares) technique. Extracted from the pitch step data are the following outcomes:

- State of the vehicle (u, w, q): These parameters encapsulate crucial information about the vehicle's translational and rotational motion during pitch maneuvers. The longitudinal velocity (u), vertical velocity (w), and pitch rate (q) contribute significantly to understanding the dynamic response of the vehicle.
- F_z: Hydrodynamic force in the z-direction (applied for identification): This force component is fundamental in comprehending the vertical forces acting on the vehicle. Accurate identification of F_z aids in modeling and predicting the heave behavior of the vehicle in different operating conditions.
- M_y: Hydrodynamic moment along the y-direction (used for identification): This moment is pivotal in characterizing the rotational dynamics of the vehicle about its vertical axis during pitch maneuvers. Understanding M_y is essential for predicting and controlling the pitching motion of the vehicle.

2.2.1 Obtaining CZ

$$\begin{aligned}
\text{const} &= 0.5 \cdot \rho \cdot S_{\text{ref}} \\
A &= \begin{bmatrix} u \cdot w \cdot \text{const} & L_{\text{ref}} \cdot u \cdot q \cdot \text{const} \end{bmatrix} \\
b &= F_z - \text{CZ}_0 \cdot |u| \cdot u \cdot \text{const} \\
x &= \text{lsqr}(A, b) \\
\text{CZuw} &= x(1) \\
\text{CZuq} &= x(2)
\end{aligned}$$

After using lsqr command to find CZ_{uw} and CZ_{uq} the fit graph is shown below:

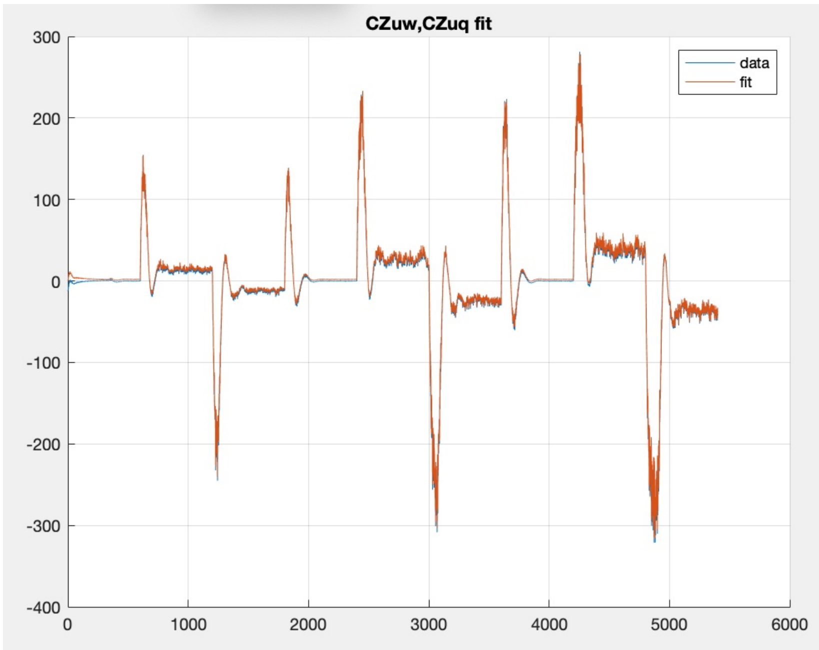

Figure 3: CZuw and CZuq Fit

2.2.2 Obtaining CM

$$\begin{aligned}
My &= \text{forces_values.HydrodynamicForces.My_Nm.Data}(:) \\
\text{const} &= 0.5 \cdot \rho \cdot S_{\text{ref}} \cdot L_{\text{ref}} \\
Ay &= \begin{bmatrix} u \cdot w \cdot \text{const} & L_{\text{ref}} \cdot u \cdot q \cdot \text{const} \end{bmatrix} \\
by &= My - \text{CM}_0 \cdot |u| \cdot u \cdot \text{const} \\
y &= \text{lsqr}(Ay, by) \\
\text{CMuw} &= y(1) \\
\text{CMuq} &= y(2)
\end{aligned}$$

After using lsqr command to find CM_{uw} and CM_{uq} the fit graph is shown below:

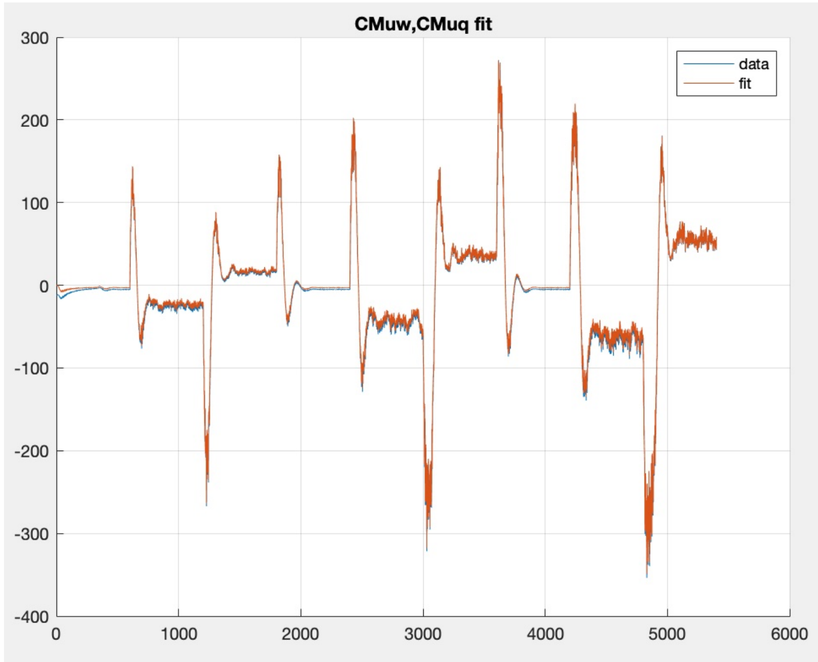

Figure 4: CMuw and CMuq Fit

2.3 Obtaining $CY_{uv}, CY_{ur}, CN_{uv}, CN_{ur}$ using Dieudonnée data

Figuring out these parameters is a bit tricky. If we don't have the right starting values, using the least square method is difficult due to various possible minimum points. So, we turned to the fminsearchcon method, a modified version of the fmincon Matlab method.

To make it work, fminsearchcon needs a function, starting values, and upper and lower bounds. These boundaries and initial values are important to ensure that the method finds the best solution within the given limits. Essentially, they set the boundaries for where the search should happen.

We obtained information from the data, including the vehicle state (u, v, r), hydrodynamic force F_y in the y-direction, and hydrodynamic moment M_z along the z-direction. To simplify the equation, we separate out factors A and B.

$$A = 1/2 * \rho * S_{\text{ref}} * u * v$$
$$B = 1/2 * \rho * S_{\text{ref}} * L_{ref} * u * r$$

2.3.1 Obtaining CY_{uv}, CY_{ur}

In the process of estimating the hydrodynamic coefficients C_{Yuv} and C_{Yur} for an autonomous underwater vehicle (AUV), the following steps were undertaken:

- Loaded data from a MAT file containing vehicle state information and hydrodynamic force data, along with AUV parameters from a JSON file.
- Plotted the water speed (U) over time using the loaded vehicle state data.
- Extracted relevant data for hydrodynamic forces (F_x, F_y) and angular velocity (r).

- Defined physical parameters such as density (ρ), kinematic viscosity (ν), reference length (L_{ref}), and reference surface area (S_{ref}).
- Set up a linear system of equations:

$$A_y = \begin{bmatrix} u \cdot v \cdot 0.5 \cdot \rho \cdot S_{\text{ref}} & L_{\text{ref}} \cdot u \cdot r \cdot 0.5 \cdot \rho \cdot S_{\text{ref}} \end{bmatrix}$$

$$b = F_y$$

- Defined an objective function to minimize the difference between measured forces and estimated forces:

$$\text{fun}_{C_Y}(\mathbf{x}_{C_Y}) = \sum \left(F_y - (u \cdot v \cdot 0.5 \cdot \rho \cdot S_{\text{ref}} \cdot x_{C_{Yuv}} + L_{\text{ref}} \cdot u \cdot r \cdot 0.5 \cdot \rho \cdot S_{\text{ref}} \cdot x_{C_{Yur}})\right)^2$$

- Used the 'fminsearchbnd' optimization function to find the coefficients C_{Yuv} and C_{Yur}:

$$\mathbf{x}_{C_Y} = \text{fminsearchbnd}(\text{fun}_{C_Y}, \mathbf{x}_{0C_Y}, \text{lb}, \text{ub})$$

- Plotted the reference and fitted forces for validation.

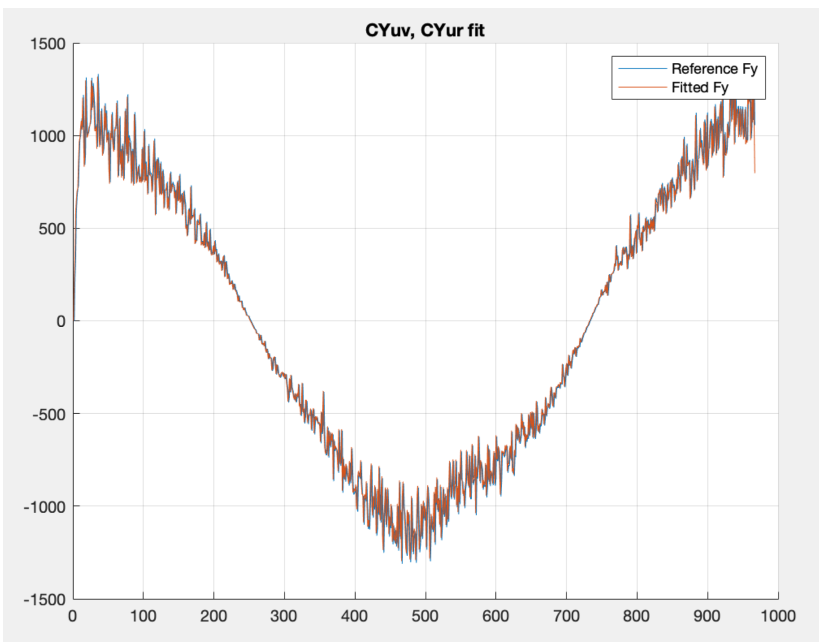

Figure 5: CYuv and CYur Fit

In summary, the hydrodynamic coefficients C_{Yuv} and C_{Yur} for the autonomous underwater vehicle (AUV) were estimated through a systematic process. The analysis involved the formulation of a linear system of equations based on hydrodynamic forces and their corresponding coefficients, incorporating relevant physical parameters. An objective function was defined to minimize the discrepancy between measured and estimated forces using the 'fminsearchbnd' optimization method. The resulting coefficients were obtained as $\mathbf{x}_{C_Y}$, and the validation was performed by plotting the reference and fitted forces over time.

2.3.2 Obtaining CN_{uv}, CN_{ur}

In the process of estimating the hydrodynamic coefficients C_{Nuv} and C_{Nur} for an autonomous underwater vehicle (AUV), the following steps were undertaken:

- Cleared all variables and closed all figures to start with a clean environment.
- Executed a script ('U_plot.m') and added the necessary path.
- Loaded data from a MAT file containing vehicle state information and hydrodynamic force data, along with AUV parameters from a JSON file.
- Plotted the water speed (U) over time using the loaded vehicle state data for visualization.
- Extracted relevant data for hydrodynamic forces (F_x) and angular velocity (r).
- Defined physical parameters such as density (ρ), kinematic viscosity (ν), reference length (L_{ref}), and reference surface area (S_{ref}).
- Set up a linear system of equations:

$$A_n = \begin{bmatrix} u \cdot v \cdot L_{\text{ref}} \cdot 0.5 \cdot \rho \cdot S_{\text{ref}} & L_{\text{ref}}^2 \cdot u \cdot r \cdot 0.5 \cdot \rho \cdot S_{\text{ref}} \end{bmatrix}$$

$$b = M_z$$

- Defined an objective function to minimize the difference between measured yaw moments (M_z) and estimated moments:

$$\text{fun}_{C_N}(\mathbf{x}_{C_N}) = \sum \left(M_z - \left(u \cdot v \cdot L_{\text{ref}} \cdot 0.5 \cdot \rho \cdot S_{\text{ref}} \cdot x_{C_{Nuv}} + L_{\text{ref}}^2 \cdot u \cdot r \cdot 0.5 \cdot \rho \cdot S_{\text{ref}} \cdot x_{C_{Nur}} \right) \right)^2$$

- Used the 'fminsearchbnd' optimization function to find the coefficients C_{Nuv} and C_{Nur}:

$$\mathbf{x}_{C_N} = \text{fminsearchbnd}(\text{fun}_{C_N}, \mathbf{x}_{0C_N}, \text{lb}, \text{ub})$$

- Plotted the reference and fitted yaw moments for validation.

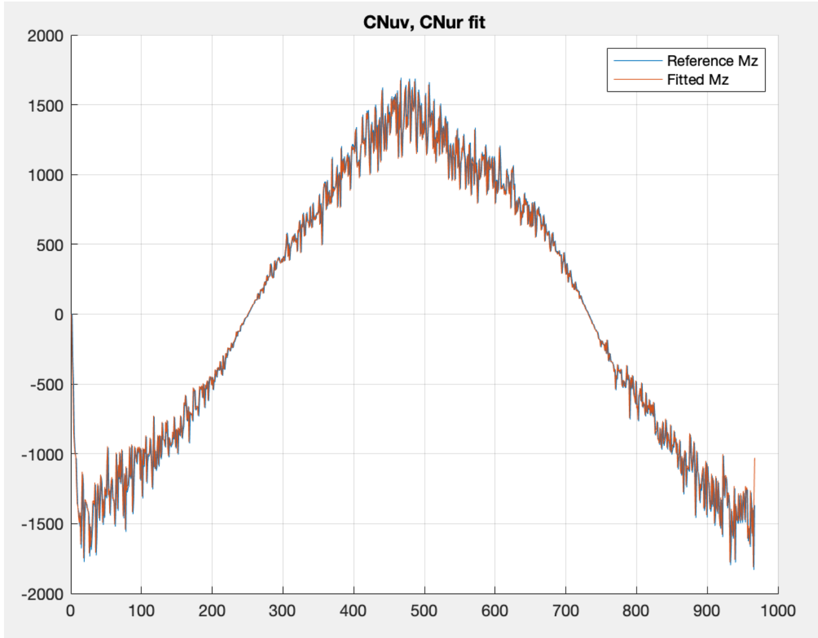

Figure 6: CNuv and CNur Fit

In conclusion, we collectively estimated hydrodynamic coefficients C_{Nuv} and C_{Nur} for an autonomous underwater vehicle. Our approach involved joint data preprocessing, formulation of a linear system of equations, and the use of the 'fminsearchbnd' optimization method. The resulting optimal coefficients were collectively validated through plotting reference and fitted yaw moments, ensuring a reliable understanding of the AUV's yaw dynamics.

3 Linearisation

3.1 Introduction

In addressing the dynamics of a Remotely Operated Vehicle (ROV), it's critical to simplify the system by linearizing the nonlinear dynamic equations at an equilibrium point. This simplification is crucial as it converts complex nonlinear behaviors into a linear form, which significantly eases the control problem. Initially, a PID controller was applied to the AUV's four movement planes, but due to observed instabilities, there's a need to develop a more advanced control mechanism. The focus is thus shifted to enhancing control in the vertical plane for this project.

3.2 The Dynamic Model

The AUV's behavior is governed by a comprehensive nonlinear state equation, which is shown as follows:

$$M\dot{v} = C(v)v + D(v)v + g(\eta) - \tau + w$$

Expanded, the dynamic model becomes:

$$
\begin{bmatrix}
m+X_{\dot{u}} & X_{\dot{v}} & X_{\dot{w}} & X_{\dot{p}} & mz_g+X_{\dot{q}} & -my_g+X_{\dot{r}} \\
Y_{\dot{u}} & m+Y_{\dot{v}} & Y_{\dot{w}} & -mz_g+Y_{\dot{p}} & Y_{\dot{q}} & mx_g+Y_{\dot{r}} \\
Z_{\dot{u}} & Z_{\dot{v}} & m+Z_{\dot{w}} & my_g+Z_{\dot{p}} & -mx_g+Z_{\dot{q}} & Z_{\dot{r}} \\
K_{\dot{u}} & -mz_g+K_{\dot{v}} & my_g+K_{\dot{w}} & I_x+K_{\dot{p}} & I_{xy}+K_{\dot{q}} & I_{xz}+K_{\dot{r}} \\
mz_g+M_{\dot{u}} & M_{\dot{v}} & -mx_g+M_{\dot{w}} & I_{yx}+M_{\dot{p}} & I_y+M_{\dot{q}} & I_{yz}+M_{\dot{r}} \\
my_g+N_{\dot{u}} & mx_g+N_{\dot{v}} & N_{\dot{w}} & I_{zx}+N_{\dot{p}} & I_{zy}+N_{\dot{q}} & I_z+N_{\dot{r}}
\end{bmatrix}
\begin{pmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \\ \dot{p} \\ \dot{q} \\ \dot{r} \end{pmatrix} =
$$

$$
\begin{bmatrix}
0 & 0 & 0 & -mz_g r & -mw & mv \\
0 & 0 & 0 & mw & -mz_g r & -mu \\
0 & 0 & 0 & m\left(z_g p - v\right) & -m\left(z_g q + u\right) & 0 \\
mz_g r & -mw & -m\left(z_g p - v\right) & 0 & -I_z r & I_y q \\
mw & mz_g r & -m\left(z_g q + u\right) & 0 & 0 & -I_x p \\
-mv & mu & 0 & -I_y q & I_x p & 0
\end{bmatrix}
\begin{pmatrix} u \\ v \\ w \\ p \\ q \\ r \end{pmatrix} +
$$

$$
\frac{1}{2}\rho S_{ref}
\begin{bmatrix}
C_{X0}|u| & 0 & 0 & 0 & 0 & 0 \\
0 & C_{Yuv}u & 0 & 0 & 0 & C_{Yur}L_{ref}u \\
C_{Z0}|u| & 0 & C_{Zuw}u & 0 & C_{Zuq}L_{ref}u & 0 \\
0 & 0 & 0 & L_{ref}^2 C_{Lup}u & 0 & 0 \\
L_{ref}C_{M0}|u| & 0 & C_{Muq}L_{ref}u & 0 & L_{ref}^2 C_{Muq}u & 0 \\
0 & C_{Nuv}L_{ref}u & 0 & 0 & 0 & L_{ref}^2 C_{Nur}u
\end{bmatrix}
\begin{pmatrix} u \\ v \\ w \\ p \\ q \\ r \end{pmatrix} +
$$

$$
\begin{bmatrix}
0 \\
0 \\
0 \\
(W-B)\cdot\cos(\theta)\cdot\cos(\psi) \\
-\left(z_g\cdot W - z_b\cdot B\right)\cdot\cos(\theta)\cdot\cos(\psi) \\
-\left(z_g\cdot W - z_b\cdot B\right)\cdot\sin(\theta)
\end{bmatrix} + \tau
$$

3.3 Dynamic Model for Vertical Control

To control the AUV's vertical movements, we focus on certain parts of its motion. We start with a complex equation that involves the vehicle's mass and forces acting on it. This equation helps us understand how the vehicle moves up or down. But for controlling the vehicle in water, we only need to look at some specific types of motion, not all of them. The next part of our work will explain which types of motion we need to control to keep the vehicle stable as it moves vertically.

The dynamic equation governing the vertical control of the Autonomous Underwater Vehicle (AUV) is expressed as follows:

$$
\dot{v} = M^{-1}(C(v)v + D(v)v + g(\eta) - \tau + w)
$$

This formulation yields the first-time derivatives of the six AUV states, represented by the vector:

$$
\dot{v} = \begin{pmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \\ \dot{p} \\ \dot{q} \\ \dot{r} \end{pmatrix}
$$

However, since vertical plane control alone doesn't require the control of all states, the subsequent subsection will elucidate which specific states are pertinent for control in this context.

3.4 Specify the Controlled States

In defining the control law for the vertical plane, several crucial considerations come into play:

1. Controlling the depth (z)
2. Reducing the pitch motion (q)
3. Aiming for minimal deviation from the target depth, especially avoiding oscillatory movements (θ and $\int z$)
4. Stabilizing vertical speed (w)

Considering these aspects, the specific states that require control are denoted by the vector:

$$
X = \begin{pmatrix} w \\ q \\ z \\ \theta \\ \int z \end{pmatrix}
$$

The time derivatives of the states in X are then formulated as:

$$
\dot{X} = \begin{pmatrix} \dot{w} \\ \dot{q} \\ \dot{z} \\ \dot{\theta} \\ \int \dot{z} \end{pmatrix}
$$

3.5 AUV Model Equation

To prepare for linearizing the Autonomous Underwater Vehicle (AUV) model, we've broken down the complex final model equation. This step is crucial for simplifying the equation and making it more manageable for the linearization process. This breakdown helps in understanding and manipulating the equation more effectively for our control system design.

$$
\begin{bmatrix} m+Z_{\dot{w}} & -mx_g+Z_{\dot{q}} \\ M_{\dot{w}} & I_{zy}+N_{\dot{q}} \end{bmatrix} \begin{bmatrix} \dot{w} \\ \dot{q} \end{bmatrix} = \begin{bmatrix} 0 & 0 & m\left(z_g q+u_0\right) \\ mw & -m\left(z_g q+u_0\right) & 0 \end{bmatrix} \begin{bmatrix} u_0 \\ w \\ q \end{bmatrix} +
$$

$$
\frac{1}{2}\rho S_{ref} \begin{bmatrix} C_{Z0}\left|u_0\right| & C_{Z_{uw}u_0} & C_{Z_{uq}L_{ref}u_0} \\ L_{ref}C_{M0}\left|u_0\right| & C_{M_{uw}L_{ref}u_0} & L_{ref}^2 C_{M_{uq}u_0} \end{bmatrix} \begin{bmatrix} u_0 \\ w \\ q \end{bmatrix} +
$$

$$
\begin{bmatrix} (W-B)\cdot\cos(\theta) \\ -\left(z_g\cdot W - z_b\cdot B\right)\cdot\sin(\theta) \end{bmatrix} + \frac{1}{2}\rho S_{ref} \begin{bmatrix} C_Z BAR u_0\left|u_0\right| \\ L_{ref}C_M BAR u_0\left|u_0\right| \end{bmatrix}
$$

In Matlab, this equation is broken down into several main matrices, namely A, B, C, D, and M, as shown below:

$$
\mathbf{A} = \begin{bmatrix} 0 & 0 & m\left(z_g q+u_0\right) \\ mw & -m\left(z_g q+u_0\right) & 0 \end{bmatrix} \begin{bmatrix} u_0 \\ w \\ q \end{bmatrix}
$$

$$
\mathbf{B} = \frac{1}{2}\rho S_{ref} \begin{bmatrix} C_{Z0}\left|u_0\right| & C_{Z_{uw}u_0} & C_{Z_{uq}L_{ref}u_0} \\ L_{ref}C_{M0}\left|u_0\right| & C_{M_{uw}L_{ref}u_0} & L_{ref}^2 C_{M_{uq}u_0} \end{bmatrix} \begin{bmatrix} u_0 \\ w \\ q \end{bmatrix}
$$

$$
\mathbf{C} = \begin{bmatrix} (W-B)\cdot\cos(\theta) \\ -\left(z_g\cdot W - z_b\cdot B\right)\cdot\sin(\theta) \end{bmatrix}
$$

$$
\mathbf{D} = \frac{1}{2}\rho S_{ref} \begin{bmatrix} C_Z BAR u_0\left|u_0\right| \\ L_{ref}C_M BAR u_0\left|u_0\right| \end{bmatrix}
$$

BAR is the input for the control equation. By doing the mass matrix inversion, we obtain the $\dot{X}$

$$
\dot{X} = \begin{pmatrix} \dot{w} \\ \dot{q} \\ \dot{z} \\ \dot{\theta} \\ \int \dot{z} \end{pmatrix} = \begin{pmatrix} \boldsymbol{M}^{-1}(\boldsymbol{A}+\boldsymbol{B}+\boldsymbol{C}+\boldsymbol{D}) \\ -u_0\sin\theta + w\cos\theta \\ q \\ z \end{pmatrix}
$$

3.6 Set the Equilibrium Point

Equilibrium point is a point where the state equation becomes zero when evaluated at that point.

$$
f\left(x^e, u^e\right) = 0
$$

Applying this to our state equation, we will get:

$$f\left(x^e, u^e\right) = \begin{pmatrix} \boldsymbol{M}^{-1}(\boldsymbol{A}+\boldsymbol{B}+\boldsymbol{C}+\boldsymbol{D}) \\ -u_0 \sin\theta + w\cos\theta \\ q \\ z \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

Solving this equation, we will get several equilibrium points:

$$\begin{bmatrix} \dot{w} \\ \dot{q} \end{bmatrix} = \boldsymbol{M}^{-1}(\boldsymbol{A}+\boldsymbol{B}+\boldsymbol{C}+\boldsymbol{D}) = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$$\dot{z} = -u_0 \sin\theta + w\cos\theta = 0$$

$$\dot{\theta} = q = 0$$

From equilibrium point (1), with $u_0 = 2$ m/s and $q_e = 0$, we will get the value of BAR and θ.

$$BAR_e = -0.03048$$

$$\theta_e = -0.19917$$

From equilibrium point (2), using θ. we can find the value of w.

$$w_e = u_0 \tan\theta = 2\tan(-0.19917) = -0.4037$$

3.7 Computing The Jacobian Matrix

To evaluate the $\dot{x}$, we need to compute the corresponding continuous tangent of the linearized control system. Jacobian Matrix is used to solve this problem.

$$\frac{\partial f}{\partial x} = \begin{bmatrix} \frac{\partial f_1}{x_1} & \frac{\partial f_1}{x_2} & \frac{\partial f_1}{x_3} & \frac{\partial f_1}{x_4} & \frac{\partial f_1}{x_5} \\ \frac{\partial f_2}{x_1} & \frac{\partial f_2}{x_2} & \frac{\partial f_2}{x_3} & \frac{\partial f_2}{x_4} & \frac{\partial x_2}{x_5} \\ \frac{\partial f_3}{x_1} & \frac{\partial f_3}{x_2} & \frac{\partial f_3}{x_3} & \frac{\partial f_3}{x_4} & \frac{\partial f_3}{x_5} \\ \frac{\partial f_4}{x_1} & \frac{\partial f_4}{x_2} & \frac{\partial f_4}{x_3} & \frac{\partial f_4}{x_4} & \frac{\partial f_4}{x_5} \\ \frac{\partial f_5}{x_1} & \frac{\partial f_5}{x_2} & \frac{\partial f_5}{x_3} & \frac{\partial f_5}{x_4} & \frac{\partial f_5}{x_5} \end{bmatrix}$$

3.8 Evaluate Jacobian at Equilibrium Point

To determine the system's behavior around its normal operating condition, we calculate the Jacobian matrix at the equilibrium point. This involves taking partial derivatives of the system's equations with respect to each state variable, evaluated at the point where the vehicle is in a steady state (equilibrium). The resulting Jacobian matrix provides insights into the system's stability and responsiveness, aiding in the design of control systems that effectively manage the AUV's dynamics in its typical operational environment. After we have got the Jacobian and all of the equilibrium points, we can then evaluate Jacobian at these equilibrium points to get the **A** and **B** matrices for our control equation. For initialization, we also substitute the initial velocity u_0 to the Jacobian. In Matlab, we create the Jacobian Matrix using our pre-made Jacobian function. Solving the equation in Matlab, we will obtain the **A** and **B** Matrix as shown below:

$$\mathbf{A} = \begin{pmatrix} -0.3102 & -1.034 & 0 & 0 & 0 \\ 1.306 & -7.835 & 0 & -0.0383 & 0 \\ 1.0000 & 0 & 0 & -2.0000 & 0 \\ 0 & 1.0000 & 0 & 0 & 0 \\ 0 & 0 & 1.0000 & 0 & 0 \end{pmatrix}$$

$$\mathbf{B} = \begin{pmatrix} -0.04136 \\ -0.1567 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

3.9 LQR Command

The Linear-Quadratic Regulator (LQR) method calculates the optimal feedback gain matrix K. It minimizes a cost function J, integrating the state vector y and control input u:

$$J = \int_0^\infty \left[y^T Q y + u^T R u \right] dt$$

The MATLAB 'lqrd' command is used for discrete full-state feedback regulation from a continuous system.

3.10 Controllability

System controllability is essential for LQR effectiveness. The controllability matrix C_o must have full rank:

$$C_o = \left[B | AB | A^2 B | \dots | A^{n-1} B \right]$$

The controllability has full rank, that is, the rank is equal to the number of states in the state-space model. In addition, there is a MATLAB command ctrb that takes A and B to generate the controllability matrix and then checks the rank of the matrix and we found that our controllability matrix has a rank of 5 which equals the number of states. We used LQRD command to get a discrete full-state-feedback regulator from a continuous system. We don't need to discretize before using it. We used matrices A and B from the state space model. $\mathrm{Kd} = \mathrm{lqrd}(\mathrm{A}, \mathrm{B}, \mathrm{Q}, \mathrm{R})$

4 Result and Discussion

4.1 Tuning of the LQR

The simulator-based control of the AUV hinges on the equilibrium point $[w_e, q_e, \theta_e, BAR_e]'$ and the gain matrix K. Variations in the model's equation affect both the equilibrium point and Jacobian matrices, thereby impacting the K matrix. The tuning involved testing two model equations in the simulator, using the symbolic toolbox in 'control.m' for calculating equilibrium points, Jacobian matrices, and fine-tuning the K matrix.

4.2 Tuning with $\dot{z} = w$ in the model equation

Setting $w = \dot{z}$ in the model equation becomes:

$$\begin{pmatrix} \dot{w} \\ \dot{q} \\ \dot{z} \\ \dot{\theta} \\ \int \dot{z} \end{pmatrix} = \begin{pmatrix} \boldsymbol{M}^{-1}(\boldsymbol{A} + \boldsymbol{B} + \boldsymbol{C} + \boldsymbol{D}) \\ w \\ q \\ z \end{pmatrix}$$

After computing the equilibrium points for this model equation, we obtained:

$$\begin{pmatrix} w_e \\ q_e \\ \theta_e \\ BAR_e \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ -0.1992 \\ -0.0305 \end{pmatrix}$$

The Jacobian Matrices obtained from the new model equation at the equilibrium point are:

$$A = \begin{pmatrix} -0.4375 & -0.6383 & 0 & 0.0000 & 0 \\ 0.2385 & -1.1326 & 0 & -0.0391 & 0 \\ 1.0000 & 0 & 0 & 0 & 0 \\ 0 & 1.0000 & 0 & 0 & 0 \\ 0 & 0 & 1.0000 & 0 & 0 \end{pmatrix} \quad B = \begin{pmatrix} -0.1136 \\ -0.0888 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

This values of A and B, together with the default values provided where $Q = \text{diag}([1e{-}\,6, 1, 100, 10, 0.1])$ and $R = \text{diag}([1e9])$, are used to tune the LQR. The K matrix obtained:

$$K = \begin{pmatrix} -0.0366 & 0.0013 & -0.0164 & -0.0219 & -0.0000 \end{pmatrix}$$

The result obtained is provided in the Figure below.

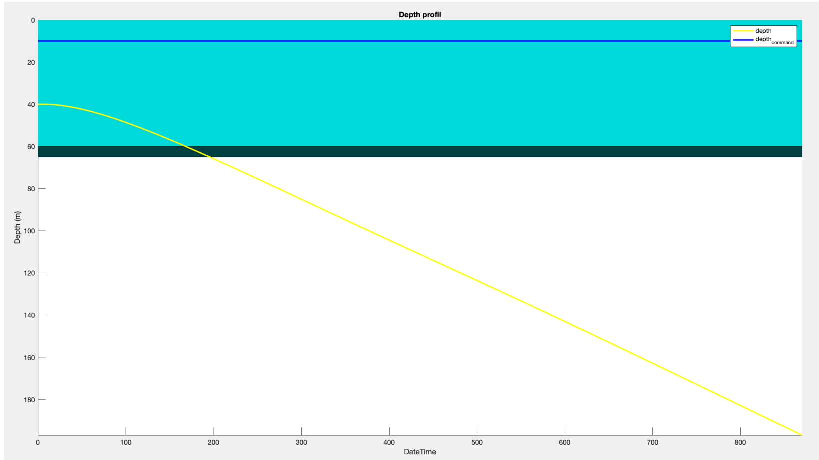

Figure 7: Depth Profile of AUV with $\dot{z} = w$ in the model equation

After several attempts to improve this output as well through tuning, the efforts all proved futile once again because the results never moved close to the desired value. Hence, we attempted to look for another way to tune the model.

4.3 Tuning with $\dot{z} = -u_0 \sin\theta + w\cos\theta$ in the model equation

Utilizing the formula $\dot{z} = -u_0 \sin\theta + w\cos\theta$ in the model equation, we have the model equation:

$$\begin{pmatrix} \dot{w} \\ \dot{q} \\ \dot{z} \\ \dot{\theta} \\ \int \dot{z} \end{pmatrix} = \begin{pmatrix} \boldsymbol{M}^{-1}(\boldsymbol{A} + \boldsymbol{B} + \boldsymbol{C} + \boldsymbol{D}) \\ -u_0 \sin\theta + w\cos\theta \\ q \\ z \end{pmatrix}$$

After computing the equilibrium points, we obtained:

$$\begin{pmatrix} w_e \\ q_e \\ \theta_e \\ BAR_e \end{pmatrix} = \begin{pmatrix} 0.0138 \\ 0 \\ 0.0069 \\ -0.0837 \end{pmatrix}$$

The Jacobian Matrices obtained are:

$$A = \begin{pmatrix} -0.4375 & -0.6383 & 0 & 0.0000 & 0 \\ 0.2385 & -1.1326 & 0 & -0.0391 & 0 \\ 1.0000 & 0 & 0 & -2.0000 & 0 \\ 0 & 1.0000 & 0 & 0 & 0 \\ 0 & 0 & 1.0000 & 0 & 0 \end{pmatrix} \quad B = \begin{pmatrix} -0.1136 \\ -0.0888 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

This values of A and B, together with the default values provided where $Q = \text{diag}([1e\!-6, 1, 100, 10, 0.1])$ and $R = \text{diag}([1e9])$, are used to tune the LQR. The K matrix obtained:

$$K = \begin{pmatrix} -0.0710 & -0.1506 & 0.0041 & -0.2175 & 0.0000 \end{pmatrix}$$

The result obtained is provided in the Figure below. We could not improve the output using this

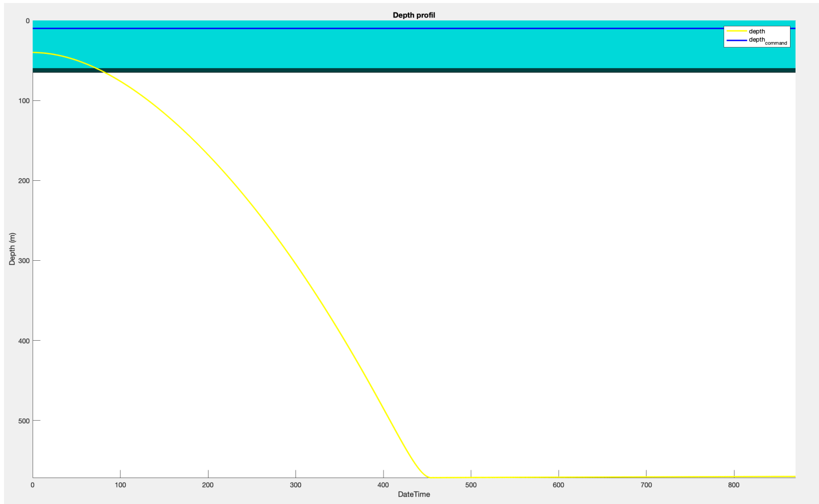

Figure 8: Depth Profile of AUV with $\dot{z} = -u_0 \sin\theta + w\cos\theta$ in the model equation

model equation, because the result was never close to the desired point, hence we proposed to merge the two model equations together.

4.4 Tuning while combining the two approaches

After several futile attempts, we proceeded to try controlling the system at the equilibrium point of the first model where $\dot{z} = -u_0 \sin\theta + w\cos\theta$ and we linearized the model and tuned using $\dot{z} = w$. We attempted this to find a solution to the control problem. Hence, the model equation for the equilibrium point:

$$\begin{pmatrix} \dot{w} \\ \dot{q} \\ \dot{z} \\ \dot{\theta} \\ \int \dot{z} \end{pmatrix} = \begin{pmatrix} \boldsymbol{M}^{-1}(\boldsymbol{A}+\boldsymbol{B}+\boldsymbol{C}+\boldsymbol{D}) \\ -u_0 \sin\theta + w\cos\theta \\ q \\ z \end{pmatrix}$$

After computing the equilibrium points for this model equation, we obtained:

$$\begin{pmatrix} w_e \\ q_e \\ \theta_e \\ BAR_e \end{pmatrix} = \begin{pmatrix} 0.01377 \\ 0 \\ 0.00689 \\ -0.08367 \end{pmatrix}$$

Afterwards, the Jacobian matrices and the remaining aspects till the values of K matrix are gotten are obtained using another model equation:

$$\begin{pmatrix} \dot{w} \\ \dot{q} \\ \dot{z} \\ \dot{\theta} \\ \int \dot{z} \end{pmatrix} = \begin{pmatrix} \boldsymbol{M}^{-1}(\boldsymbol{A}+\boldsymbol{B}+\boldsymbol{C}+\boldsymbol{D}) \\ w \\ q \\ z \end{pmatrix} \quad \begin{pmatrix} w_e \\ q_e \\ \theta_e \\ BAR_e \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ -0.19917 \\ -0.03048 \end{pmatrix}$$

The Jacobian Matrices obtained from this model equation at its equilibrium point are:

$$A = \begin{pmatrix} -0.4375 & -0.6383 & 0 & 0.0001 & 0 \\ 0.2385 & -1.1326 & 0 & -0.0383 & 0 \\ 1.0000 & 0 & 0 & 0 & 0 \\ 0 & 1.0000 & 0 & 0 & 0 \\ 0 & 0 & 1.0000 & 0 & 0 \end{pmatrix} \quad B = \begin{pmatrix} -0.1136 \\ -0.0888 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

This values of A and B, together with the default values provided where $Q = \mathrm{diag}([1e-6, 1, 100, 10, 0.1])$ and $R = \mathrm{diag}([1e9])$, are used to tune the LQR. The K matrix obtained:

$$K = \begin{pmatrix} -0.0204 & 0.0005 & -0.0090 & -0.0125 & -0.0000 \end{pmatrix}$$

The result obtained from simulating this value of K matrix with the earlier stated value of equilibrium point is provided in Figure below.

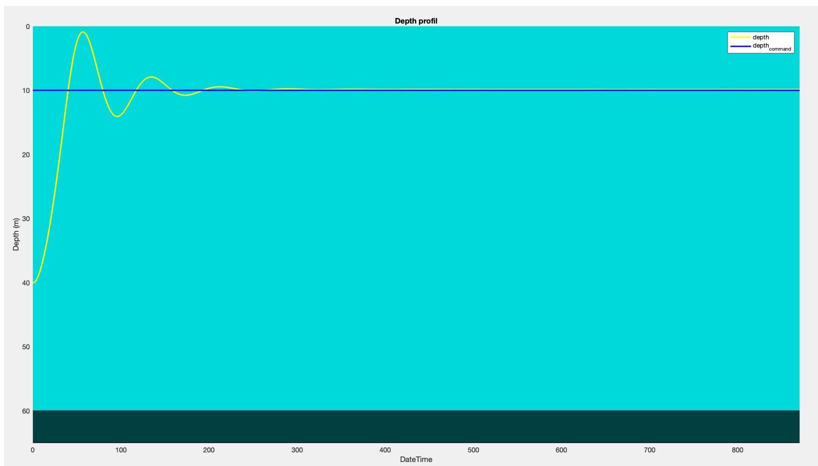

Figure 9: Depth Profile of AUV by combining the 2 approaches

5 Best Result Obtained

Following the execution of simulations using the previously mentioned parameters, we observed initial outcomes that delineated the modeled system's behavior. These initial findings were positive. For enhanced results, we proceeded to adjust the tuning by setting $Q = \mathrm{diag}[1e8, 1e8, 350, 450, 1e-5]$ and $R = \mathrm{diag}([1e8])$.

The K matrix obtained:

$$K = \begin{pmatrix} -0.0889 & -0.0170 & -0.0025 & -0.0061 & -0.0000 \end{pmatrix}$$

5.1 Depth Profile

The trajectory of the AUV's vertical path is depicted. Transitioning from a depth of 40 m to 10 m, the AUV took about 212 seconds to descend to 40 m and approximately 379 seconds to achieve minimal oscillations.

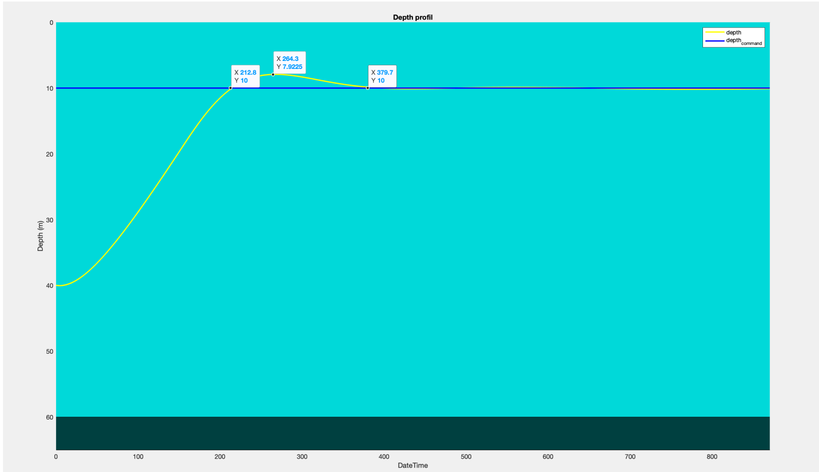

Figure 10: Depth profile after tunning further

5.2 XY Plot

This shows the trajectory of the AUV as viewed from the top of the ocean

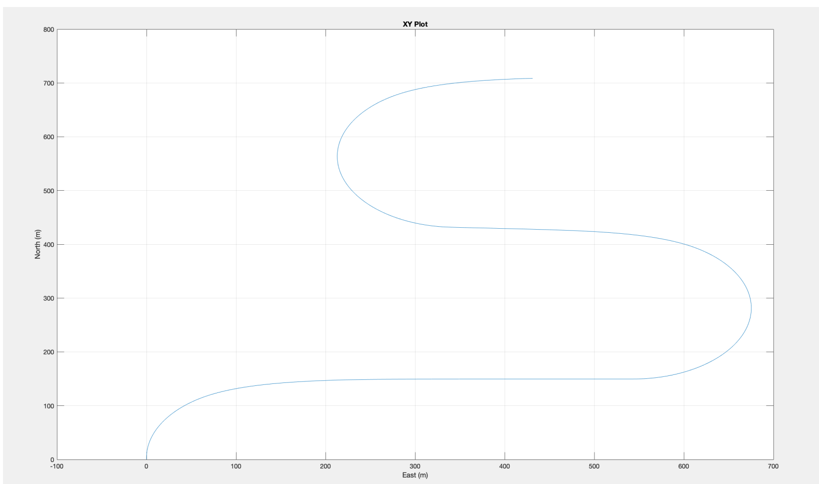

Figure 11: The XY Trajectory

5.3 Euler angles

In the AUV's vertical navigation, while roll (ϕ) and yaw (ψ) angles remain minor, the pitch angle (θ) is significant. This angle is pivotal for pitch movements. Adjusting the thrusters to a positive pitch angle induces ascent. As the AUV nears its target depth, post 200 seconds, the pitch angle reduces, aligning to zero around 300 seconds, coinciding with reaching the 10-meter depth.

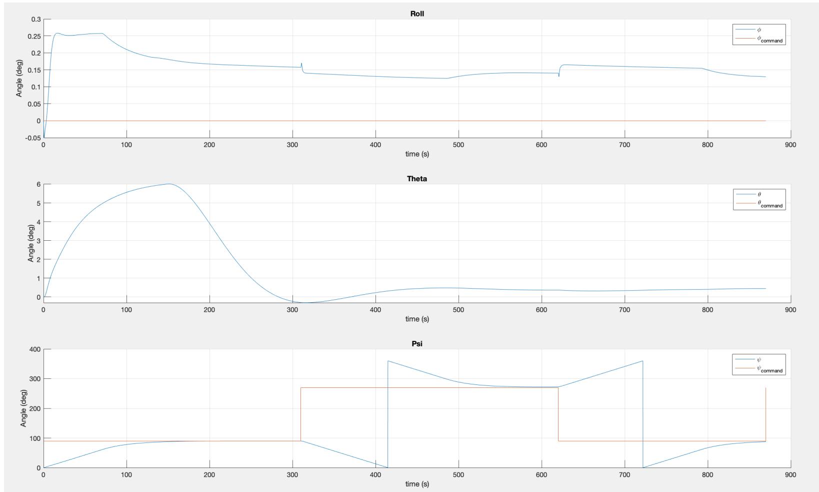

Figure 12: The Euler Angle Plots

5.4 Linear speed

The AUV's longitudinal speed is maintained around 2 m/s, aiding its forward motion. With an inclined angle θ, this speed facilitates vertical ascent. The minor lift speed plays a subtle role in vertical control, adjusting until reaching the 10-meter depth, stabilizing around 380 seconds. Remarkably, the lift speed remains oscillation-free, owing to the low 'w' value.

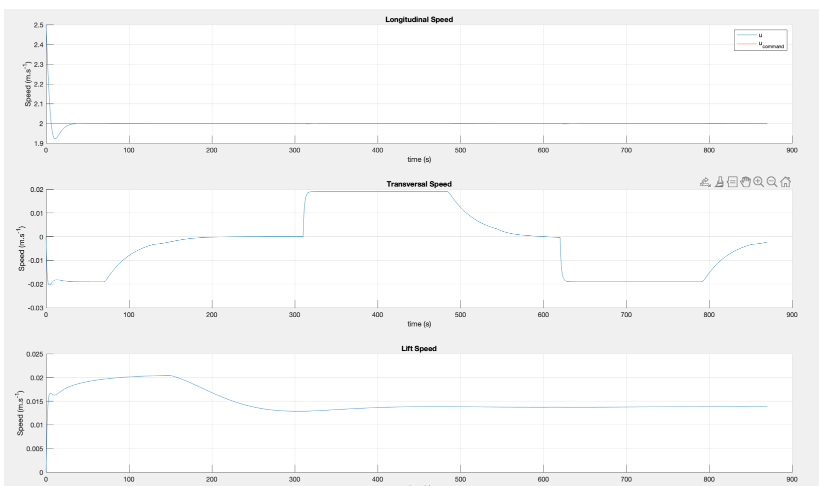

Figure 13: AUV'S Linear Speed Plot

5.5 Angular Speed

The pitch rate graph illustrates the AUV's attitude adaptation for reaching the specified depth. This graph demonstrates the pitch angle's rate of change in response to the depth command, exhibiting few oscillations due to the low 'q' value.

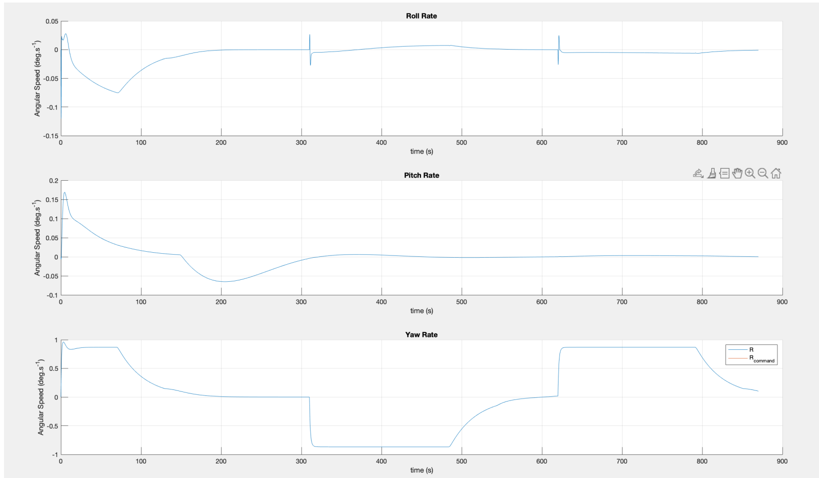

Figure 14: AUV's Angler Speed

5.6 Actuators

The BAR actuator, managing the AUV's vertical navigation, shows a smooth plot in its operational graph. This indicates non-aggressive actuator responses, crucial for avoiding undue wear and potential damage to the system.

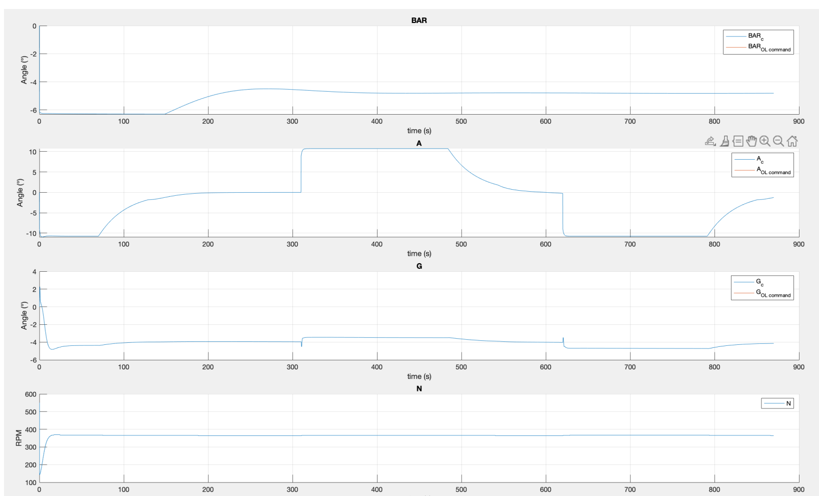

Figure 15: AUV's Actuators

6 Code Guide

These are the list of files and folders added to the report and a description of what they mean:

- `control.m`: Implementation of the LQR.
- `param_identification.m`: Model identification.
- `PilotDepth_LQR.m`: Depth command.
- `Plots`: screenshots of result plots.

References

[1] "Guide to Practical Work," [PDF document].

[2] "AUV Model Equation," [PDF document].

[3] Joo, M. G. (2019). "A controller comprising tail wing control of a hybrid autonomous underwater vehicle for use as an underwater glider." *International Journal of Naval Architecture and Ocean Engineering*

[4] Geranmehr, B., & Rafee Nekoo, S. (2014, October). "The nonlinear suboptimal diving control of an autonomous underwater vehicle." *Proceedings of the International Conference on Robotics and Mechatronics (ICRoM)*