Marmara University
Faculty of Engineering

DEVELOPMENT OF CASCADED NONLINEAR DYNAMIC INVERSION MOTION CONTROLLER FOR A TORPEDO-TYPE AUTONOMOUS UNDERWATER ROBOT

MURAT TAHA YEMİŞÇİ

GRADUATION PROJECT REPORT
Department of Mechanical Engineering

Supervisor
Associate Professor Mehmet Berke GÜR

ISTANBUL, 2026

Marmara University

Faculty of Engineering

Development of Cascaded Nonlinear Inversion Motion Controller for a Torpedo Type Autonomous Underwater Robot

by

Murat Taha Yemişçi

January 19, 2026, Istanbul

SUBMITTED TO THE DEPARTMENT OF MECHANICAL ENGINEERINGIN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE

OF

BACHELOR OF SCIENCE
AT

MARMARA UNIVERSITY

Signature of Author(s) .....
Department of Mechanical Engineering

Certified By .....
Project Supervisor, Department of Mechanical Engineering

Accepted By ....
Head of the Department of Mechanical Engineering

ACKNOWLEDGEMENT

I would like to express my deepest gratitude to my supervisor, Associate Professor Mehmet Berke Gür, for his invaluable guidance, mentorship, and continuous encouragement throughout the course of this thesis. His profound knowledge in control theory and his insightful feedback played a pivotal role in shaping this study, particularly in navigating the complexities of nonlinear dynamic inversion and zero dynamics analysis.

Finally, I would like to thank my family for their unwavering moral support, patience, and belief in me throughout my academic journey.

January, 2026 **Murat Taha Yemişçi**

Table of Contents

ABBREVIATIONS

AUV	: Autonomous Underwater Vehicle
BIBO	: Bounded-Input Bounded-Output
DOF	: Degrees of Freedom
LHP	: Left Half Plane
LOS	: Line-of-Sight (Guidance Algorithm)
MIMO	: Multi-Input Multi-Output
MP	: Minimum Phase
NDI	: Nonlinear Dynamic Inversion
NED	: North-East-Down (Coordinate Frame)
NMP	: Non-Minimum Phase
RHP	: Right Half Plane
RPM	: Revolutions Per Minute
SISO	: Single-Input Single-Output
TSS	: Time-Scale Separation
UAV	: Unmanned Aerial Vehicle

LIST OF FIGURES

LIST OF TABLES

1. Introduction

About 75% of the Earth's surface is occupied by water, with most of the Earth's surface water being ocean water [1]. However, it is estimated that about 95% of the world's oceans have yet to be explored [2]. The largely unexplored parts of the planet are thought to have diverse resources, including new sources of food, renewable energy, and raw materials. The main challenge in the exploration of the underwater world is the extreme nature of the conditions found in this environment.

Recently, Autonomous Underwater Vehicles (AUVs) have appeared as an important area in the exploration of these frontier regions. Compared to tethered systems, AUVs are capable of complex operations without human control in real time. This makes it possible to apply these systems in dangerous regions. The applications of AUVs include three-dimensional mapping of the seafloor [3], geomorphological surveys [4], rapid environment assessments [5], inspections of underwater pipelines [6], as well as several military missions like mine clearance. Autonomy is important in furthering the understanding of underwater ecosystems as well as in maritime asset protection.

Despite this, the successful use of AUVs relies greatly on the quality of their motion control systems in terms of precision and robustness. The underwater environment has its unique characteristics that make it unsuitable for the use of simple linear control systems like the proportional-integral-derivative controller for high-performance tasks. AUVs can be described as highly nonlinear mechanical systems with strong coupling between the DOF. They are also prone to complex hydrodynamic forces and external disturbances like ocean currents. Additionally, the torpedo-type AUVs like the REMUS 100 are underactuated systems with less actuators (propeller, rudder, and stern plane) than the number of DOF they control. The current research aims at the design and realization of a high-level motion control scheme for a torpedo-type AUV based on Nonlinear Dynamic Inversion (NDI) techniques. The Nonlinear Dynamic Inversion approach uses the mathematical model of the AUV to algebraically eliminate the nonlinearity in the dynamics of the AUV over its whole flight regime. In particular, the current research focuses on the non-minimum phase characteristics of the AUV during depth control maneuvers using a Three-Level Cascaded Control Structure that adopts the concept of Time-Scale Separation. This approach combines a LOS guidance law with inner-loop Nonlinear Dynamic Inversion control structures.

2. Literature Review

The control of autonomous vehicles operating in fluid environments presents significant challenges due to inherent nonlinearities, coupled dynamics, and environmental uncertainties. Nonlinear Dynamic Inversion (NDI), also referred to as feedback linearization, has emerged as a powerful control strategy to address these challenges by algebraically canceling nonlinear dynamics to enforce a desired linear response. This section reviews the development of NDI, its stability considerations in non-minimum phase systems, and its applicability to Autonomous Underwater Vehicles (AUVs).

2.1.Foundations and Aerospace Applications of NDI

The theoretical foundation of using nonlinear transformations for automatic control was significantly advanced by Meyer et al. [11], who demonstrated that global transformations could render a nonlinear system equivalent to a controllable linear system, thereby overcoming the limitations of local linearization points used in traditional gain scheduling. Building on this, Lane and Stengel [12] formalized the design of flight control systems using nonlinear inverse dynamics, highlighting the method's superior ability to handle the full flight envelope compared to conventional linear controllers.

In recent years, NDI has been extensively applied to high-performance aircraft. Akyüz et al. [8] investigated the maneuvering control of F-16 aircraft, demonstrating that when NDI is augmented with Model Reference Adaptive Control (MRAC), it can successfully execute high-angle-of-attack maneuvers that would destabilize conventional linear controllers. Similarly, Albostan and Gökaşan [13] applied NDI combined with Eigenstructure Assignment to manage the coupled dynamics of F-16s during aggressive maneuvering. To address the issue of model sensitivity van 't Veld [14] proposed "Incremental Nonlinear Dynamic Inversion" (INDI). This approach uses sensor measurements of the state derivatives to reduce dependency on the system model, offering a robust alternative for systems with uncertain aerodynamic or hydrodynamic parameters.

2.2. Stability in Non-Minimum Phase Systems

A critical limitation of standard dynamic inversion arises when the system exhibits non-minimum phase behavior (unstable zero dynamics). In such cases, a direct inversion of the input-output dynamics leads to unbounded internal states, causing system failure. This is particularly relevant for tail-controlled missiles and VTOL (Vertical Take-Off and Landing) aircraft, which share dynamic similarities with the

pitch control of torpedo-type AUVs.

Devasia [1] and Devasia et al. [2] addressed this by developing an iterative inversion approach for nonlinear non-minimum phase systems. Their work on VTOL aircraft demonstrated that stable tracking can be achieved by computing a non-causal feedforward input, effectively managing the internal instability. Complementing this, Chen and Paden [3] introduced a stable inversion technique that separates the system dynamics into stable and unstable manifolds, solving for bounded trajectories offline. More recently, Ye et al. [6] proposed an "Output Redefinition-Based Dynamic Inversion" for hypersonic vehicles. By redefining the output to a point where the zero dynamics become minimum-phase, they successfully stabilized the vehicle without requiring complex iterative solutions.

2.3. Technology Transfer to Autonomous Underwater Vehicles (AUVs)

While the majority of NDI research focuses on aerospace applications, the theoretical framework is directly transferable to AUVs. Both aircraft and underwater vehicles are governed by 6-Degree-of-Freedom (6-DOF) equations of motion characterized by strong coupling and nonlinear damping forces.

The "Physical Dynamic Inversion" approach discussed by Zhang and Holzapfel [9] emphasizes model-based control that respects physical constraints, a concept highly applicable to the actuator limitations of the REMUS AUV. Furthermore, the adaptive techniques explored by Harris and Valasek [10] provide a pathway to handle the varying hydrodynamic coefficients (added mass and drag) encountered by underwater vehicles.

In this project, the robust NDI methodologies established in the cited aerospace literature will be adapted for the REMUS AUV. Specifically, the similarities between aerodynamic stability derivatives and hydrodynamic coefficients will be leveraged to design a motion controller capable of managing the vehicle's nonlinear underwater dynamics.

3. Mathematical Modeling of REMUS 100 AUV

The motion of an AUV can be studied in two stages, the first one is the kinematic details like positions, angular displacements, velocities that are considered. The other approach is dynamic details like forces and moments that drive the system to make its motion.

3.1. Kinematics of The Vehicle

When the motion of an underwater vehicle is inspected, there are two coordinate systems that are taken into consideration. The first one is referred to as body-fixed coordinate system, where the motion of the underwater vehicle is referenced into the vehicle's center of buoyancy. The origin of this coordinate system moves along with the AUV during its motion. Besides this, there are 6 different dimensional motion types allowing the AUV to move. Surge, sway and heave velocities represent linear motion and roll, pitch and yaw velocities represent rotational motion with respect to the body-fixed reference system. The visualization is given in Figure 1 below.

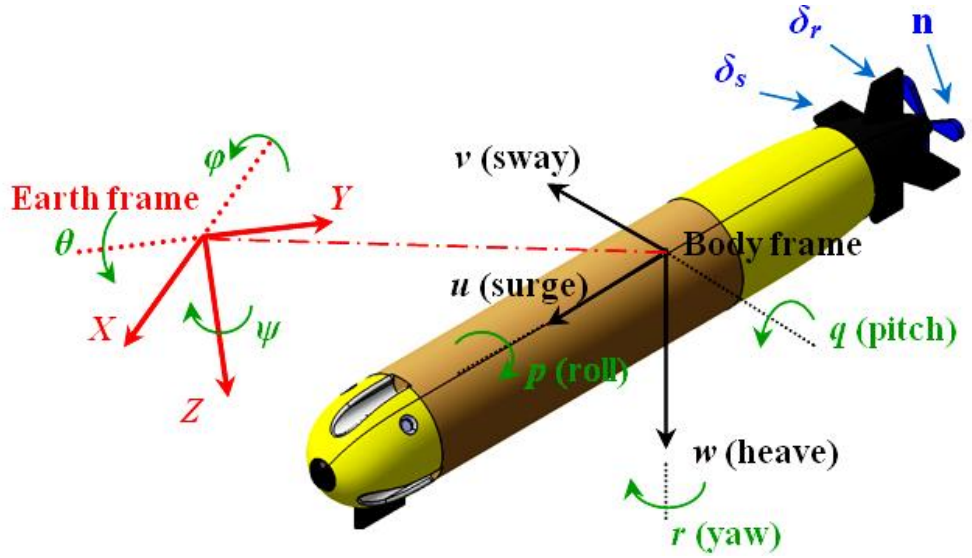

Figure 1: Coordinate Axes and Velocities of REMUS100 AUV

https://www.researchgate.net/figure/The-schematic-of-a-six-DOF-model-of-REMUS_fig1_286759304

The second coordinate system is referred to as earth reference or inertial coordinate system, where the AUVs displacement and orientation is taken into consideration. In this coordinate system, the distance between the origin and the position of the AUV is tracked in three axes. On the other hand, the vehicle's orientation or alignment is referenced angularly, which these angles called as Euler angles.

The positions and velocities of the vehicle can be described in vector form as follows:

$$\eta_1 = [x \quad y \quad z]^T \; and \; \eta_2 = [\phi \quad \theta \quad \psi]^T \tag{1}$$

The vectors η_1 and η_2 respectively correspond to displacements and angular velocities with respect to earth fixed or inertial coordinate system.

$$v_1 = [u \quad v \quad w]^T \; and \; v_2 = [p \quad q \quad r]^T \tag{2}$$

The vectors υ_1 and υ_2 respectively correspond to linear and angular velocities with respect to the body fixed coordinate system.

$$\tau_1 = [X \quad Y \quad Z]^T \; and \; \tau_2 = [K \quad M \quad N]^T \tag{3}$$

The vectors τ_1 and τ_2 correspond to forces and moments acting with respect to the body fixed coordinate system.

One may conclude that the derivation of positions and angular displacements may directly provide linear and angular velocities, but it requires a special transformation matrix that is called Jacobi matrix. For the transformation of linear motion, Jacobi matrix is given below:

$$\mathbf{J_1}(\eta_1) = \begin{bmatrix} cos\psi \; cos\theta & -sin\psi \; cos\phi + cos\psi \; sin\theta \; sin\phi & sin\psi \; sin\phi + cos\psi \; sin\theta \; cos\phi \\ sin\psi \; cos\theta & cos\psi \; cos\phi + sin\psi \; sin\theta \; sin\phi & -cos\psi \; sin\phi + \; sin\psi \; sin\theta \; cos\phi \\ -sin\theta & cos\theta \; sin\phi & cos\theta \; cos\phi \end{bmatrix} \tag{4}$$

Using this transformation matrix, it is possible to convert between first time derivatives of displacements in earth fixed coordinate system and linear velocities in body fixed coordinate system as below:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} = \mathbf{J_1}(\eta_1) \begin{bmatrix} u \\ v \\ w \end{bmatrix} \tag{5}$$

Similarly, the relation between angular velocities in body fixed coordinates and first derivation of angular displacements can be transformed with the following Jacobi matrix:

$$\mathbf{J_2}(\eta_2) = \begin{bmatrix} 1 & sin\phi \; tan\theta & cos\phi \; tan\theta \\ 0 & cos\phi & -sin\phi \\ 0 & sin\phi/cos\theta & cos\phi/cos\theta \end{bmatrix} \tag{6}$$

Applying the operation made before, transformation in angular motion can be done as follows:

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \mathbf{J_2}(\eta_2) \begin{bmatrix} p \\ q \\ r \end{bmatrix} \tag{7}$$

Notice that a singularity may occur as the angle θ is 90° which would make the result infinity. This condition physically corresponds to the vehicle turning its nose directly upward or downward, which is a condition generally not performed by the vehicle. Therefore, mathematically this singularity is not achieved in most of the operations.

3.2. Vehicle Dynamics

In this study, the REMUS100 Autonomous Underwater Vehicle (AUV) model is utilized to develop and test a nonlinear dynamic inversion (NDI) control strategy for autonomous underwater navigation. The REMUS100 is a torpedo-shaped AUV designed for applications such as oceanographic surveys, autonomous docking, and shallow-water mine reconnaissance. The dynamic model used in this study is based on a six-degree-of-freedom nonlinear model described in the work by Timothy Prestero [5].

The REMUS model incorporates the complete nonlinear dynamics, including hydrodynamic forces, buoyancy, added mass effects, and control inputs from the vehicle's propeller and control fins. The dynamic equations of motion are derived from rigid body mechanics, with external forces and moments computed using hydrodynamic coefficients. These coefficients were experimentally determined through tow tank tests and field experiments.

In this project, the NDI method is applied to control the REMUS100 AUV's motion in a simulated environment using a MATLAB/Simulink model. NDI is chosen for its ability to handle nonlinear dynamics by directly inverting the dynamic equations to achieve desired outputs. The control design involves defining a virtual control input that linearizes the nonlinear dynamics, allowing the control system to achieve precise depth and trajectory control.

The environment that AUV operates in has forces that make the system highly non-linear in terms of its dynamics. Some of these forces are buoyancy, lifting force, added mass, crossflow drag etc. To make an analysis of these forces, it is preliminary to define the center of gravity and buoyancy of the vehicle.

$$r_G = \begin{bmatrix} x_G \\ y_G \\ z_G \end{bmatrix} \; and \; r_B = \begin{bmatrix} x_B \\ y_B \\ z_B \end{bmatrix} \tag{8}$$

In the dynamics of an AUV, there are three forces and three moments acting which are forces acting on x-y-z axes in body fixed coordinates respectively and moments acting along the line of actions of these axes. Assuming the center of buoyancy at y_B is negligible compared to the other coordinates and similarly I_{xy}, I_{xz} and I_{yz} are negligible compared to the I_{xx}, I_{yy} and I_{zz} it has been shown that the equations of motions of the AUV is:

For x-direction:

$$\begin{aligned}(m - X_{\dot{u}})\dot{u} + mz_g\dot{q} - my_g\dot{r} \\ = X_{HS} + X_{u|u|}u|u| + \left(X_{wq} - m\right)wq + \left(X_{qq} + mx_q\right)q^2 \\ + (X_{vr} + m)vr + \left(X_{rr} + mx_g\right)r^2 - my_gpq - mz_gpr + X_{prop}\end{aligned} \tag{9}$$

For y-direction:

$$\begin{aligned}(m - Y_{\dot{v}})\dot{v} - mz_g\dot{p} - \left(mx_g - Y_{\dot{r}}\right)\dot{r} \\ = Y_{HS} + Y_{v|v|}v|v| + Y_{r|r|}r|r| + my_gr^2 + (Y_{ur} - m)ur \\ + \left(Y_{wp} + m\right)wp + \left(Y_{pq} - mx_g\right)pq + Y_{uv}uv + my_gp^2 + mz_gqr \\ + Y_{uu\delta_r}u^2\delta_r\end{aligned} \tag{10}$$

For z-direction:

$$\begin{aligned}(m - Z_{\dot{\omega}})\dot{w} + my_g\dot{p} - \left(mx_g - Z_{\dot{q}}\right)\dot{q} \\ = Z_{HS} + Z_{w|w|}w|\omega| + Z_{q|q|}q|q| + \left(Z_{uq} + m\right)uq + \left(Z_{vp} - m\right)vp \\ + \left(Z_{rp} - mx_g\right)rp + Z_{uw}uw + mz_g(p^2 + q^2) - my_grq + Z_{uu\delta_s}u^2\delta_s\end{aligned} \tag{11}$$

Rotational motion about x-axis:

$$\begin{aligned}-mz_g\dot{v} + my_g\dot{w} + \left(I_{xx} - K_{\dot{p}}\right)\dot{p} \\ = K_{HS} + K_{p|p|}p|p| - \left(I_{zz} - I_{yy}\right)qr + m(uq - vp) - mz_g(wp - ur) \\ + K_{prop}\end{aligned} \tag{12}$$

Rotational motion about y-axis:

$$
\begin{aligned}
&mz_g\dot{u} - \left(mx_g + M_{\dot{\omega}}\right)\dot{w} + \left(I_{yy} - M_{\dot{q}}\right)\dot{q} \\
&\qquad = M_{HS} + M_{w|w|}w|w| + M_{q|q|}q|q| + \left(M_{uq} + mx_g\right)uq + \left(M_{vp} + mx_g\right)vp \\
&\qquad + \left[M_{rp} - (I_{xx} - I_{zz})\right]rp + mz_g(vr - wq) + M_{uw}uw + M_{uu\delta_s}u^2\delta_s
\end{aligned} \tag{13}
$$

Finally, rotational motion about z-axis:

$$
\begin{aligned}
&-my_g\dot{u} + \left(mx_g - N_{\dot{v}}\right)\dot{v} + (I_{zz} - N_{\dot{r}})\dot{r} \\
&\qquad = N_{HS} + N_{v|v|}v|v| + N_{r|r|}r|r| + \left(N_{ur} - mx_g\right)ur + \left(N\omega p + mx_g\right)wp \\
&\qquad + \left[N_{pq} - \left(I_{yy} - I_{xx}\right)\right]pq - my_g(vr - wq) + N_{uv}uv + N_{uu\delta_r}u^2\delta_r
\end{aligned} \tag{14}
$$

Furthermore, this data can be shown in matrix format:

$$
\begin{bmatrix}
m - X_{\dot{u}} & 0 & 0 & 0 & mz_g & -my_g \\
0 & m - Y_{\dot{v}} & 0 & -mz_g & 0 & mx_g - Y_{\dot{r}} \\
0 & 0 & m - Z_{\dot{\omega}} & my_g & -mx_g - Z_{\dot{q}} & 0 \\
0 & -mz_g & my_g & I_{xx} - K_{\dot{p}} & 0 & 0 \\
mz_g & 0 & -mx_g - M_{\dot{\omega}} & 0 & I_{yy} - M_{\dot{q}} & 0 \\
-my_g & mx_g - N_{\dot{v}} & 0 & 0 & 0 & I_{xx} - N_{\dot{r}}
\end{bmatrix}
\begin{bmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \\ \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix}
=
\begin{bmatrix} \sum X \\ \sum Y \\ \sum Z \\ \sum K \\ \sum M \\ \sum N \end{bmatrix}
\tag{15}
$$

Taking the inverse of the first matrix provides an easier approach for the calculation of linear and angular accelerations:

$$
\begin{bmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \\ \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix}
=
\begin{bmatrix}
m - X_{\dot{u}} & 0 & 0 & 0 & mz_g & -my_g \\
0 & m - Y_{\dot{v}} & 0 & -mz_g & 0 & mx_g - Y_{\dot{r}} \\
0 & 0 & m - Z_{\dot{\omega}} & my_g & -mx_g - Z_{\dot{q}} & 0 \\
0 & -mz_g & my_g & I_{xx} - K_{\dot{p}} & 0 & 0 \\
mz_g & 0 & -mx_g - M_{\dot{\omega}} & 0 & I_{yy} - M_{\dot{q}} & 0 \\
-my_g & mx_g - N_{\dot{v}} & 0 & 0 & 0 & I_{xx} - N_{\dot{r}}
\end{bmatrix}^{-1}
\begin{bmatrix} \sum X \\ \sum Y \\ \sum Z \\ \sum K \\ \sum M \\ \sum N \end{bmatrix}
\tag{16}
$$

This notation will provide an easier way for the calculation of acceleration and make it more suitable for vehicle simulation.

4. Theoretical Background

This chapter establishes the theoretical foundations of the control strategies employed in this study, specifically focusing on the mathematical principles of Nonlinear Dynamic Inversion (NDI) and the stability analysis of non-minimum phase systems.

4.1.Non-linear Dynamic Inversion (NDI) Concept

Nonlinear Dynamic Inversion (NDI) is a control technique used to directly linearize the nonlinear dynamics of a system, enabling the design of feedback controllers for complex dynamic systems. By analytically inverting the system dynamics, This method is particularly useful in applications where highly nonlinear behavior, such as hydrodynamic effects in underwater vehicles, must be managed effectively. NDI offers a systematic approach to feedback control where the nonlinearities are systematically considered, enhancing precision and stability in dynamic systems.

In a multi-input multi-output (MIMO) system the state space equation of the system can be represented as:

$$\dot{x} = f(x) + \mathbf{g}(\mathbf{x})u \tag{17}$$

$$y = h(x) \tag{18}$$

It should be noted that G(x) is a matrix that may depend nonlinearly on x. It is also assumed that the state derivative $\dot{x}$ is linearly dependent on the control input u. Furthermore, the input vector u is described as having a dimension of m, while the output vector y is described as having a dimension of p. The state transformation in MIMO systems is performed to find a new set of state variables where the control inputs influence the outputs in a decoupled, linearized manner. This allows standard linear control techniques to be applied to nonlinear systems.

4.1.1 What Problems Does NDI Address?

Nonlinear Dynamic Inversion (NDI) is designed to overcome several key limitations that arise when using traditional control techniques on nonlinear or multi-input multi-output (MIMO) systems. One of the primary challenges it addresses is the presence of nonlinear dynamics, which are common in real-world systems such as underwater vehicles, aircraft, and robotic manipulators. Classical controllers like PID assume a linear relationship between input and output and often fail to perform effectively when system behavior varies significantly with operating conditions. NDI directly accounts for nonlinearities by using the full nonlinear model of the system to compute control actions, leading to better tracking accuracy and stability.

NDI also enhances tracking performance by proactively generating control inputs based on model knowledge, rather than reacting to accumulated errors like PID does. This predictive nature reduces steady-state error and transient overshoot. Furthermore, because NDI computes the exact control effort needed for a desired behavior, it helps avoid actuator saturation and ensures smoother transitions. Lastly, since NDI is model-based, it supports structured tuning and can be easily adapted to changes in system configuration or dynamics, making it highly versatile and robust for advanced control applications.

4.1.2 How Does NDI Work?

The transformation begins by differentiating the output equations until the control input explicitly appears:

$$\dot{y} = \frac{\partial h}{\partial x} f(x) + \frac{\partial h}{\partial x} \mathbf{g}(\mathbf{x}) u \tag{19}$$

The control law for the transformed system is designed using feedback linearization, where the desired output can be utilized as a virtual control input to simplify the control design. This approach transforms the nonlinear system into a linearized form where the control inputs can be directly computed:

$$u = \mathbf{G^{-1}}(x)(y_{des} - F(x)) \tag{20}$$

The Nonlinear Dynamic Inversion (NDI) method can be effectively applied to control the REMUS Autonomous Underwater Vehicle (AUV) by managing both its fast and slow dynamics. Fast dynamics, such as angular rates, can be controlled using an inner loop control strategy, while slower dynamics, such as position and depth, can be regulated through an outer loop design.

4.1.3 Why NDI is Suitable for the REMUS AUV

The REMUS AUV, like many underwater vehicles, exhibits strongly nonlinear and coupled dynamics due to its interaction with the surrounding fluid environment. These nonlinearities arise from quadratic drag, added mass, cross-coupling effects between velocity components, and the hydrodynamic influence of control surfaces such as rudders and stern planes. This complexity makes it difficult to apply traditional control strategies like PID, which rely on decoupled and linear behavior and are often inadequate for systems with such highly coupled and velocity-dependent dynamics.

What is needed for effective control of REMUS is a strategy that can:

1. handle nonlinear and time-varying dynamics,
2. decouple the influence of inputs on outputs,
3. allow for multi-variable tracking (e.g., surge, sway, heave, pitch rate, and yaw rate), and
4. perform robustly under trajectory-following tasks in 3D space such as spirals or dives.

Nonlinear Dynamic Inversion (NDI) is particularly well-suited to these needs. Because NDI uses the system's full nonlinear model, it directly incorporates all coupling effects and drag characteristics into the control computation. The inversion of the input-output map allows NDI to allocate actuator effort proportionally and accurately across multiple objectives. NDI enables us to design the closed-loop output behavior explicitly that includes tracking error dynamics. This makes it possible to guarantee stability and tracking performance across a broad range of operating conditions, without the need for constant gain retuning as required in PID controllers. Thus, NDI aligns perfectly with the REMUS control challenge: it is designed for systems where classical assumptions do not hold, and precise, model-based inversion is required to achieve performance and robustness in a nonlinear, and coupled environment.

4.2. Input-Output Linearization and Zero Dynamics

Feedback linearization differs from conventional linearization (Jacobian linearization) in that it exacts a global transformation of the system dynamics rather than approximating them around a specific operating point. The central idea is to algebraically transform the nonlinear system dynamics into a linear one so that linear control techniques can be applied. However, the applicability of this method is strictly constrained by the internal stability of the system, which is analyzed through the concept of Zero Dynamics.

4.2.1 Relative Degree and Internal Dynamics

Consider a Single-Input Single-Output (SISO) nonlinear system described by the affine form:

$$\dot{x} = f(x) + \mathbf{g}(\mathbf{x})u \tag{21}$$

$$y = h(x) \tag{22}$$

where $x \in R^n$ is the state vector, $u \in R$ is the control input, and $y \in R$ is the output. The relative degree r of the system is defined as the number of times the output y must be differentiated with respect to time until the control input u explicitly appears in the equation. Mathematically, this is expressed using Lie derivatives:

$$L_g L_f^k h(x) = 0 \quad \text{for } k < r-1$$
$$L_g L_f^{r-1} h(x) \neq 0 \tag{23}$$

If the relative degree r is equal to the system order n (r=n), the system is fully feedback linearizable, and there are no internal dynamics. However, for many physical systems (including under-actuated underwater vehicles), the relative degree is strictly less than the system order (r < n). In such cases, the system possesses n-r states that are "hidden" from the input-output relationship. These hidden states constitute the Internal Dynamics.

To analyze these dynamics, a coordinate transformation (diffeomorphism) is performed to convert the system into the Byrnes-Isidori Normal Form. The states are partitioned into external states $\xi = \left[y, \dot{y}, \dots, y^{(r-1)}\right]^T$ and internal states η.

4.2.2 Zero Dynamics and Non-Minimum Phase Behavior

The Zero Dynamics are defined as the internal dynamics of the system when the output and all its derivatives are constrained to be identically zero for all time ($y(t) \equiv 0 \Rightarrow \xi(t) = 0$). By calculating the specific control input u_{zero} required to maintain this constraint and substituting it back into the internal state equations, one obtains the autonomous differential equation:

$$\dot{\eta} = f_{zero}(\eta) \tag{24}$$

The stability of this equation dictates the feasibility of the feedback linearization control strategy:

1. **Minimum Phase System:** If the zero dynamics are asymptotically stable (i.e., the eigenvalues of the linearized zero dynamics lie in the left-half plane), the system is termed Minimum Phase. In this case, controlling the output y ensures that the internal states η remain bounded, making Input-Output Linearization a valid and safe control strategy.

2. **Non-Minimum Phase (NMP) System:** If the zero dynamics are unstable (i.e., eigenvalues in the right-half plane), the system is Non-Minimum Phase. Physically, this manifests as an "inverse response" where the system initially moves in the opposite direction of the desired setpoint (undershoot).

For NMP systems, standard Input-Output Linearization cannot be applied directly to the output y. Attempting to force the tracking error to zero would require unbounded control energy and would cause the internal states to diverge, leading to catastrophic system failure. Therefore, identifying NMP behavior is a critical prerequisite for designing the control architecture.

5. Controller Design and Implementation

5.1. Design Justification: Zero Dynamics Analysis

5.1.1. Analysis of Depth Control

To evaluate the internal stability of the system under the proposed Nonlinear Dynamic Inversion (NDI) controller, a zero dynamics analysis was conducted. This analysis is crucial for underactuated systems like the AUV, where the number of control inputs is less than the system states. Specifically, we investigate whether maintaining the output (depth, z) at a constant value results in stable or unstable internal behavior (pitch dynamics).

1. Kinematic Constraints on the Zero Dynamics

The output of the system is defined as the depth position:

$$\mathrm{y} = \mathrm{h(x)} = \mathrm{z} \tag{25}$$

To derive the zero dynamics, the output is constrained to be identically zero for all time ($y(t) \equiv 0$) . This implies that all time derivatives of the output must also vanish:

$$z(t) = 0, \qquad \dot{z}(t) = 0, \qquad \ddot{z}(t) = 0 \tag{26}$$

Using the kinematic transformation from the body frame to the inertial frame (assuming zero roll, $\phi \approx 0$, and zero sway, $v \approx 0$), the depth rate is given by:

$$\dot{z} = -u\, sin\theta + w\, cos\theta \tag{27}$$

To apply the constraint $\dot{z} = 0$, we derive the geometric relationship between the heave velocity (w) and the pitch angle (θ) on the zero dynamics.

$$-u\ sin\theta + w\ cos\theta = 0 \Longrightarrow w = u\ tan\theta \tag{28}$$

This relationship indicates that to maintain a constant depth, any pitch deflection (θ) must be accompanied by a corresponding heave velocity (w) to counteract the vertical component of the surge velocity.

2. Dynamics of the Internal States

Differentiating the constraint equation once more yields the acceleration constraint:

$$\ddot{z} = \frac{d}{dt}(-u\ sin\theta + w\ cos\theta) \tag{29}$$

Expanding this term using the chain rule ($\dot{\theta} = q$):

$$\ddot{z} = -\dot{u}sin\theta - uq\ cos\theta + \dot{w}cos\theta - wq\ sin\theta = 0 \tag{30}$$

3. Internal Dynamics Equation

Substituting the dynamic equations of motion for surge ($\dot{u}$) and heave ($\dot{w}$) into this equation allows us to solve for the required control input (δ_s^*) that maintains the constraint. The simplified longitudinal dynamics (assuming a diagonal mass matrix) are:

$$\dot{u} = \frac{X_{total}}{m - X_{\dot{u}}}$$

$$\dot{w} = \frac{Z_{total} + Z_{\delta_s}\delta_s}{m - Z_{\dot{w}}} \tag{31}$$

$$\dot{q} = \frac{M_{total} + M_{\delta_s}\delta_s}{I_{yy} - M_{\dot{q}}}$$

Solving the equation $\ddot{z} = 0$ for the control input yields the feedback linearization law restricted to the zero dynamics:

$$\delta_s^* = -\frac{L_f^2 h(x)}{L_g L_f h(x)} \tag{32}$$

Where the denominator represents the decoupling term (control effectiveness in the vertical direction).

Substituting the calculated δ_s^* and the kinematic constraint $w = u\,tan\theta$ into the pitch moment equation ($\dot{q}$), we obtain the autonomous differential equation governing the internal dynamics:

$$\dot{q}_{zero} = f(\theta, q, u_0) \tag{33}$$

Where u_0 is the constant surge velocity. This equation describes the rotational motion of the vehicle when it is forced to move in a straight horizontal line.

4. Linearization and Stability Assessment

To assess local stability, the zero dynamics equation was linearized around the equilibrium point $x_{eq} = [\theta, q]^T = [0,0]^T$. The Jacobian matrix A_{zero} is derived as:

$$A_{zero} = \left[\begin{matrix} \frac{\partial \dot{\theta}}{\partial \theta} & \frac{\partial \dot{\theta}}{\partial q} \\ \frac{\partial \dot{q}_{zer0}}{\partial \theta} & \frac{\partial \dot{q}_{zer0}}{\partial q} \end{matrix}\right]\Bigg|_{\theta=0,q=0} = \begin{bmatrix} 0 & 1 \\ 5.58 & 1.98 \end{bmatrix} \tag{34}$$

Using the hydrodynamic parameters of the REMUS 100 AUV, the eigenvalues of this matrix were computed. The analysis yielded one positive real eigenvalue ($\lambda > 0$) and one negative real eigenvalue.

The existence of a positive eigenvalue in the right-half plane (RHP) indicates that the zero dynamics are unstable. Consequently, the system is Non-Minimum Phase (NMP) with respect to the output z.

5.1.2 Analysis of Pitch Control

To contrast with the depth control scenario and verify the inherent stability of the vehicle's attitude dynamics, a zero dynamics analysis was performed for the pitch angle output. In this case, the control objective is to maintain the pitch angle at zero, irrespective of the vehicle's depth or vertical velocity.

1. Constraints on the Zero Dynamics Manifold

The output is defined as the pitch angle:

$$y = h(x) = \theta \tag{35}$$

To derive the zero dynamics, we impose the constraint that the output is identically zero for all time ($y(t) \equiv 0$). Consequently, its time derivatives must also vanish:

$$\theta(t) = 0, \quad \dot{\theta}(t) = q = 0, \quad \ddot{\theta}(t) = \dot{q} = 0 \tag{36}$$

Unlike the depth analysis, where a kinematic constraint linked the heave velocity (w) to the pitch angle, here the heave velocity w is a free internal state. The constraints strictly freeze the rotational states (θ and q).

2. Computation of the Required Control Input

The necessary control input δ_s^* to maintain the zero-pitch constraint is derived from the pitch dynamics equation. Using the simplified diagonal mass matrix model, the pitch acceleration is given by:

$$\dot{q} = \frac{M_{total} + M_{\delta_s}\delta_s}{I_{yy} - M_{\dot{q}}} \tag{37}$$

Expanding the total moment term M_{total} and imposing $\dot{q} = 0$ and $q = 0$:

$$M_{ww}w + M_{qq}(0) - (m_z - m_x)u_0 w + M_{\delta_s}\delta_s^* = 0 \tag{38}$$

Solving for δ_s^*, we obtain the feedback linearization control law required to keep the vehicle level:

$$\delta_s^* = -\frac{1}{M_{\delta_s}}[M_{ww} - (m_z - m_x)u_0]w \tag{39}$$

3. Internal Dynamics Equation

To find the behavior of the remaining internal state (heave velocity, w), we substitute the derived control input δ_s^* into the heave dynamics equation:

$$m_z\dot{w} = Z_{ww}w + Z_{qq}(0) + mu_0(0) + Z_{\delta_s}\delta_s^* \tag{40}$$

Note that the centripetal term $m\,u\,q$ and pitch damping $Z_{qq}q$ vanish because $q = 0$. Substituting δ_s^*:

$$m_z\dot{w} = Z_{ww}w - \frac{Z_{\delta_s}}{M_{\delta_s}}[M_{ww} - (m_z - m_x)u_0]w \tag{41}$$

This results in a first-order differential equation of the form:

$$\dot{w}_{zero} = \lambda_{zero} \cdot w \tag{42}$$

4. Stability Assessment

The stability of the zero dynamics is determined by the sign of the scalar coefficient λ_{zero} (which corresponds to the eigenvalue of the internal dynamics). Rearranging the terms:

$$\lambda_{zero} = \frac{1}{m_z}\left(Z_{ww} - \frac{Z_{\delta_s}}{M_{\delta_s}}[M_{ww} - (m_z - m_x)u_0]\right) \tag{43}$$

Numerical evaluation yields a negative eigenvalue $\lambda_{zero} < 0$

The internal dynamics for the pitch output are stable. This implies that if the pitch angle is constrained to zero, any initial vertical velocity w will decay to zero (or a small equilibrium) rather than growing unbounded. Therefore, the system is Minimum Phase with respect to the pitch output θ.

5. Conclusion of the Analysis

The contrast between the two analyses is significant for the control design:

- **Depth Output (z):** Non-Minimum Phase (Unstable Zero Dynamics). Direct inversion leads to instability.
- **Pitch Output ($\boldsymbol{\theta}$):** Minimum Phase (Stable Zero Dynamics). Direct inversion is safe and effective.

This justifies the use of a cascaded control architecture or output redefinition, where the internal loop controls the, pitch variable θ, and the outer loop manages the depth z by generating pitch commands.

5.2. Control Architecture Overview

The implementation of the control strategy for the REMUS 100 is fundamentally shaped by the specific stability characteristics and physical constraints derived in the previous stability analysis. A primary challenge in controlling this vehicle is its under-actuated nature; while it operates in six degrees of freedom, it possesses only three independent control inputs (surge thrust, rudder deflection, and stern plane deflection). Crucially, the vehicle lacks direct actuation in the sway (v) and heave (w) axes, rendering it impossible to translate laterally or vertically without forward motion.

Furthermore, as demonstrated in the Zero Dynamics Analysis, the vehicle exhibits Non-Minimum Phase (NMP) behavior when the vertical position (depth, z) is selected as the direct output for dynamic inversion. This implies that a direct Input-Output Linearization of the depth dynamics would result in unstable internal dynamics, making standard inversion techniques infeasible for direct position control.

To reduce the NMP instability and address the under-actuation constraints, this study employs a Three-Layer Cascaded Control Architecture. The design is grounded in the Time-Scale Separation (TSS) principle, a methodology that has become an industry standard in high-performance aerospace flight control systems. The fundamental premise of TSS is that the system dynamics can be decomposed into distinct loops based on their bandwidth. By exploiting the fact that rotational dynamics (body rates) evolve

much faster than attitude kinematics (angles), which in turn evolve faster than the translational trajectory (position). The proposed architecture is structured into three hierarchical layers:

1. **Outer Loop (Guidance and Path Correction):**

Since the position dynamics cannot be inverted directly due to NMP characteristics, the outermost loop governs the slow translational dynamics using a Line-of-Sight (LOS) guidance strategy. The LOS algorithm acts as a stable interface, converting position errors (cross-track and depth errors) into desired orientation commands. Specifically, it generates the desired pitch angle (θ_{des}) and heading angle (ψ_{des}) required to align the vehicle's velocity vector with the desired path.

2. **Middle Loop (Attitude Control):**

The middle layer is responsible for the vehicle's orientation. This loop receives the desired angles from the LOS block and inverts the Euler kinematic equations to compute the necessary angular velocities. Its output is the desired body rates: Pitch rate (q_{des}) and Yaw rate (r_{des}).

3. **Inner Loop (Rate and Speed Control):**

The innermost loop handles the fastest dynamics of the system. This is the "Dynamic Inversion" layer where the forces and moments are generated. It takes the desired rates (q_{des}, r_{des}) and the desired surge speed (u_{des}), performs the inversion of the vehicle's Newton-Euler equations of motion, and calculates the actual physical control inputs required from the actuators (δ_s, δ_r, RPM).

This hierarchical structure ensures that the unstable zero dynamics associated with depth are bypassed, as the depth is controlled indirectly via the stable pitch attitude, thereby ensuring robust path following while maintaining internal stability.

5.3.Mathematical Modeling for Control

The general equations of motion for the REMUS 100, as derived in Prestero's work, can be expressed in he control-affine state-space form:

$$\dot{x} = f(x) + g(x)u \tag{44}$$

Where x is the state vector and u is the control input vector defined as:

$$x = [u, v, w, p, q, r, x, y, z, \phi, \theta, \psi]^T \tag{45}$$

$$u = [n, \delta_s, \delta_r]^T \tag{46}$$

Here, n represents the propeller RPM, δ_s is the stern plane deflection, and δ_r is the rudder deflection.

The system dynamics are split into dynamic and kinematic parts to facilitate the inversion process:

By isolating the acceleration vector $\dot{x}$, we formulated the dynamic part of the state-space representation:

$$\dot{x} = \mathbf{M}^{-1}(\tilde{f}(x) + f_{HS}(x)) + \mathbf{g}(\mathbf{x})u \tag{47}$$

Based on the REMUS model analysis, $\tilde{g}(x)$ is a 6x3 matrix:

$$\tilde{\mathbf{g}}(\mathbf{x}) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & Y_{uu\delta_r}u^2 \\ 0 & Z_{uu\delta_s}u^2 & 0 \\ 0 & 0 & 0 \\ 0 & M_{uu\delta_s}u^2 & 0 \\ 0 & 0 & N_{uu\delta_r}u^2 \end{bmatrix} \tag{48}$$

Where M is the 6x6 mass-inertia matrix defined previously:

$$\mathbf{M} = \begin{bmatrix} m - X_{\dot{u}} & 0 & 0 & 0 & mz_g & -my_g \\ 0 & m - Y_{\dot{v}} & 0 & -mz_g & 0 & mx_g - Y_{\dot{r}} \\ 0 & 0 & m - Z_{\dot{\omega}} & my_g & -mx_g - Z_{\dot{q}} & 0 \\ 0 & -mz_g & my_g & I_{xx} - K_{\dot{p}} & 0 & 0 \\ mz_g & 0 & -mx_g - M_{\dot{\omega}} & 0 & I_{yy} - M_{\dot{q}} & 0 \\ -my_g & mx_g - N_{\dot{v}} & 0 & 0 & 0 & I_{xx} - N_{\dot{r}} \end{bmatrix} \tag{49}$$

(Note: Coefficients x_g and y_g were substituted as 0 based on model assumption)

The full $\mathbf{g}(\mathbf{x})$ matrix for the 12-dimensional state vector includes the dynamic part and zeros for

the kinematic states:

$$\mathbf{g}(\mathbf{x}) = \begin{bmatrix} \mathbf{M}^{-1}\tilde{\mathbf{g}}(\mathrm{x}) \\ 0_{6x3} \end{bmatrix}_{12x3} \tag{50}$$

Where 0_{6x3} is a 6x3 zero matrix, indicating that the control inputs do not directly affect the kinematic state derivatives.

For control-independent vector f(x) covers all dynamics not directly multiplied by the control input u It consists of two parts:

1. **Dynamic Part $\mathbf{f_{dyn}(x)}$:** This comes from the control-independent forces and moments affecting acceleration.

$$\mathrm{f}_{\mathrm{dyn}}(\mathrm{x}) = \mathbf{M}^{-1}\big(\tilde{\mathrm{f}}(\mathrm{x}) + \mathrm{f}_{\mathrm{HS}}(\mathrm{x})\big) \tag{51}$$

Where f_{HS} is the hydrostatic forces vector that acting on the vehicle, expressed in the body-fixed frame, results from the difference between weight (W) and buoyancy (B) projected onto the body axes via the vehicle's orientation (Euler angles: ϕ, θ, ψ). It is defined as:

$$\mathrm{f}_{\mathrm{HS}}(\mathrm{x}) = \begin{bmatrix} -(W-B)\cdot\sin\theta \\ (W-B)\cdot\cos\theta\cdot\sin\phi \\ (W-B)\cdot\cos\theta\cdot\cos\phi \\ -z_g\cdot W\cdot\cos\theta\cdot\sin\phi \\ -z_g\cdot W\cdot\sin\theta \\ 0 \end{bmatrix} \tag{52}$$

Here, the hydrostatic moment equations simplify because the REMUS model strategically places the body-frame coordinate origin exactly at the Center of Buoyancy (CB), making $x_b = 0, y_b = 0, z_b = 0$ by definition . Additionally, due to assumed vehicle symmetry, the Center of Gravity (CG) lies directly below the origin on the z-axis, making $x_g = 0$ and $y_g = 0$. Substituting these zero values into the general moment formulas eliminates terms involving $x_g = 0, y_g = 0, x_b = 0, y_b = 0$ and $z_b = 0$.

1. **Kinematic Part $\mathbf{f_{kin}(x)}$:** This describes how body-frame velocities are transformed into Earth-frame rates using the kinematic transformation matrices $\mathbf{J_1}$ and $\mathbf{J_2}$.

$$\mathrm{f_{kin}(x)} = \begin{bmatrix} \mathbf{J_1}(\eta_1)\nu_1 \\ \mathbf{J_2}(\eta_2)\nu_2 \end{bmatrix}_{6x1} \tag{53}$$

The complete vector is:

$$\mathrm{f(x)} = \begin{bmatrix} \mathrm{f_{dyn}(x)} \\ \mathrm{f_{kin}(x)} \end{bmatrix}_{12x1} \tag{54}$$

Combining the derived $\mathrm{f_{dyn}(x)}$ and $\mathrm{f_{kin}(x)}$, the full state-space representation for the REMUS 100 AUV in control-affine form is explicitly written as:

$$\dot{\mathrm{x}} = \begin{bmatrix} \mathbf{M}^{-1}\big(\tilde{\mathrm{f}}(\mathrm{x}) + \mathrm{f_{HS}(x)}\big) \\ \mathbf{J_1}(\eta_1)\nu_1 \\ \mathbf{J_2}(\eta_2)\nu_2 \end{bmatrix} + \begin{bmatrix} \mathbf{M}^{-1}\tilde{\mathrm{g}}(\mathrm{x}) \\ 0_{6x3} \end{bmatrix} \begin{bmatrix} n \\ \delta_s \\ \delta_r \end{bmatrix} \tag{55}$$

5.4. Outer Loop – Line of Sight (LOS) Algorithm

The outermost layer of the control architecture is responsible for the guidance of the REMUS 100 AUV. Since the vehicle is under-actuated and exhibits Non-Minimum Phase (NMP) characteristics in its depth dynamics, a direct inversion of position states is not feasible. To address this, a 3D Line-of-Sight (LOS) guidance strategy is implemented.

The primary objective of this loop is to convert the position errors into desired orientation commands (θ_{des}, ψ_{des}) and a desired surge speed command (u_{des}), which serve as reference inputs for the inner loops.

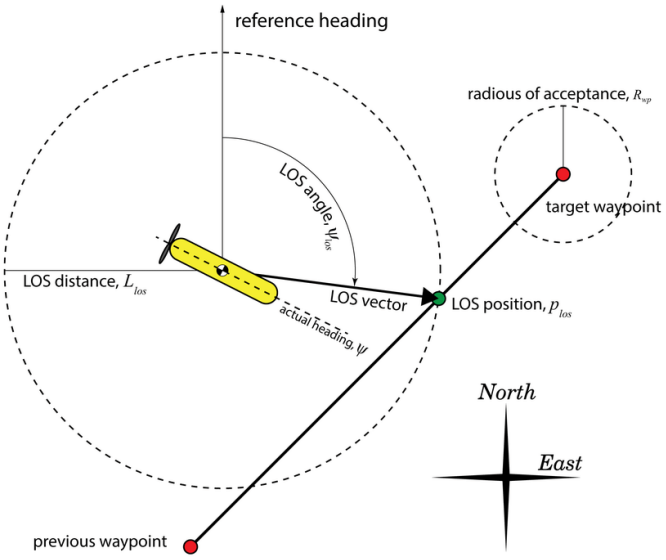

Figure 2: Line of Sight Guidance Law

https://www.researchgate.net/figure/Line-of-sight-guidance-law_fig11_320564903

4.1.1. Path Error Dynamics

To accurately compute the tracking errors, the system utilizes a path-tangential reference frame attached to a moving virtual target. Let $P = [x, y, z]^T$ denote the vehicle's position and $P_v = [x_v, y_v, z_v]^T$ denote the virtual target's position in the inertial frame. The orientation of the virtual target's velocity vector is defined by the path pitch angle θ_p and path heading angle ψ_p.

The error vector $e = [x_e, y_e, z_e]^T$ is calculated by transforming the inertial position difference into the path frame using the rotation matrix $\mathbf{R_p^T}$:

$$\begin{bmatrix} x_e \\ y_e \\ z_e \end{bmatrix} = \mathbf{R_p^T}(P - P_v) = \begin{bmatrix} cos\psi_p cos\theta_p & sin\psi_p cos\theta_p & -sin\theta_p \\ -sin\psi_p & cos\psi_p & 0 \\ cos\psi_p sin\theta_p & sin\psi_p sin\theta_p & cos\theta_p \end{bmatrix} \begin{bmatrix} x - x_v \\ y - y_v \\ z - z_v \end{bmatrix} \tag{56}$$

Here:

- x_e: Along-track error (longitudinal distance to the virtual target).
- y_e: Cross-track error (lateral deviation from the path).
- z_e: Vertical track error (depth deviation from the path).

4.1.2. Computation of Steering Angles

To eliminate the cross-track (y_e) and vertical (z_e) errors, the guidance law computes local steering angles, denoted as ψ_r (relative heading) and θ_r (relative pitch). A hyperbolic tangent function (tanh) is employed to map these errors to a bounded angular range. This saturation technique ensures smooth convergence to the path without inducing excessive overshoot or actuator saturation when errors are large.

The steering laws are defined as:

$$\psi_r = \tanh\left(-\frac{k_y \cdot y_e}{\Delta_y}\right) \tag{57}$$

$$\theta_r = \tanh\left(\frac{k_z \cdot z_e}{\Delta_z}\right)$$

Where k_y, k_z are the guidance gains and Δ_y , Δ_z represent the look-ahead distances for lateral and vertical planes, respectively.

4.1.3. Generation of Global Orientation Commands (θ_{des}, ψ_{des})

The local steering angles (θ_r, ψ_r) define the required orientation relative to the path frame. To be used by the vehicle's attitude controller (Middle Loop), these must be transformed back into the global inertial frame. This geometric transformation combines the path's current orientation (θ_p, ψ_p) with the steering corrections.

The Desired Pitch Angle (θ_{des}) is computed by projecting the total desired velocity vector onto the vertical axis:

$$\theta_{des} = \arcsin\left(\sin\theta_p \cos\theta_r \cos\psi_r + \cos\theta_p \sin\theta_r\right) \tag{58}$$

Similarly, the Desired Heading Angle (ψ_{des}) is derived from the horizontal components of the desired vector. Defining the intermediate projection components χ_x and χ_y:

$$\begin{aligned}\chi_x = &-\sin\psi_p \sin\psi_r \cos\theta_r - \sin\theta_p \sin\theta_r \cos\psi_p \\ &+ \cos\psi_p \cos\psi_r \cos\theta_p \cos\theta_r\end{aligned} \tag{59}$$

$$\chi_y = \cos\psi_p \sin\psi_r \cos\theta_r - \sin\theta_p \sin\theta_r \sin\psi_p + \sin\psi_p \cos\psi_r \cos\theta_p \cos\theta_r \quad (60)$$

The global desired heading is then obtained using the four-quadrant arctangent function:

$$\psi_{des} = \tan^{-1}(\chi_y, \chi_x) \quad (61)$$

These calculated θ_{des} and ψ_{des} values are fed into the Middle Loop as the tracking references.

4.1.4. Generation of Desired Surge Speed (u_{des})

The speed controller aims to maintain a constant distance from the virtual target, effectively synchronizing the vehicle's motion with the desired mission timing. The base speed is determined by the virtual target's velocity magnitude ($|V_p|$), adjusted by a proportional feedback term based on the along-track error (x_e).

Additionally, the command is scaled by the cosine of the steering angles. This ensures that if the vehicle needs to turn sharply, the surge speed is increased to maintain the velocity projection along the path. The control law for Desired Surge Speed is:

$$u_{des} = \frac{|V_p| - k_x \cdot x_e}{\cos\psi_r \cdot \cos\theta_r} \quad (62)$$

5.5. Middle Loop Design – Slow Dynamics

This layer is responsible for tracking the desired orientation (θ_{des}, ψ_{des}) generated by the LOS guidance. Since the NDI framework operates on state derivatives, a linear PID controller is first employed to bridge the gap between the reference angles and the required angular rates. The controller acts on the tracking error e(t) to generate a control signal representing the required Euler rates:

$$\begin{bmatrix} \dot{\theta}_{req} \\ \dot{\psi}_{req} \end{bmatrix} = K_P e(t) + K_I \int e(t)dt + K_D \dot{e}(t) \quad (63)$$

These rates serve as the input for the Kinematic NDI block. By inverting the Euler kinematic matrix and assuming stabilized roll ($\phi \approx 0$), the controller computes the necessary body-frame angular velocities (q_{des},r_{des}) for the inner loop:

The kinematic relationship is governed by the transformation matrix $\mathbf{J_2}\ (\eta_2)$:

$$\dot{\eta_2} = \mathbf{J_2}\ (\eta_2)\nu_2 \Longrightarrow \begin{bmatrix}\dot{\theta}\\ \dot{\psi}\end{bmatrix} = \begin{bmatrix}cos\phi & -sin\phi\\ \frac{sin\phi}{cos\theta} & \frac{cos\phi}{cos\theta}\end{bmatrix}\begin{bmatrix}q\\ r\end{bmatrix} \tag{64}$$

To find the desired body rates ($\nu_{2,des}$), the transformation is inverted:

$$\begin{bmatrix}q_{des}\\ r_{des}\end{bmatrix} = \mathbf{J_2^{-1}}\ (\eta_2)\begin{bmatrix}\dot{\theta}_{req}\\ \dot{\psi}_{req}\end{bmatrix} \tag{65}$$

5.6.Inner Loop Design – Fast Dynamics

Similarly to the attitude loop, the inner layer requires a linear compensation stage to bridge the gap between the commanded velocities and the required dynamic response. Since the Dynamic Inversion algorithm operates on state derivatives (accelerations), a PID controller is implemented to process the velocity tracking error vector, defined as $e_\nu = [u_{des} - u, q_{des} - q, r_{des} - r]^T$.

This controller computes the required body accelerations ($\dot{\nu}_{req}$), which serve as the virtual control inputs for the subsequent inversion block:

$$\dot{\nu}_{req} = \begin{bmatrix}\dot{u}_{req}\\ \dot{q}_{req}\\ \dot{r}_{req}\end{bmatrix} = K_P e_\nu + K_I \int e_\nu dt + K_D \dot{e_\nu} \tag{66}$$

These calculated accelerations represent the necessary dynamic change to eliminate velocity errors and are directly fed into the vehicle's inverse dynamics model.

The inner loop controls the fast dynamics of the vehicle, specifically the body angular rates (u, q, r). This loop linearizes the rotational dynamics by canceling out the nonlinearities.

Using the Lie Derivative notation as formulated in the design phase:

$$y = h(x) = \mathbf{C}\,x \tag{67}$$

$$\mathbf{C} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Then the rate vector can be derived as:

$$\dot{y} = \frac{\partial h}{\partial x}\frac{\partial x}{\partial t} = \mathbf{C}\dot{x} \tag{68}$$

$$\dot{y} = L_f h(x) + L_g h(x)u = F(x) + \mathbf{G}(\mathbf{x})u$$

For the angular rates, the relative degree is 1, meaning the control inputs appear directly in the first derivative of the states. The desired dynamics are enforced as:

$$u = \mathbf{G_{red}^{-1}}\left(\begin{bmatrix} \dot{u}_{des} \\ \dot{q}_{des} \\ \dot{r}_{des} \end{bmatrix} - F_{red}\right) \tag{69}$$

Simplification of the Control Matrix ($\mathbf{G_{red}}$): As noted in the implementation derivation, the full 12× 3 control matrix is reduced to a 3 × 3 matrix (G_{red}) by focusing on the active control channels. Based on the REMUS 100 configuration. Thus, the reduced matrix G_{red} and the reduced Dynamics vector F_{red} are constructed as:

$$\mathbf{G_{red}} = \begin{bmatrix} m - X_{\dot{u}} & 0 & 0 \\ 0 & \dfrac{M_{uu\delta_s}u^2}{I_{yy} - M_{\dot{q}}} & 0 \\ 0 & 0 & \dfrac{N_{uu\delta_r}u^2}{I_{zz} - N_{\dot{r}}} \end{bmatrix} \tag{70}$$

$$F_{red} = f_{dyn}([1,5,6]) \tag{71}$$

Here, indices 1, 5, and 6 correspond to Surge, Pitch, and Yaw dynamics respectively. The term $X_{\dot{u}}$

represents the thrust coefficient, while $M_{uu\delta_s}$ and $N_{uu\delta_r}$ are the control effectiveness coefficients derived from Prestero's fin lift models.

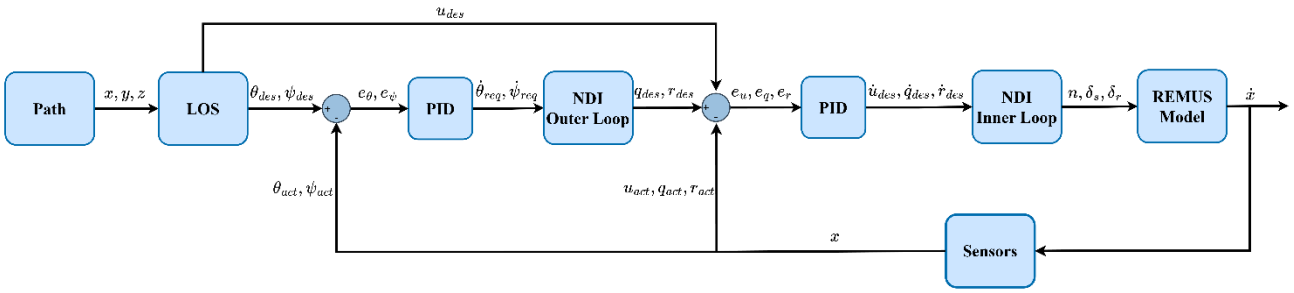

Figure 3: Cascaded Control Architecture

5.7.PID Gain Tuning via Error Dynamics and Pole Placement

The primary advantage of the Nonlinear Dynamic Inversion (NDI) method is its ability to cancel out the system's nonlinearities, effectively reducing the plant dynamics to a chain of integrators. This linearization allows for the analytical determination of PID controller gains using the Pole Placement technique, rather than relying on heuristic trial-and-error methods. This section details the derivation of the error dynamics and the calculation of the control gains based on the desired system performance.

5.7.1 Derivation of Error Dynamics

As derived in Equation (33), the NDI control law linearizes the plant such that the relationship between the virtual control input (ν) and the state derivative $(\dot{x})$ becomes a simple integrator:

$$\dot{x} = \nu \tag{72}$$

To control this linearized system, a Proportional-Integral (PI) control structure is employed for the virtual control input. Let r(t) be the reference signal and y(t) be the system output. The tracking error is defined as $e(t) = r(t) - y(t)$. The control law is formulated as:

$$\nu = \dot{r} + K_P e + K_I \int e \; dt \tag{73}$$

Substituting the linearized system dynamics ($\dot{y} = \nu$) into the control law, and assuming the reference is a step input ($\dot{r} \approx 0$), the closed-loop error dynamics can be derived as follows:

$$\dot{y} = K_P(r - y) + K_I \int (r - y)\ dt \tag{74}$$

Rearranging terms to express the equation in terms of error (e):

$$\dot{e} + K_P e + K_I \int e\ dt = 0 \tag{75}$$

Differentiating with respect to time yields the second-order homogeneous differential equation governing the error dynamics:

$$\ddot{e} + K_P \dot{e} + K_I e = 0 \tag{76}$$

In the Laplace domain, this corresponds to the characteristic equation:

$$s^2 + K_P s + K_I = 0 \tag{77}$$

5.7.2 Pole Placement Strategy

To ensure stability and achieve the desired transient response, the characteristic equation (Eq. 77) is matched with the standard second-order system characteristic polynomial:

$$s^2 + 2\zeta\omega_n s + \omega_n^2 = 0 \tag{78}$$

Here, ζ (damping ratio) determines the overshoot and stability margin, while ω_n (natural frequency) dictates the response speed (bandwidth). By equating the coefficients of Eq. 77 and Eq. 78, the PI gains are derived analytically:

$$K_P = 2\zeta\omega_n$$
$$K_I = \omega_n^2 \quad (79)$$

The derivative gain (K_D) is omitted in this design ($K_D \approx 0$) to avoid noise amplification, as the NDI algorithm inherently handles the system's dynamic rates.

5.7.3 Time-Scale Separation (TSS) and Parameter Selection

To maintain stability within the cascaded control architecture described in Section 4.1.5, the Time-Scale Separation (TSS) principle is applied. This principle requires that the inner loop dynamics be significantly faster than the outer loop dynamics to prevent loop interaction and actuator saturation.

1. **Inner Loop (Rate Control - q, r):**

This loop controls the fast body angular rates. The bandwidth is selected based on the physical limitations of the REMUS 100 actuators.

- **Target:** Fast response with minimal overshoot.
- **Parameters:** $\omega_{n,inner} = 1{,}5\ rad/s\ and\ \zeta = 0.9$
- **Resulting Gains:** $K_P = 2.7$, $K_I = 2.25$.

2. **Middle Loop (Attitude Control - θ, ψ)**

This loop generates rate commands for the inner loop. According to TSS, it is designed to be approximately 5 times slower than the inner loop to ensure the inner loop can track the commands without lag.

- **Target:** Critical damping to prevent overshoot in orientation.
- **Parameters:** $\omega_{n,middle} = 0.3\ rad/s\ and\ \zeta = 1$.
- **Resulting Gains:** $K_P = 0.6$, $K_I = 0.09$

The calculated gains for the REMUS 100 control system are summarized in Table 1 below.

Control Loop	Variable	Target ω_n(rad/s)	Target ζ	Calculated K_P	Calculated K_I
Inner Loop	Pitch Rate (q)	1.5	1	3.0	2.25
Inner Loop	Yaw Rate (r)	1.5	1	3.0	2.25
Middle Loop	Pitch Angle (θ)	0.3	1	0.6	0.09
Middle Loop	Heading Angle (ψ)	0.3	1	0.6	0.09

Table 1: Calculated PID Gains using Pole Placement

This analytical approach ensures that the controller parameters are grounded in the physical capabilities of the vehicle and theoretical stability criteria, rather than arbitrary tuning.

6. Simulation Results and Discussion

This section presents the simulation results obtained from the application of the Nonlinear Dynamic Inversion (NDI) control strategy to the REMUS 100 AUV. The primary objective of these simulations is to validate the theoretical stability analysis presented in Chapter 4 and to evaluate the tracking performance of the proposed 3-layer cascaded control architecture.

6.1. Performance of the Cascaded Architecture

Following the validation of the stability properties, the full 3-layer cascaded architecture (LOS Guidance + Attitude NDI + Rate NDI) was evaluated on a 3D path-following scenario. The reference path consists of waypoints requiring simultaneous depth changes and heading maneuvers.

Trajectory Tracking:

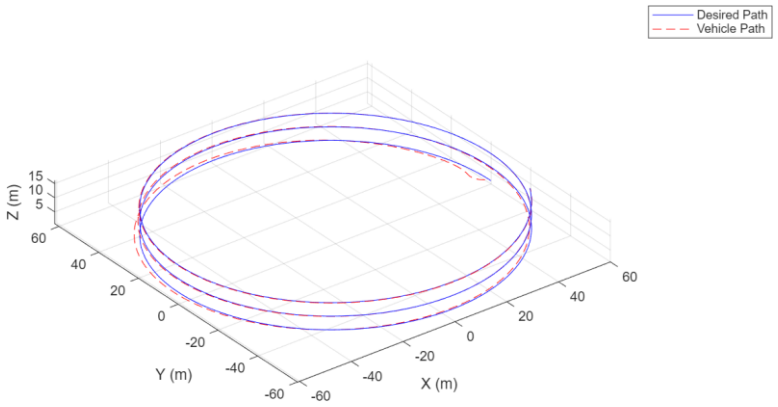

Figure 4: Trajectory Tracking Performance

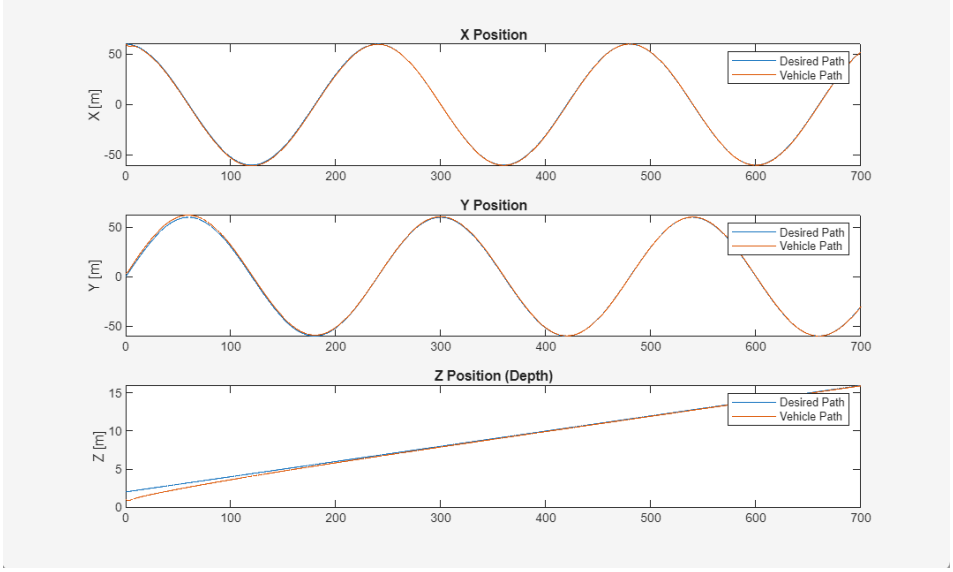

Figure 5: x,y,z Positions

Figure 3 depicts the 3D trajectory of the REMUS 100 compared to the desired path. The results show that the LOS guidance algorithm successfully generates appropriate pitch (θ_{des}) and heading (ψ_{des}) commands to eliminate cross-track and depth errors. The vehicle converges to the path with minimal overshoot.

Control Effort and Actuator Saturation:

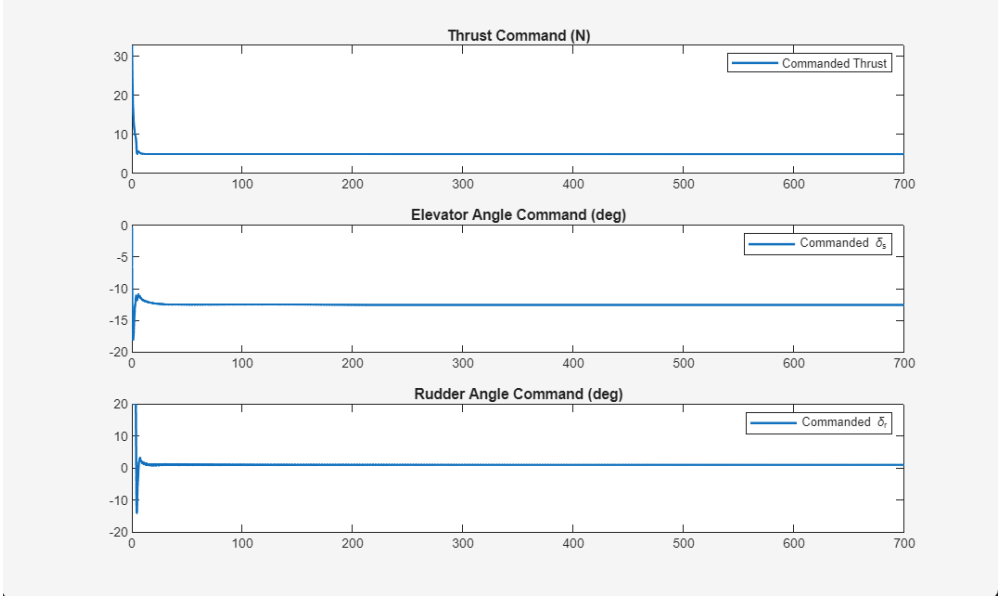

Figure 6: Actuator Commands

Figure 6 displays the control surface deflections (δ_s, δ_r) generated by the innermost NDI loop. The control signals remain smooth and within the physical saturation limits of the actuators ($\pm$ 30°).

6.2. Discussion

The simulation results provide conclusive evidence supporting the theoretical framework established in this study. The success of the proposed controller can be attributed to the rigorous handling of the zero dynamics:

Decoupling via Time-Scale Separation: The division of the control logic into "Guidance" (slow), "Attitude" (medium), and "Rate" (fast) loops aligns perfectly with the physical bandwidths of the AUV. This prevents the dynamic coupling between the fast rigid-body modes and the slow trajectory kinematics from destabilizing the system.

In conclusion, the proposed NDI-based cascaded architecture successfully solves the under-actuated path-following problem for the REMUS 100, ensuring both global trajectory convergence and local internal stability.

7. Conclusion

This thesis proposes to address this problem in designing a path-following control system with high performance capabilities in REMUS 100 Autonomous Underwater Vehicle (AUV), especially in overcoming difficulties in under-actuated systems and strongly nonlinear coupled hydrodynamic equations.

The major contribution is in the theoretical analysis of the system's internal stability through Zero Dynamics. The derivation of the system's internal dynamics on the zero dynamics analysis reveals the system's Non-Minimum Phase (NMP) characteristics when depth is chosen as the direct output for inversion. The combination of stern plane deflection and heave response results in an unstable internal mode, making it impossible to apply Input-Output Linearization directly to the system's depth dynamics. However, the system's attitude dynamics (pitch) are Minimum Phase.

Based on the above results, the Three-Layer Cascaded Control Architecture was designed and realized, based on the principle of Time-Scale Separation, decomposing the system into three layers, namely guidance, attitude, and rate loops, as follows:

Outer Loop: A Line-of-Sight (LOS) guidance method handles the slow translational motion, thus bypassing the NMP instability by posing the depth control problem as a regulation problem of the trajectory, rather than a state tracking problem.

Middle and Inner Loops: Nonlinear Dynamic Inversion (NDI) control is used in the stable attitude and rate dynamics, allowing for the accurate inversion of nonlinear effects such as Coriolis forces, gyroscopic torques, and hydrodynamic damping.

The simulation outcome validates the proposed control method, with the cascaded NDI control method providing better tracking for complex three-dimensional maneuvers, with good convergence of the paths while maintaining smooth control surface motion. The difference between direct depth inversion and the cascaded method highlights the importance of zero dynamics analysis for underactuated systems design. In conclusion, it is evident that Nonlinear Dynamic Inversion, as a control strategy within a hierarchically separated architecture, is a robust and efficient approach to controlling AUVs. This approach not only overcomes stability problems arising from non-minimum phase systems, it also leverages a complete nonlinear model of the vehicle to ensure consistent and high-performance operation over a wide range of operation. Future work can be based on enhancing this approach with adaptive components for real-time estimation of uncertain hydrodynamic parameters.

8. References

[1] National Oceanic and Atmospheric Administration (NOAA), "How much of the ocean have we explored?" .

[2] United Nations Educational, Scientific and Cultural Organization (UNESCO), "The Science of Ocean Exploration,".

[3] Christopher von Alt, "Autonomous Underwater Vehicles," ALPS Workshop, Woods Hole Oceanographic Institution, March 2003.

[4] Xianbo Xiang, Caoyang Yu, and Qin Zhang, "Robust Fuzzy 3D Path Following for Autonomous Underwater Vehicle Subject to Uncertainties," Computers and Operations Research, Vol. 84, 2017, pp. 165–177.

[5] Timothy Prestero, "Verification of a Six-Degree of Freedom Simulation Model for the REMUS Autonomous Underwater Vehicle," MIT-WHOI Joint Program Thesis, 2001.

[6] Kimon P. Valavanis, Denis Gracanin, Maja Matijasevic, Ramesh Kolluru, and Georgios A. Demetriou, "Control Architectures for Autonomous Underwater Vehicles," IEEE Control Systems Magazine, December 1997.

[8] Ahmet Burak Akyüz, Mehmet Berke Gür, and Ömer Tunahan Liman, "NDI ve Jenerik Algoritma ile Uçağa İstenilen Manevraları Yaptırma," Bahçeşehir University Research Paper, 2024.

[9] F. Zhang and F. Holzapfel, "Flight Control Using Physical Dynamic Inversion," AIAA Guidance, Navigation, and Control Conference, 2015.

[10] Harris, J., and Valasek, J., "Direct L1-Adaptive Nonlinear Dynamic Inversion Control for Command Augmentation Systems," AIAA Guidance, Navigation, and Control Conference, 2018.

[11] G. Meyer, R. Su, and L. R. Hunt, "Applications of Nonlinear Transformations to Automatic Flight Control," Automatica, Vol. 20, No. 1, 1984, pp. 103–107.

[12] S. H. Lane and R. F. Stengel, "Flight Control Design Using Nonlinear Inverse Dynamics," Automatica, Vol. 24, No. 4, 1988, pp. 471–483.

[13] Albostan, O., and Gökasan, M., "High Angle of Attack Manoeuvring Control of F-16 Aircraft Based on Nonlinear Dynamic Inversion and Eigenstructure Assignment," 7th European Conference for Aeronautics and Space Sciences, 2017.

[14] Thor L. F., Morten B., Roger S., “Line of Sight Path Following for Underactuated Marine Craft” , 2023

[15] W. Ariza Ramirez, “Gaussian Processes applied to system identification, navigation and control of underwater vehicles”, PhD Thesis, Australian Maritime College, 2019

[1] S. Devasia, "Nonlinear Inverse: Application to VTOL Aircraft," University of Washington, Department of Mechanical Engineering, Technical Notes.

[2] S. Devasia, D. Chen, and B. Paden, "Nonlinear inversion-based output tracking," *IEEE Transactions on Automatic Control*, vol. 41, no. 7, pp. 930–942, 1996.

[3] D. Chen and B. Paden, "Stable inversion of nonlinear non-minimum phase systems," *International Journal of Control*, vol. 64, no. 1, pp. 81–97, 1996.

[4] J.-J. E. Slotine, "Contraction Analysis and Nonlinear Control," *MIT Course Notes 2.152*, Massachusetts Institute of Technology, 2005.

[5] A. Isidori, *Nonlinear Control Systems*, 3rd ed. London: Springer-Verlag, 1995.

[6] L. Ye, Q. Zong, J. L. Crassidis, and B. Tian, "Output Redefinition-Based Dynamic Inversion Control for a Nonminimum Phase Hypersonic Vehicle," *IEEE Transactions on Industrial Electronics*, vol. 65, no. 4, pp. 3447–3457, 2018.

[7] P. B. Koganti and F. E. Udwadia, "Unified Approach to Modeling and Control of Rigid Multibody Systems," *Journal of Guidance, Control, and Dynamics*, vol. 40, no. 7, pp. 1–15, 2016.

[8] A. B. Akyüz, M. B. Gür, and Ö. T. Liman, "NDI ve Jenerik Algoritma ile Uçağa İstenilen Manevraları Yaptırma (Making the Aircraft Perform Desired Maneuvers with NDI and Generic Algorithm)," *Bahçeşehir University & TAI Research Paper*, 2024.

[9] F. Zhang and F. Holzapfel, "Flight Control Using Physical Dynamic Inversion," in *AIAA Guidance, Navigation, and Control Conference*, San Diego, CA, 2015.

[10] J. Harris and J. Valasek, "Direct L1-Adaptive Nonlinear Dynamic Inversion Control for Command Augmentation Systems," in *AIAA Guidance, Navigation, and Control Conference*, Kissimmee, FL, 2018.

[11] G. Meyer, R. Su, and L. R. Hunt, "Applications of Nonlinear Transformations to Automatic Flight Control," *Automatica*, vol. 20, no. 1, pp. 103–107, 1984.

[12] S. H. Lane and R. F. Stengel, "Flight Control Design Using Nonlinear Inverse Dynamics," *Automatica*, vol. 24, no. 4, pp. 471–483, 1988.

[13] O. Albostan and M. Gökaşan, "High Angle of Attack Manoeuvring Control of F-16 Aircraft Based on Nonlinear Dynamic Inversion and Eigenstructure Assignment," in *7th European Conference for Aeronautics and Space Sciences (EUCASS)*, 2017.

[14] R. C. van 't Veld, "Incremental Nonlinear Dynamic Inversion Flight Control: Stability and Robustness Analysis and Improvements," M.S. thesis, Dept. Control and Simulation, Delft Univ. of Technology, Delft, Netherlands, 2016.