COMPARATIVE ANALYSIS OF AUTOPILOT ARCHITECTURES FOR AIR DEFENSE MISSILES

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

MERVE ÇELİKBUDAK ALTINTAŞ

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
ELECTRICAL AND ELECTRONICS ENGINEERING

SEPTEMBER 2024

Approval of the thesis:

COMPARATIVE ANALYSIS OF AUTOPILOT ARCHITECTURES FOR AIR DEFENSE MISSILES

submitted by **MERVE ÇELİKBUDAK ALTINTAŞ** in partial fulfillment of the requirements for the degree of **Master of Science in Electrical and Electronics Engineering Department, Middle East Technical University** by,

Prof. Dr. Naci Emre Altun
Dean, Graduate School of **Natural and Applied Sciences** ______________

Prof. Dr. İlkay Ulusoy
Head of Department, **Electrical and Electronics Engineering** ______________

Prof. Dr. Mehmet Kemal Leblebicioğlu
Supervisor, **Electrical and Electronics Engineering, METU** ______________

Examining Committee Members:

Prof. Dr. Mehmet Önder Efe
Computer Engineering, Hacettepe University ______________

Prof. Dr. Mehmet Kemal Leblebicioğlu
Electrical and Electronics Engineering, METU ______________

Prof. Dr. Afşar Saranlı
Electrical and Electronics Engineering, METU ______________

Date: 05.09.2024

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Surname: Merve Çelikbudak Altıntaş

Signature :

ABSTRACT

COMPARATIVE ANALYSIS OF AUTOPILOT ARCHITECTURES FOR AIR DEFENSE MISSILES

Çelikbudak Altıntaş, Merve
M.S., Department of Electrical and Electronics Engineering
Supervisor: Prof. Dr. Mehmet Kemal Leblebicioğlu

September 2024, 103 pages

High precision control is essential during the ascent phase of surface-to-air defense missiles, due to the need for sudden attitude maneuvers to align with the tracking system. In this thesis, a comprehensive 6-DOF nonlinear missile model is developed to simulate the behavior of a tactical surface-to-air missile. Various nonlinear autopilot designs, including Nonlinear Active Disturbance Rejection Control (ADRC) and PID controllers, are implemented and compared under different architectural frameworks. These designs have been thoroughly examined and compared not only in terms of control performance but also regarding energy consumption.

To further explore the guidance accuracy, an acceleration controller is implemented and integrated as a third loop in a cascaded configuration with attitude controllers. This additional control loop is analyzed using PNG across selected autopilot architectures. The comparison focuses on their performance when engaging maneuvering targets, taking into account the adaptability and responsiveness of different control strategies under dynamic conditions.

In addition to performance evaluations, the study incorporated detailed models of disturbance effects and delays, enabling a thorough comparison of the various autopilot architectures. Specifically, control actuation system delays, and sensor measurement delays are considered for assessing how different controllers handle real-world system challenges. These analyses provide a deeper understanding of the trade-offs between response speed, system stability, and guidance performance, particularly when comparing the conventional PID control strategies with second-order nonlinear ADRC techniques.

This thesis contributes valuable insights to the missile control systems, focusing on stability, precision, and energy consumption. It also emphasizes the importance of disturbance rejection, and delay robustness in enhancing the overall performance, and reliability of missile guidance systems.

Keywords: Nonlinear Active Disturbance Rejection Control (ADRC), Proportional-Integral-Derivative (PID), Extended State Observer (ESO), Tracking Differentiator (TD), Nonlinear State Error Feedback (NLSEF), Air Defense Missile, Autopilot System, Disturbance Rejection

ÖZ

HAVA SAVUNMA FÜZELERİ İÇİN OTOPİLOT MİMARİLERİNİN KARŞILAŞTIRMALI ANALİZİ

Çelikbudak Altıntaş, Merve
Yüksek Lisans, Elektrik ve Elektronik Mühendisliği Bölümü
Tez Yöneticisi: Prof. Dr. Mehmet Kemal Leblebicioğlu

Eylül 2024 , 103 sayfa

Karadan havaya atılan hava savunma füzelerinin yükselme safhasında, hedefin takip sistemine uygun şekilde yönlenebilmesi için ani yönelim manevralarına ihtiyaç duyulmaktadır. Bu tezde, karadan havaya atılan taktik bir füzenin davranışını ve operasyonel konseptini daha gerçekçi bir şekilde modellemek amacıyla kapsamlı bir 6 serbestlik dereceli (6-DOF) doğrusal olmayan füze modeli geliştirilmiştir. Doğrusal Olmayan Aktif Bozucu Etki Bastırma Kontrolü (ADRC) ve PID kontrolcüleri de dahil olmak üzere doğrusal olmayan çeşitli otopilotlar, farklı mimari tasarımlarda modellenmiş ve karşılaştırılmıştır. Bu tasarımlar, kontrol performansları ve enerji tüketimleri açısından kapsamlı bir şekilde incelenmiş ve değerlendirilmiştir.

Güdüm performansını daha ayrıntılı analiz etmek amacıyla, ivme kontrolcüsü tasarlanarak seri bağlı bir mimaride en dıştaki üçüncü döngü olarak entegre edilmiştir. Bu ek kontrol döngüsü, seçili otopilot mimarilerinde PNG algoritması kullanılarak analiz edilmiştir. Seçilen otopilot mimarilerinin hedefi vurma performansları farklı dinamik koşullar altında karşılaştırılmıştır.

Kontrol ve güdüm performansı değerlendirmelerine ek olarak, çeşitli bozucu etkiler ve gecikmeler modellenerek farklı otopilot mimarileri bu etkiler altında kapsamlı bir şekilde karşılaştırılmıştır. Farklı kontrolcülerin operasyonel koşullardaki gecikmeye karşı dayanıklılıklarını değerlendirebilmek amacıyla kontrol tahrik sistemi gecikmeleri ve sensör ölçüm gecikmeleri dikkate alınmıştır. Bu analizler, özellikle geleneksel PID kontrol stratejileri ile ikinci dereceden doğrusal olmayan ADRC tekniklerinin tepki hızı, sistem kararlılığı ve güdüm performansı açısından karşılaştırılmasını sağlamıştır.

Bu tez, füze kontrol sistemlerindeki kararlılık, hassasiyet ve enerji tüketimi konularına odaklanmaktadır. Ayrıca, bozucu etkilerin bastırılması ve gecikme dayanıklılığının, füze kontrol ve güdüm sistemlerinin genel performansını ve güvenilirliğini artırmadaki önemi vurgulanmaktadır.

Anahtar Kelimeler: Doğrusal Olmayan Aktif Bozucu Etki Bastırma Kontrolü (ADRC), Oransal-İntegral-Türevsel (PID), Genişletilmiş Durum Gözlemcisi (GDG), İzleme Farklılaştırıcısı, Doğrusal Olmayan Durum Hatası Geri Besleme, Hava Savunma Füzesi, Otopilot Sistemi, Bozucu Etki Bastırma

In loving memory of my grandparents, Safiye and Şerif...

ACKNOWLEDGMENTS

I would like to express my deepest gratitude to my supervisor, Prof. Dr. Mehmet Kemal Leblebicioğlu, for his invaluable guidance, advice, and insights throughout the research. It has been such a great experience for me to work with him, not only in terms of my research but also in life, and I truly value our insightful conversations throughout this time. I would also like to extend my thanks to the jury members, Prof. Dr. Mehmet Önder Efe, and Prof. Dr. Afşar Saranlı for their valuable comments on this thesis.

I would also like to thank the Scientific and Technological Research Council of Turkey (TÜBİTAK) for the BİDEB 2228-A MSc scholarship.

I am profoundly thankful to Fahrettin Kağan İpek for his continuous encouragement, valuable advice, and support. My sincere thanks also go to my colleagues Hazel Demirel Bayrı, Atakan Süslü, and Tahir Yanık for their valuable support throughout this study.

I am deeply grateful to my mother, Fadile Çelikbudak, my father, Fehmi Çelikbudak, and my little sister, Cemre Çelikbudak Orhon, for their love, unwavering belief, and continued support.

Lastly, I extend my heartfelt thanks to my dear husband, Resul Sercan Altıntaş, for his never-ending support, patience, and the cheerful presence in my life every day. I cannot imagine what my life would be like without him.

TABLE OF CONTENTS

CHAPTERS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

LIST OF ABBREVIATIONS

ABBREVIATIONS

DOF	Degree of Freedom
EOM	Equations of Motion
ECI	Earth-Centered Inertial
ECEF	Eearh-Centered Earth-Fixed
a_x, a_y, a_z	Translational accelerations in the the body frame
p, q, r	Body rotational rates
ω	Angular velocity
m	Mass
$\vec{V}$	Linear velocity vector
$\vec{F}$	Force vector
$\vec{M}$	Moment vector
$\vec{H}$	Angular momentum vector
I_{xx}, I_{yy}, I_{zz}	Elements of inertia matrix
I_{xy}, I_{xz}, I_{yz}	Cross inertia terms
u, v, w	Translational velocity components in the the body frame
ϕ, θ, ψ	Euler angles (roll, pitch, yaw)
ρ	Density of the air
V_∞	Magnitude of the velocity of the missile with respect to air
M	Mach number
a	Speed of sound
α	Angle of attack
β	Side-slip angle
δ_a	Control surface deflection (Aileron)

δ_e	Control surface deflection (Elevator)
δ_r	Control surface deflection (Rudder)
X	Axial force in the positive x-direction
Y	Axial force in the positive y-direction
Z	Axial force in the positive z-direction
L	Rolling moment in the positive p-direction
M	Pitching moment in the positive q-direction
N	Yawing moment in the positive r-direction
C_x	Axial force coefficient
C_y	Side force coefficient
C_z	Normal force coefficient
C_l	Rolling moment coefficient
C_m	Pitching moment coefficient
C_n	Yawing moment coefficient
C_D	Drag force coefficient
C_L	Lift force coefficient
CP	Center of pressure
CG	Center of gravity
IMU	Inertial Measurement Unit
MEMS	Micro-Electromechanical System
B_a	Accelerometer bias error
SF_a	Accelerometer scale-factor error
MA_a	Accelerometer misalignment error
n_a	Accelerometer random noise
B_g	Gyroscope bias error
SF_g	Gyroscope scale-factor error
MA_g	Gyroscope misalignment error

n_g	Gyroscope random noise
CAS	Control Actuation System
DC	Direct Current
KVL	Kirchoff’s Voltage Law
R	Armature resistance
L	Armature inductance
e_a	Counter electromotive force
k_e	Counter electromotive force constant
$b_m\dot{\Theta}$	Friction torque
T_c	Load torque
T	Motor torque
k_m	Motor torque constant
PID	Proportional-Integral-Derivative
K_p	Proportional gain
K_i	Integral gain
K_d	Derivative gain
ADRC	Active Disturbance Rejection Control
TD	Tracking Differentiator
ESO	Extended State Observer
NLSEF	Nonlinear State Error Feedback
PSO	Particle Swarm Optimization
J	Cost function
ISE	Integral of Squared Error
IAE	Integral of Absolute Error
ITAE	Integral of Time-Weighted Absolute Error
ITSE	Integral of Time-Weighted Squared Error
PNG	Proportional Navigation Guidance

LOS	Line of sight
N	Navigation constant
P	Power
I	Current
V	Voltage
R	Resistance
E	Energy

CHAPTER 1

INTRODUCTION

1.1 Motivation and Problem Definition

Air defense missiles need to make agile maneuvers in a controlled manner in order to be able to execute the tracking system commands quickly and accurately. Therefore, tactical missile autopilots are used to achieve successful intercepts. These controllers can be composed of various architectures and designed in different ways. In the classical approach, missile autopilots are designed using linear control methods [1]. Generally, the system is linearized at certain trim points, controller behaviors are examined in the frequency domain, gains are calculated for certain points and controllers are designed. The large number of these design points is challenging in terms of processing load. In addition, especially the uncertainties of aerodynamic parameters, thrust uncertainties, nonlinear behaviors of actuators, fast changing parameters and non-modeled dynamics affect the precise modeling of missile systems [2]. These controllers designed for a system that cannot be precisely modeled may negatively affect the flight performance under disturbance effects. To address this, adaptive control techniques have been introduced in the literature to enhance robustness and improve performance in uncertain environments [3].

An adaptive controller attempts to perform real-time estimation of process uncertainties and then generates a control input to predict, counteract, or minimize undesirable deviations from the desired closed-loop system behavior. Beyond their adaptive capabilities, these controllers can also be designed to learn the patterns to improve the system performance over time [4]. For instance, a tracking error integrator can be regarded as a type of learning controller. It continuously accumulates and integrates the errors using both past and current data. Adaptive controller loops generate their output by integrating nonlinear functions of the system's tracking errors, enabling more dynamic and responsive control. While adaptive control is a powerful control strategy, it is not a solution for every control challenge. It has own of difficulties, including increased complexity, the need for real-time parameter estimation, and potential stability issues, particularly in highly uncertain or rapidly changing environments. Moreover, adaptive controllers require careful tuning and is sensitive to noise and disturbances, which can complicate their design, and implementation. In missile applications, robustness is crucial as the system must handle model uncertainties, external disturbances and varying conditions. Predictable and reliable flight performance is essential for ensuring successful operation since particularly in highly safety-critical applications.

Nonlinear Dynamic Inversion (NDI) is also commonly used in flight control problems requiring precise control and offers numerous advantages [5], [6]. However, it has certain limitations when dealing with disturbances. The main goal of NDI is to linearize the nonlinear dynamics of the system, making control easier. However, its reliance on accurate system models means that any uncertainties or inaccuracies can degrade performance. In the presence of external disturbances, NDI struggles due to its lack of inherent robustness.

Robust control theory focuses on the techniques for designing controllers that ensure precise and rapid responses, even when faced with uncertainties or disturbances in the plant model. In this thesis, Active Disturbance Rejection Control (ADRC) strategy is explored as a means of improving the robustness of air defense missiles. ADRC is a modern control strategy designed to increase the performance and robustness of various dynamic systems against uncertainties and external factors. By actively compensating internal and external disturbances, ADRC provides a flexible approach to missile control, improving not only stability but also accuracy under a wide range of operational conditions. Its most important feature is to increase the system stability against model uncertainties and external factors [7]. Therefore, in order to improve the system performance, ADRC controllers with different architectures are designed and compared in terms of performance and energy efficiency. Additionally, cascaded PID controller architectures are also implemented to provide a benchmark for comparison. Moreover, some disturbance effects are modeled and various autopilot architectures are compared in terms of the rejection capability and precision of missile guidance performance.

Although ADRC has been applied to the nonlinear control of various systems, no significant research has been found in the literature specifically focusing on the implementation of nonlinear ADRC for missile control systems. Linear ADRC (LADRC) has only been implemented as an attitude controller in [8], [9], and [10]. Due to the extreme agility, rapid dynamics and nonlinear model of air defense missiles, linear ADRC is often inadequate for achieving precise control. Therefore, a second-order nonlinear ADRC strategy is considered in this thesis, offering enhanced adaptability and robustness to effectively manage the complex and dynamic conditions in which these missiles operate.

1.2 Contributions and Novelties

The contribution to the literature can be summarized as follows:

- Implementation of a nonlinear second-order ADRC for missile control systems, introducing a novel approach in missile control and guidance.
- Detailed implementation and analysis of the control actuation system, for the comparison of power and energy consumption of different autopilot architectures using a detailed 6-DOF tactical nonlinear missile model.
- Extensive simulation-based verification and comparison of various autopilot structures, highlighting the performance, stability, and robustness under different operational scenarios.
- Discussion of the different autopilot architectures, analyzing their disturbance rejection capabilities and overall missile guidance accuracy.

1.3 The Outline of the Thesis

The thesis is composed of five chapters.

Chapter 1 includes the scope of the research and introduces the methods used in the thesis. In addition, similar research studies and some applications in the literature that inspired this thesis are explained.

Chapter 2 provides a comprehensive summary of the 6-DOF equations of motion that form the missile model. In addition to this, subsystem models including inertial measurement unit and control actuation system mathematical models are given in detail. Lastly, selected missile characteristics are presented.

In Chapter 3, the design and implementation of autopilot and guidance algorithms for tactical missiles are explored in detail. The chapter presents the mathematical models and tuning methods for both the PID controller and the nonlinear ADRC controller, considering their underlying principles and practical implementation. The chapter also presents various autopilot architectures, like hybrid architectures, including cascaded PID with parallel ADRC in both outer and inner loop configurations, are explored to identify their advantages and disadvantages in missile autopilot systems. Furthermore, the chapter covers the design of the acceleration controller as a third loop cascaded to attitude controllers, which plays a critical role in missile dynamics, and the implementation of Proportional Navigation Guidance (PNG), which is essential for ensuring the missile accurately follows its target.

Chapter 4 presents the simulation results. Performance comparison analysis is conducted to analyse the effectiveness of various controller architectures used in the thesis. The results are given highlighting important performance metrics such as response time and energy consumption. Furthermore, a comprehensive discussion of the results provides insight into the advantages and limitations of each structure. This chapter also aims to provide a comprehensive understanding of how each method performs under various disturbances and model uncertainties.

Chapter 5 provides a summary of the study with concluding remarks.

CHAPTER 2

MISSILE MODELING

In this chapter, a detailed overview of the 6-DOF equations of motion, which form the foundation of the missile model, is provided. The reference coordinate frames and the equations governing both translational and rotational motion are given to accurately simulate missile nonlinear dynamics and design effective control algorithms. Additionally, subsystem models such as the inertial measurement unit (IMU) and control actuation system (CAS) are described in detail. Furthermore, selected missile characteristics are presented.

2.1 Coordinate Systems

There are several coordinate systems used to define both the translational and rotational dynamics of a vehicle. The following coordinate systems are of primary importance for modeling vehicle dynamics. All these reference coordinate systems adhere to a right-handed orientation and feature orthogonal axes.

2.1.1 Body Fixed Coordinate System

The origin is located at the center of mass of the vehicle. In the aircraft convention, the x-axis points forward, the z-axis points downward, and the y-axis extends outward from the right wing. This coordinate system is fixed to the vehicle, meaning its orientation moves with the vehicle's motion. It's important to note that this coordinate system is not inertial, as its orientation is relative to the vehicle's motion, rather than remaining fixed in space. Body fixed coordinate system is illustrated in Figure 2.1.

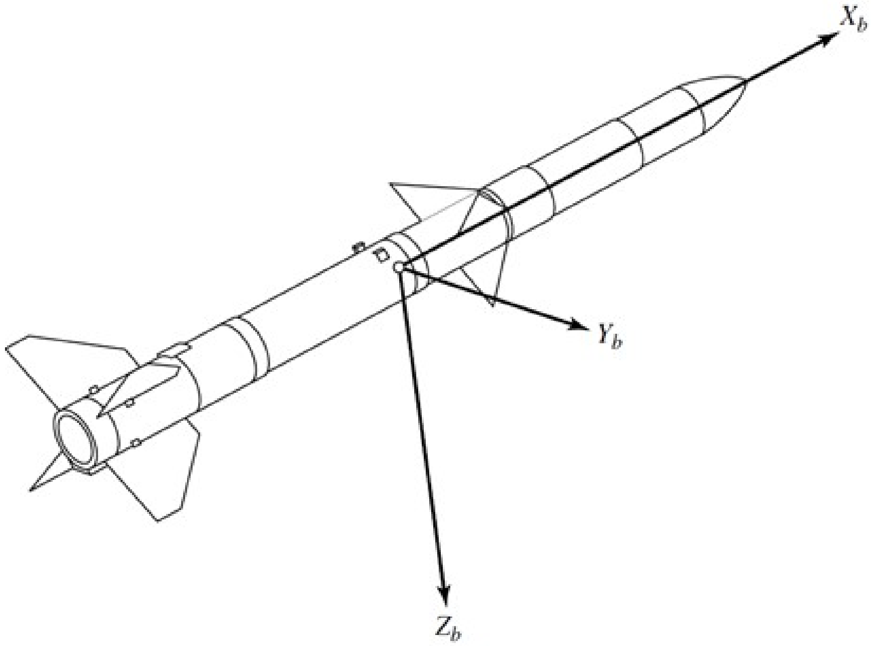


Figure 2.1: Body fixed coordinate system [11]

2.1.2 Earth-Centered Coordinate Systems

2.1.2.1 Earth-Centered Inertial (ECI) Coordinate System

The Earth-Centered Inertial Coordinate System is non-rotating, maintaining a fixed orientation relative to distant celestial objects. Its origin remains fixed at the center of the Earth, providing a stable reference point. The x-axis of this system aligns with the direction towards the vernal equinox, a point in space where the Earth's orbit intersects the celestial equator. Perpendicular to the x-axis, the y-axis extends 90 degrees to the east in the equatorial plane, providing a horizontal reference. The z-axis of the Earth-centered inertial coordinate system points northward, coinciding with the axis of rotation of the Earth.

2.1.2.2 Earth-Centered Earth-Fixed (ECEF) Coordinate System

The Earth-Centered Earth-Fixed Coordinate System is non-inertial, as it rotates along with the Earth's rotation. It is positioned at the center of the Earth and the origin of this coordinate system serves as a fixed point of reference. The x'-axis of this rotating coordinate system aligns with the intersection of the Earth's equatorial plane and the Greenwich Meridian, providing a directional reference relative to Earth's rotation. Perpendicular to the x'-axis, the y'-axis extends 90 degrees to the east in the equatorial plane, establishing a horizontal reference frame that follows Earth's rotation. The z'-axis of this rotating coordinate system points northward along the Earth's rotation axis, maintaining consistency with the Earth's rotational movement.

Representation of body, ECI and ECEF coordinate systems is shown in Figure 2.2.

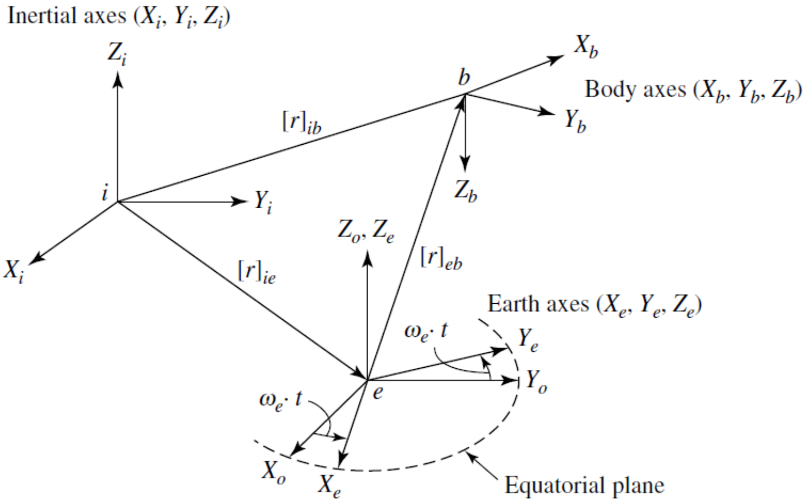


Figure 2.2: Representation of body, ECI and ECEF coordinate systems [11]

2.2 General Equations of Motion

The nonlinear equations of motion are developed by applying Newton's second law and law of conservation of linear momentum. The differential equations are written in the body fixed coordinate system, as detailed in references [12], [13], [14], [15], and [16].

It is assumed that the aircraft has six degrees of freedom (6-DOF). 6-DOF consists of three translational dynamics along the body axes and three rotational dynamics around the body axes. Translational linear velocities (u, v, w) and rotational rates (roll, pitch, and yaw rate) are illustrated in Figure 2.3.

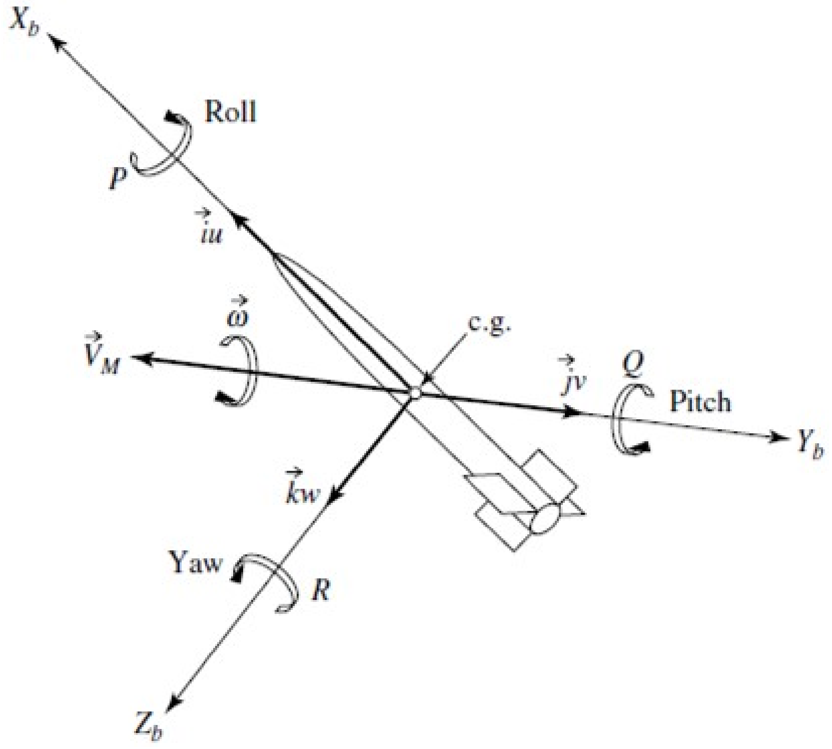


Figure 2.3: Representation of 6-DOF [11]

2.2.1 Translational Dynamics

The summation of all external forces acting on a body is equal to the time rate of change of the momentum of the body, and the summation of the external moments acting on the body is equal to the time rate of change of moment of momentum (angular momentum). Especially, Newton's laws of motion are given for a single particle. For a system of n particles, the linear momentum is the summation of the linear momenta of all particles in the system. Newton's second law in translational form can be written as:

$$\sum \vec{F} = \frac{d}{dt}\bigg|_I (m\vec{V}) \tag{2.1}$$

Note that Newton's second law is only valid in an Inertial Coordinate System.

In general, a vector A can be transformed from a fixed (e.g., inertial) to a rotating coordinate system by the relation. This is called as Coriolis Theorem.

$$\left(\frac{d\vec{A}}{dt}\right)_I = \left(\frac{d\vec{A}}{dt}\right)_B + \vec{\omega} \times \vec{A} \tag{2.2}$$

ω denotes angular velocity relative to the fixed (inertial) system and × denotes the vector cross product.

Using Coriolis Theorem and assuming mass is constant; the force equation can be expressed as:

$$\sum \vec{F} = m\left(\left.\frac{d}{dt}\right|_B \vec{V} + \vec{\omega} \times \vec{V}\right) \tag{2.3}$$

The linear velocity vector, $\vec{V}$, can be broken up into components $u, v,$ and w along the $X_B, Y_B,$ and Z_B axes. $\mathbf{i}, \mathbf{j}$, and $\mathbf{k}$ are unit vectors along the body axes.

$$\left.\frac{d}{dt}\right|_B \vec{V} = \dot{u}\mathbf{i} + \dot{v}\mathbf{j} + \dot{w}\mathbf{k} \tag{2.4}$$

The angular velocity vector, $\vec{\omega}$, can be broken up into components $p, q,$ and r along the $X_B, Y_B,$ and Z_B axes.

$$\vec{\omega} = p\mathbf{i} + q\mathbf{j} + r\mathbf{k} \tag{2.5}$$

The cross-product, $\vec{\omega} \times \vec{V}$, can be calculated as follows:

$$\vec{\omega} \times \vec{V} = \begin{vmatrix} \mathbf{i} & \mathbf{j} & \mathbf{k} \\ p & q & r \\ u & v & w \end{vmatrix} = (wq - vr)\mathbf{i} + (ur - wp)\mathbf{j} + (vp - uq)\mathbf{k} \tag{2.6}$$

Components of the force vector, $\vec{F}$, can be defined as:

$$\vec{F} = F_x\mathbf{i} + F_y\mathbf{j} + F_z\mathbf{k} \tag{2.7}$$

The force equation can be re-expressed as:

$$\begin{bmatrix} F_x \\ F_y \\ F_z \end{bmatrix} = m \begin{bmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{bmatrix} + m \begin{bmatrix} wq - vr \\ ur - wp \\ vp - uq \end{bmatrix} = m \begin{bmatrix} \dot{u} + wq - vr \\ \dot{v} + ur - wp \\ \dot{w} + vp - uq \end{bmatrix} \tag{2.8}$$

2.2.2 Rotational Dynamics

The moment equation in rotational form, analogous to Newton's force equation, can be formulated as:

$$\sum \vec{M} = \left.\frac{d}{dt}\right|_I \vec{H} \tag{2.9}$$

Angular momentum can be expressed as:

$$\vec{H} = I\vec{\omega}_I \tag{2.10}$$

The inertia matrix in the body frame can be defined as:

$$I = \begin{bmatrix} I_{xx} & -I_{xy} & -I_{xz} \\ -I_{yx} & I_{yy} & -I_{yz} \\ -I_{zx} & -I_{zy} & I_{zz} \end{bmatrix} \tag{2.11}$$

Moments of inertia and product of inertia are calculated as:

$$\begin{aligned} I_{xy} = I_{yx} &= \iiint xy\,dm, & I_{xx} &= \iiint (y^2 + z^2)\,dm \\ I_{xz} = I_{zx} &= \iiint xz\,dm, & I_{yy} &= \iiint (x^2 + z^2)\,dm \\ I_{yz} = I_{zy} &= \iiint yz\,dm, & I_{zz} &= \iiint (x^2 + y^2)\,dm \end{aligned} \tag{2.12}$$

Components of the angular momentum vector, $\vec{H}$, can be defined as:

$$\vec{H} = H_x\mathbf{i} + H_y\mathbf{j} + H_z\mathbf{k} \tag{2.13}$$

Roll moment, pitch moment, and yaw moment can be defined respectively as:

$$L = \frac{dH_x}{dt}, \quad M = \frac{dH_y}{dt}, \quad N = \frac{dH_z}{dt} \tag{2.14}$$

Using the angular momentum equation, the angular momentum vector can be expressed as:

$$\begin{bmatrix} H_x \\ H_y \\ H_z \end{bmatrix} = \begin{bmatrix} I_{xx} & -I_{xy} & -I_{xz} \\ -I_{yx} & I_{yy} & -I_{yz} \\ -I_{zx} & -I_{zy} & I_{zz} \end{bmatrix} \begin{bmatrix} p_I \\ q_I \\ r_I \end{bmatrix} \tag{2.15}$$

In terms of the body axes, assuming inertia is constant and using the Coriolis Theorem, the moment equation becomes:

$$\sum \vec{M} = \frac{d}{dt}\bigg|_B \vec{H} + \vec{\omega} \times \vec{H} \tag{2.16}$$

$$\sum \vec{M} = I \frac{d}{dt}\bigg|_B \vec{\omega} + \vec{\omega} \times I\vec{\omega} \tag{2.17}$$

$$\begin{bmatrix} L \\ M \\ N \end{bmatrix} = \begin{bmatrix} I_{xx} & -I_{xy} & -I_{xz} \\ -I_{yx} & I_{yy} & -I_{yz} \\ -I_{zx} & -I_{zy} & I_{zz} \end{bmatrix} \begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} + \vec{\omega} \times \begin{bmatrix} I_{xx} & -I_{xy} & -I_{xz} \\ -I_{yx} & I_{yy} & -I_{yz} \\ -I_{zx} & -I_{zy} & I_{zz} \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix} \tag{2.18}$$

$$\begin{bmatrix} L \\ M \\ N \end{bmatrix} = \begin{bmatrix} I_{xx}\dot{p} - I_{xy}\dot{q} - I_{xz}\dot{r} + q(-I_{xz}p - I_{yz}q + I_{zz}r) - r(-I_{xy}p + I_{yy}q - I_{yz}r) \\ -I_{xy}\dot{p} + I_{yy}\dot{q} - I_{yz}\dot{r} - p(-I_{xz}p + I_{yz}q + I_{zz}r) + r(I_{xx}p - I_{xy}q - I_{xz}r) \\ -I_{xz}\dot{p} - I_{xz}\dot{q} - I_{zz}\dot{r} + p(-I_{xy}p + I_{yy}q - I_{yz}r) - q(I_{xx}p - I_{xy}q - I_{xz}r) \end{bmatrix} \tag{2.19}$$

2.2.3 Rotational Kinematics

The orientation of the aircraft relative to the Earth-fixed coordinate system is obtained by introducing three sequential rotations over the Euler angles. In order to keep track of the three sequential rotations, the Earth fixed coordinate system is defined with $X_1Y_1Z_1$. The first rotation is produced by rotating the coordinate system $X_1Y_1Z_1$ over a psi angle so that the aircraft is taken to its heading angle after which the coordinate system is re-labeled $X_2Y_2Z_2$. Figure 2.4 shows the first rotation. The change of coordinates between the Earth-fixed coordinate system and the new coordinate system $X_2Y_2Z_2$ is given as:

$$\begin{bmatrix} X_2 \\ Y_2 \\ Z_2 \end{bmatrix} = \begin{bmatrix} \cos\psi & \sin\psi & 0 \\ -\sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_I \\ Y_I \\ Z_I \end{bmatrix} \tag{2.20}$$

The second rotation is produced by rotating the coordinate system $X_2Y_2Z_2$ over a pitch attitude angle after which the coordinate system is re-labeled $X_3Y_3Z_3$. Figure 2.4 shows the second rotation, where the change of coordinates between the coordinate system $X_2Y_2Z_2$ and the coordinate system $X_3Y_3Z_3$ is given as:

$$\begin{bmatrix} X_3 \\ Y_3 \\ Z_3 \end{bmatrix} = \begin{bmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{bmatrix} \begin{bmatrix} X_2 \\ Y_2 \\ Z_2 \end{bmatrix} \tag{2.21}$$

The final rotation is conducted about a roll angle to reach the body-fixed coordinate system. Figure 2.4 shows the final rotation, where the change of coordinates between the coordinate system $X_3Y_3Z_3$ and the body-fixed coordinate system is given as:

$$\begin{bmatrix} X_B \\ Y_B \\ Z_B \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & \sin\phi \\ 0 & -\sin\phi & \cos\phi \end{bmatrix} \begin{bmatrix} X_3 \\ Y_3 \\ Z_3 \end{bmatrix} \tag{2.22}$$

The relation between the Earth fixed coordinate system and the body fixed system can be defined as:

$$\begin{bmatrix} X_B \\ Y_B \\ Z_B \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & \sin\phi \\ 0 & -\sin\phi & \cos\phi \end{bmatrix} \begin{bmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{bmatrix} \begin{bmatrix} \cos\psi & \sin\psi & 0 \\ -\sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_I \\ Y_I \\ Z_I \end{bmatrix} \tag{2.23}$$

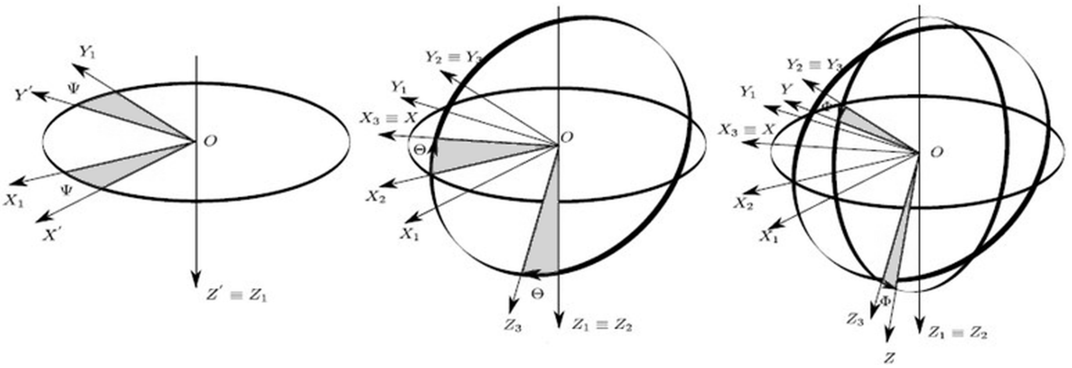


Figure 2.4: Representation of Euler angles

Apart from the six force and moment equation of motion, additional equations are required to solve for the unknown parameters. These extra equations are necessary because there are more than six unknowns due to the presence of the Euler angles in the force equations. Three equations are obtained by relating the p, q and r to the Euler rates. Note that the Euler angle rates are simply the time derivatives of the Euler angles.

Each of the three Euler angles rates $\dot{\phi}$, $\dot{\theta}$, and $\dot{\psi}$ can be conveniently displayed in one of the axis systems used in transforming a vector from inertial axis system to body axis system. The angular velocity vector, $\vec{\omega}$, can be written as:

$$\vec{\omega} = \dot{\phi}\mathbf{i} + \dot{\theta}\mathbf{j}_2 + \dot{\psi}\mathbf{k}_1 \tag{2.24}$$

Recall that the angular velocity vector, $\vec{\omega}$, can be broken up into components p, q, and r along the X_B, Y_B, and Z_B axes as:

$$\vec{\omega} = p\mathbf{i} + q\mathbf{j} + r\mathbf{k} \tag{2.25}$$

The relationship between the unit vectors of the body frame and intermediate frames using the direction cosine matrix can be expressed as:

$$\begin{aligned}\mathbf{j}_2 &= \cos\phi\,\mathbf{j} - \sin\phi\,\mathbf{k}\\ \mathbf{k}_1 &= -\sin\theta\,\mathbf{i} + \sin\phi\cos\theta\,\mathbf{j} + \cos\phi\cos\theta\,\mathbf{k}\end{aligned} \tag{2.26}$$

Substituting $\mathbf{j}_2$ and $\mathbf{k}_1$:

$$\begin{aligned}\vec{\omega} &= \dot{\phi}\mathbf{i} + \dot{\theta}(\cos\phi\,\mathbf{j} - \sin\phi\,\mathbf{k}) + \dot{\psi}(-\sin\theta\,\mathbf{i} + \sin\phi\cos\theta\,\mathbf{j} + \cos\phi\cos\theta\,\mathbf{k})\\ \vec{\omega} &= (\dot{\phi} - \dot{\psi}\sin\theta)\mathbf{i} + (\dot{\theta}\cos\phi + \dot{\psi}\sin\phi\cos\theta)\mathbf{j} + (-\dot{\theta}\sin\phi + \dot{\psi}\cos\phi\cos\theta)\mathbf{k}\end{aligned} \tag{2.27}$$

The relationship between angular velocity and the derivatives of Euler angles can be expressed as:

$$\begin{bmatrix} p \\ q \\ r \end{bmatrix} = \begin{bmatrix} 1 & 0 & -\sin\theta \\ 0 & \cos\phi & \sin\phi\cos\theta \\ 0 & -\sin\phi & \cos\phi\cos\theta \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \tag{2.28}$$

Reversely, once p, q, and r are given, the rates of the Euler angles can be expressed in the following, provided that $\cos\theta \neq 0$:

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & \sin\phi\tan\theta & \cos\phi\tan\theta \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi\sec\theta & \cos\phi\sec\theta \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix} \tag{2.29}$$

Representation of Euler angles and angular rates transformation is given in Figure 2.5.

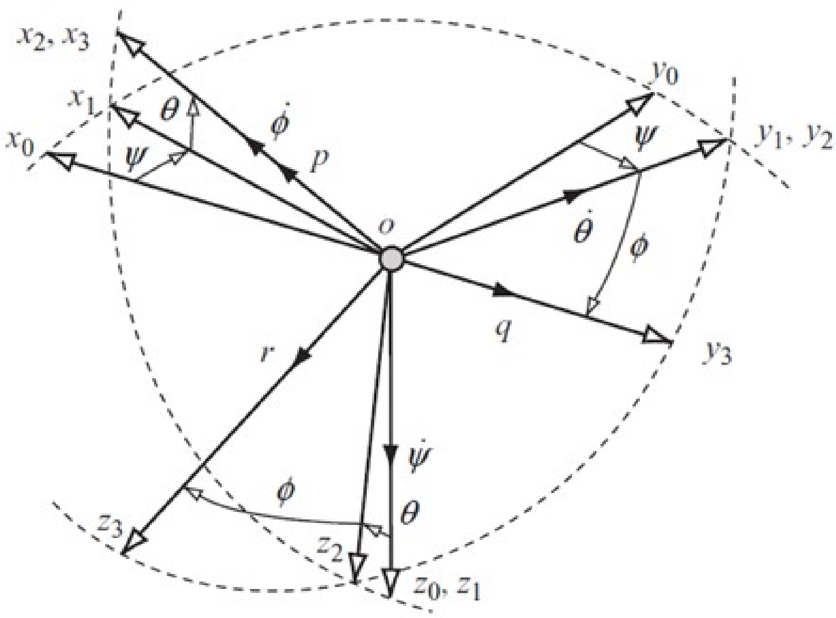


Figure 2.5: Representation of Euler angles and angular rates transformation [17]

2.2.4 Translational Kinematics

To determine velocity in the body frame, velocity derivatives are integrated. To calculate velocity and position in the inertial frame, velocity is transformed from the body frame to the inertial frame using the "direction cosine matrix," and subsequently integrated to obtain the position.

$$\vec{V}^I = C_B^I \vec{V}^B \tag{2.30}$$

$$\vec{X}^I = \int \vec{V}^I \, dt \tag{2.31}$$

C_B^I denotes the transformation matrix from the body frame to the inertial frame.

2.2.5 Forces and Moments

The forces acting on the system can be classified into aerodynamic forces, gravity forces, and thrust forces, while the moments can be categorized as aerodynamic moments and thrust moments.

2.2.5.1 Aerodynamic Forces and Moments

Calculating the aerodynamic forces and moments acting on a missile is a highly complex task, typically requiring detailed aerodynamic modeling. This modeling is performed by considering the missile's geometric characteristics, its kinematic parameters, and various control inputs that influence its aerodynamic behavior. The accuracy of these calculations is critical for understanding how the missile responds to different flight conditions and control inputs. To compute the aerodynamic coefficients, the modified Newtonian method is employed. Moreover, incorporating an accurate atmospheric model is essential to ensuring the precision of the aerodynamic simulations. In this study, the Extended International Standard Atmosphere (ISA) Model provided by MATLAB [21] is employed to accurately represent atmospheric conditions.

Aerodynamic forces defined in the body frame can be expressed as:

$$\mathbf{F}^B_{\text{Aero}} = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} QSC_x \\ QSC_y \\ QSC_z \end{bmatrix} \tag{2.32}$$

X is defined as axial force in the positive x-direction, Y is defined as side force in the positive y-direction, and Z is defined as normal force in the positive z-direction. Q denotes dynamic pressure and S denotes the surface area of the reference area. Dynamic pressure refers to the pressure of the air moving over the aircraft and is defined as:

$$Q = \frac{1}{2}\rho V_\infty^2 \tag{2.33}$$

where ρ is the density of the air and V_∞ is the magnitude of the velocity of the aircraft with respect to the air.

Mach number is defined as:

$$M = \frac{V}{a} \tag{2.34}$$

where a is the speed of sound.

Angle of attack (α) and angle of sideslip (β), which heavily influence aerodynamic forces and moments, are defined as:

$$\begin{aligned} \alpha &= \arctan\left(\frac{w}{u}\right) \\ \beta &= \arcsin\left(\frac{v}{V}\right) \end{aligned} \tag{2.35}$$

Generally, aerodynamic coefficients are nonlinear functions influenced by various factors such as Mach number (M), angle of attack (α), angle of sideslip (β), control surface deflections like aileron (δ_a), elevator (δ_e), rudder (δ_r), and body rotational rates (p, q, r).The dependence of aerodynamic force coefficients is expressed as follows:

$$\begin{bmatrix} C_x \\ C_y \\ C_z \end{bmatrix} = \begin{bmatrix} C_x(M, \alpha, \beta, \delta_a, \delta_e, \delta_r, q, r) \\ C_y(M, \alpha, \beta, \delta_a, \delta_r, p, r) \\ C_z(M, \alpha, \beta, \delta_e, q) \end{bmatrix} \tag{2.36}$$

Aerodynamic moments defined in the body frame can be expressed as:

$$\mathbf{M}^B_{\text{Aero}} = \begin{bmatrix} L \\ M \\ N \end{bmatrix} = \begin{bmatrix} QSC_l l_{\text{ref}} \\ QSC_m l_{\text{ref}} \\ QSC_n l_{\text{ref}} \end{bmatrix} \tag{2.37}$$

L is defined as the rolling moment in the positive p-direction, M is defined as the pitching moment in the positive q-direction, and N is defined as the yawing moment in the positive r-direction. l_{ref} denotes the reference length. The dependence of aerodynamic moment coefficients is expressed as follows:

$$\begin{bmatrix} C_l \\ C_m \\ C_n \end{bmatrix} = \begin{bmatrix} C_l(M, \alpha, \beta, \delta_a, \delta_r, p, r) \\ C_m(M, \alpha, \beta, \delta_e, q) \\ C_n(M, \alpha, \beta, \delta_a, \delta_r, p, r) \end{bmatrix} \tag{2.38}$$

2.2.5.2 Gravitational Forces

The gravity-originated force components are expressed as below in the body frame:

$$F^B_{\text{Gravity}} = \begin{bmatrix} F_{G_x} \\ F_{G_y} \\ F_{G_z} \end{bmatrix} = \begin{bmatrix} -mg\sin\theta \\ mg\sin\phi\cos\theta \\ mg\cos\phi\cos\theta \end{bmatrix} \tag{2.39}$$

2.2.5.3 Thrust Forces

The solid propellant rocket motor concept is utilized during the boost phase, with the main engine featuring a rectangular thrust profile. In the body coordinate frame, thrust force can be described as:

$$F^B_{\text{Thrust}} = \begin{bmatrix} F_T \\ 0 \\ 0 \end{bmatrix} \tag{2.40}$$

2.3 Subsystem Models

2.3.1 Inertial Measurement Unit (IMU) Model

Missile's linear accelerations and angular rates are measured about three reference axes with the help of inertial measurement unit (IMU). A typical IMU consists of three orthogonal gyroscopes and accelerometers which have some deterministic and stochastic errors due to the nature of inertial sensors and the calibration processes. To implement these errors Honeywell's HG1930CA50 model tactical grade IMU is selected [18]. It has three tactical grade MEMS based gyroscopes and accelerometers.

The performance characteristics of inertial sensors are given in Table 2.1, and 2.2. Since the scale factor and misalignment values are not indicated in the datasheet, typical tactical grade scale factor and misalignment repeatability values are considered in the implementation.

Table 2.1: Accelerometer error parameters

Parameter	**Unit** (1σ)	**Value**
Bias Repeatability	µg	500
Scale Factor Repeatability	ppm	1000
Misalignment Repeatability	µrad	1000
Bias In-Run Stability	µg	200
Velocity Random Walk	fps/$\sqrt{\text{h}}$	0.1

Accelerometer error is modeled using Table 2.1 and Equation 2.41 [19]:

$$\begin{bmatrix} \hat{a}_x \\ \hat{a}_y \\ \hat{a}_z \end{bmatrix} = \begin{bmatrix} SF_{ax} & MA_{ayx} & MA_{azx} \\ MA_{axy} & SF_{ay} & MA_{azy} \\ SF_{axz} & MA_{ayz} & SF_{az} \end{bmatrix} \begin{bmatrix} a_x \\ a_y \\ a_z \end{bmatrix} + \begin{bmatrix} B_a x \\ B_a y \\ B_a z \end{bmatrix} + \begin{bmatrix} n_a x \\ n_a y \\ n_a z \end{bmatrix} \tag{2.41}$$

$\hat{a}$: Accelerometer measurement output

B_a : Accelerometer bias error

SF_a : Accelerometer scale-factor error

MA_a : Accelerometer misalignment error

a : Accelerations acting along the vehicle

n_a : Accelerometer random noise

Table 2.2: Gyroscope error parameters

Parameter	**Unit** (1σ)	**Value**
Bias Repeatability	°/h	7
Scale Factor Repeatability	ppm	500
Misalignment Repeatability	μrad	1000
Bias In-Run Stability	°/h	0.25
Angular Random Walk	$°/\sqrt{\text{h}}$	0.06

Gyroscope error is modeled using Table 2.2 and Equation 2.42 [19]:

$$\begin{bmatrix} \hat{g}_x \\ \hat{g}_y \\ \hat{g}_z \end{bmatrix} = \begin{bmatrix} SF_{gx} & MA_{gyx} & MA_{gzx} \\ MA_{gxy} & SF_{gy} & MA_{gzy} \\ SF_{gxz} & MA_{gyz} & SF_{gz} \end{bmatrix} \begin{bmatrix} g_x \\ g_y \\ g_z \end{bmatrix} + \begin{bmatrix} B_g x \\ B_g y \\ B_g z \end{bmatrix} + \begin{bmatrix} n_g x \\ n_g y \\ n_g z \end{bmatrix} \tag{2.42}$$

$\hat{g}$: Gyroscope measurement output

B_g : Gyroscope bias error

SF_g : Gyroscope scale-factor error

MA_g : Gyroscope misalignment error

g : Applied angular rates

n_g : Gyroscope random noise

2.3.2 Control Actuation System (CAS)

The armature-controlled DC servomotor system with a PI controller is modeled for the aerodynamic control of the fins. Armature-controlled DC servomotor system is depicted in Figure 2.6. This motor is an electromechanical system comprising both an electrical and a mechanical subsystem.

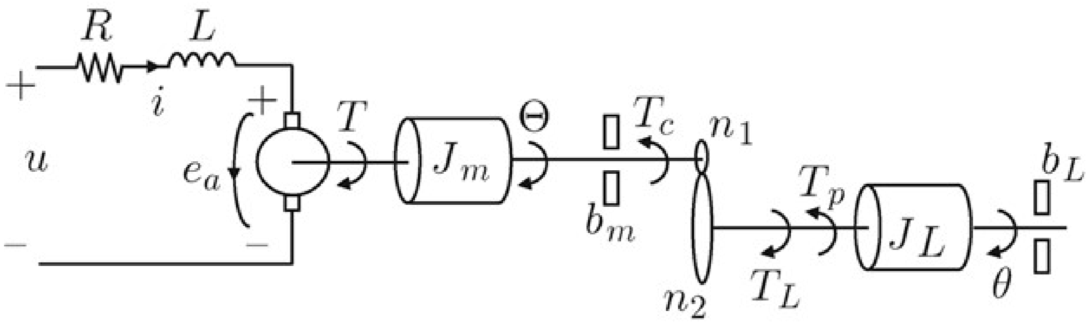


Figure 2.6: Armature-controlled DC servomotor system schematic diagram [20]

R : Armature resistance

L : Armature inductance

u : Voltage applied to the motor armature terminals

i : Electric current through the armature circuit

e_a : Counter electromotive force

T : Generated torque

T_c : Torque seen at the motor shaft

T_L : Torque applied by the motor on the load

Θ : Motor angular position

θ : Load angular position

T_p : Torque due to an external disturbance

J_m : Inertia of the motor's rotor

b_m : Motor viscous friction coefficient

J_L : Load inertia

b_L : Load viscous friction coefficient

n_1 : Teeth number at the motor axis

n_2 : Teeth number at the load axis

2.3.2.1 The Motor Electrical Subsystem Mathematical Model

According to Kirchoff's voltage law (KVL), the equation that governs the electrical part is:

$$u = Ri + L\frac{di}{dt} + e_a \tag{2.43}$$

e_a, the counter electromotive force, is proportional to the angular speed of the motor and can be expressed as:

$$e_a = k_e\dot{\Theta} \tag{2.44}$$

where k_e is the counter electromotive force constant.

2.3.2.2 The Motor Mechanical Subsystem Mathematical Model

Newton's second law must be applied individually to both the motor's rotor and its load.

- Motor's Rotor Model

In the context of rotational motion, Newton's Second Law can be applied to torques applied to the inertia using inertia and angular acceleration as:

$$J_m\ddot{\Theta} = T - b_m\dot{\Theta} - T_c \tag{2.45}$$

The term $b_m\dot{\Theta}$ represents the friction torque and T_c denotes the load torque with the negative sign indicating that these two torques oppose the motion of inertia. T, the motor torque, is proportional to the current as indicated in the following equation:

$$T = k_m i \tag{2.46}$$

where k_m is the motor torque constant.

- Load Model

Since there is a gear box, the following equations are considered:

$$\begin{aligned}\Theta &= n\theta \\ n &= \frac{n_2}{n_1} \\ T_L &= nT_c\end{aligned} \tag{2.47}$$

Newton's second law can be applied to the load as indicated following equation:

$$J_L\ddot{\theta} = T_L - b_L\dot{\theta} - T_p \tag{2.48}$$

The motor's mechanical subsystem mathematical model is obtained as:

$$\begin{aligned}J_m\ddot{\Theta} &= k_m i - b_m\dot{\Theta} - \frac{T_L}{n} \\ J_m\ddot{\Theta} &= k_m i - b_m\dot{\Theta} - \frac{J_L\ddot{\theta} + b_L\dot{\theta} + T_p}{n} \\ nJ_m\ddot{\theta} &= k_m i - b_m\dot{\Theta} - \frac{J_L\ddot{\theta} + b_L\dot{\theta} + T_p}{n} \\ (n^2 J_m + J_L)\ddot{\theta} + (n^2 b_m + b_L)\dot{\theta} &= nk_m i - T_p\end{aligned} \tag{2.49}$$

Defining:

$$J = n^2 J_m + J_L, \quad b = n^2 b_m + b_L \tag{2.50}$$

The motor's mechanical subsystem can be rewritten as:

$$J\ddot{\theta} + b\dot{\theta} = nk_m i - T_p \tag{2.51}$$

The final mathematical model can be expressed as:

$$\begin{aligned}L\frac{di}{dt} &= u - Ri - nk_e\dot{\theta} \\ J\ddot{\theta} &= -b\dot{\theta} + nk_m i - T_p\end{aligned} \tag{2.52}$$

These two differential equations constitute the mathematical model of the DC motor-load system. By solving these equations, the load position θ can be determined as a function of time, provided all the constants and the applied voltage at the armature terminals are known. By applying the Laplace transform with zero initial conditions, the following results are obtained:

$$\begin{aligned} I(s) &= \frac{1}{sL+R}\left[U(s) - nk_e s\theta(s)\right] \\ \theta(s) &= \frac{1}{s^2J+sb}\left[nk_m I(s) - T_p(s)\right] \end{aligned} \tag{2.53}$$

Armature-controlled DC servomotor system block diagram is represented in Figure 2.7.

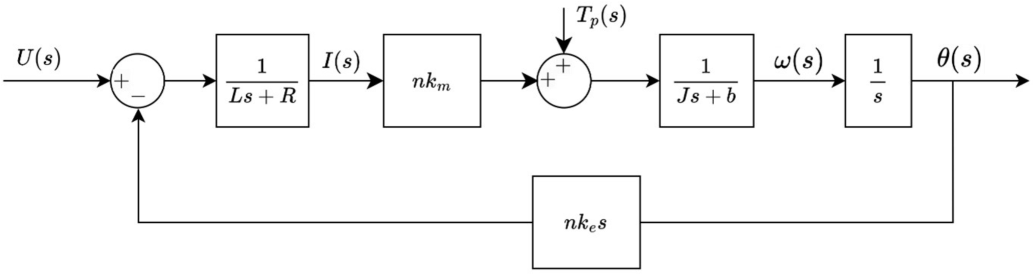


Figure 2.7: Armature-controlled DC servomotor system block diagram

Armature-controlled DC motor with PI controller block diagram is shown in Figure 2.8.

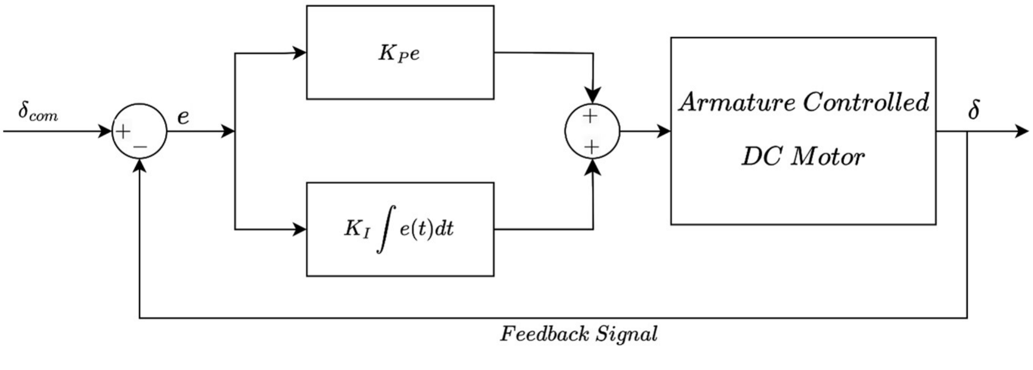


Figure 2.8: Armature-controlled DC motor with PI controller block diagram

Motor and PI parameters used for the model are given in Table 2.3.

Table 2.3: Motor and PI parameters

Parameters	**Representation**	**Unit**	**Value**
Armature resistance	R	Ω	5.5
Armature inductance	L	H	0.003
Voltage	u	V	24
Current limit	i_{max}	A	3
Transformation ratio	n	-	5
Inertia	J	kgm^2	0.001
Counter electromotive force constant	k_e	Vs/rad	0.025
Motor torque constant	k_m	Nm/A	1.6
Viscous friction coefficient	b	-	0.1
Proportional gain	K_p	-	35
Integral gain	K_I	-	30

Step response of the CAS is shown in Figure 2.9.

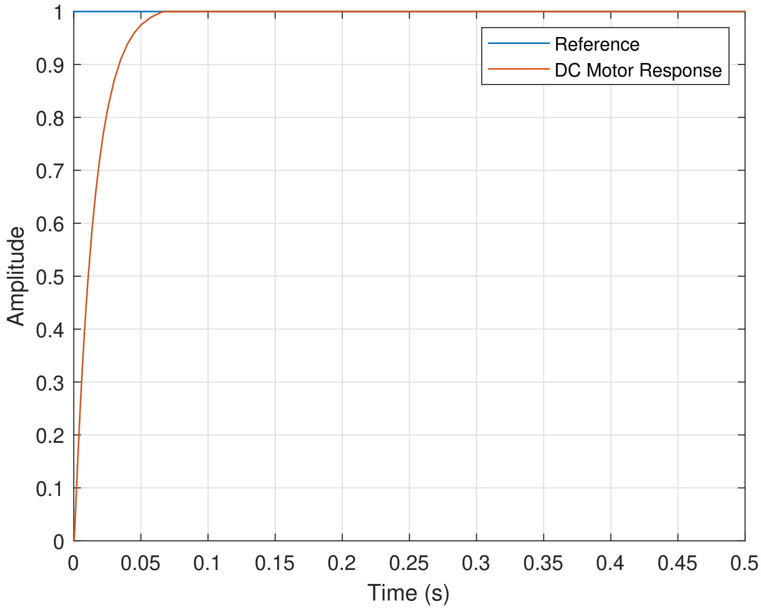


Figure 2.9: Step response of the CAS

2.4 Missile Characteristics

In this study, a tactical missile performing an atmospheric flight is considered. Tactical missiles generally consist of several key components designed to fulfill specific functions. Guidance section, warhead, propulsion system, control section and sensors/seekers are the main parts typically found in a tactical missile. The guidance section generally accommodates a seeker for target tracking and is responsible for calculating the necessary adjustments to steer the missile towards its intended destination. The control section contains control surfaces such as fins, canards, wings or thrust vectoring mechanism and maneuvers the control surfaces in response to guidance commands. In this research, wings and fins have been selected as control surfaces, and the specific missile characteristics are given in Table 2.4.

6-DOF nonlinear missile model is implemented considering the parameters listed in Table 2.4 and using MATLAB [21]. Additionally, the missile is treated as a rigid body in the derivation of the equations of motion and modeling, meaning that elastic deformations are neglected in the simulations.

Table 2.4: Missile characteristics

Parameters	**Unit**	**Value**
Total weight	kg	100
Explosive weight	kg	48
Case weight	kg	52
Inertia around X axis	kgm^2	0.28
Inertia around X axis (after burnout)	kgm^2	0.15
Inertia around Y axis	kgm^2	33.47
Inertia around Y axis (after burnout)	kgm^2	17.41
Length	m	2
Diameter	m	0.15
Center of gravity from nose	m	0.85
Center of gravity from nose (after burnout)	m	0.55
Total thrust	kNs	100

Mass and CG changing with time is given in Figure 2.10.

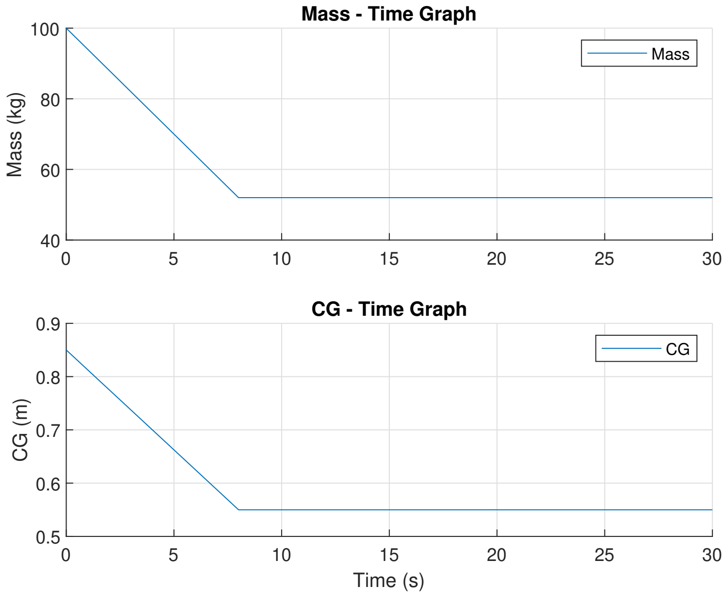


Figure 2.10: Mass and CG change vs time

In this thesis, since the selected missile needs to react quickly, the wings are designed as shown in Figure 2.11. Since the static margin, which is defined as the distance between the missile's center of pressure (CP) and center of gravity (CG), plays a critical role in the design. The negative static margin indicates that the CP is positioned ahead of the CG leading to an intentionally unstable missile configuration. The missile configurations which have negative static margin allow the missile to achieve higher maneuverability and faster response times. By selecting a configuration with a negative static margin, the missile can quickly adjust its trajectory to respond effectively to the target's movements, thus enhancing its interception capabilities.

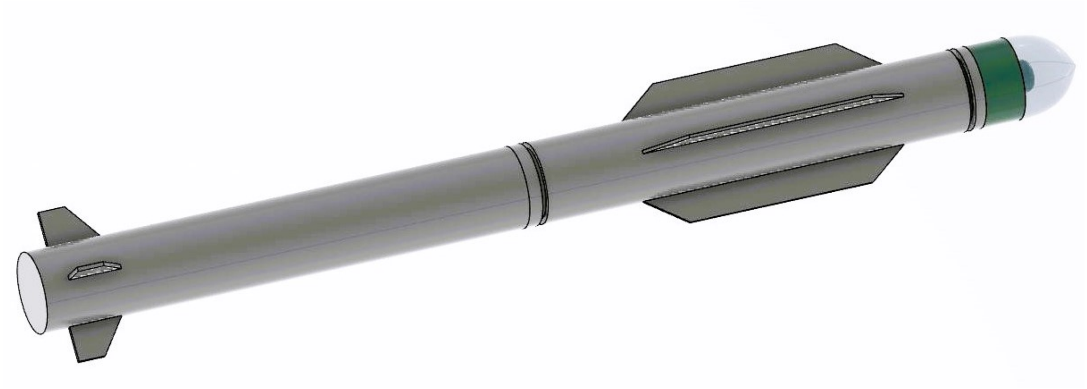


Figure 2.11: 3D model of the selected missile configuration

CHAPTER 3

AUTOPILOT AND GUIDANCE DESIGN

The selected tactical missile system model is nonlinear, and the aim of this thesis to design the controllers without linearizing the plant, allowing for a more accurate representation of the system dynamics. This approach also enhances the performance of the system in case of unmodeled system dynamics and external disturbances.

3.1 PID Controller

In this research, first of all, Proportional-Integral Derivative (PID) controller structure is considered for the selected nonlinear missile configuration. Various controller types, including angle, angular rate, acceleration, altitude, and speed controllers, are being evaluated in this study. The controlled variable's ability to sufficiently track the command input rapidly and accurately has been considered.

The PID controller consists of three components: proportional, integral, and derivative parts. The proportional component adjusts the output proportionally to the current error. The pure proportional controller has the inability to eliminate steady state error, hence the integral part is added. The integral component adjusts the output based on the accumulated error. By incorporating the integral part, steady-state error can be mitigated. However, the integration of error over time can lead to saturation of the control variable, known as integrator windup [22]. To address this issue, anti-windup strategies should be implemented to prevent control saturation and maintain stability. Lastly, the derivative action adjusts the output proportionally to the rate of change of the error. Transfer function of PID controller is given as:

$$C_p(s) = K_p + \frac{K_i}{s} + K_d s \tag{3.1}$$

where K_p is the proportional gain, K_i is the integral gain, and K_d is the derivative gain. In this configuration, known as the parallel form of the PID controller, all three actions (proportional, integral, and derivative) are fully independent and separated. This form is easy to implement, and its topology is shown in Figure 3.1. This block-diagram represents the closed-loop PID control system. The controller calculates the error "e" by subtracting the feedback signal "y" from the set point. The proportional, integral, and derivative terms are computed separately and then combined to create output signal, and subsequently applied to the plant.

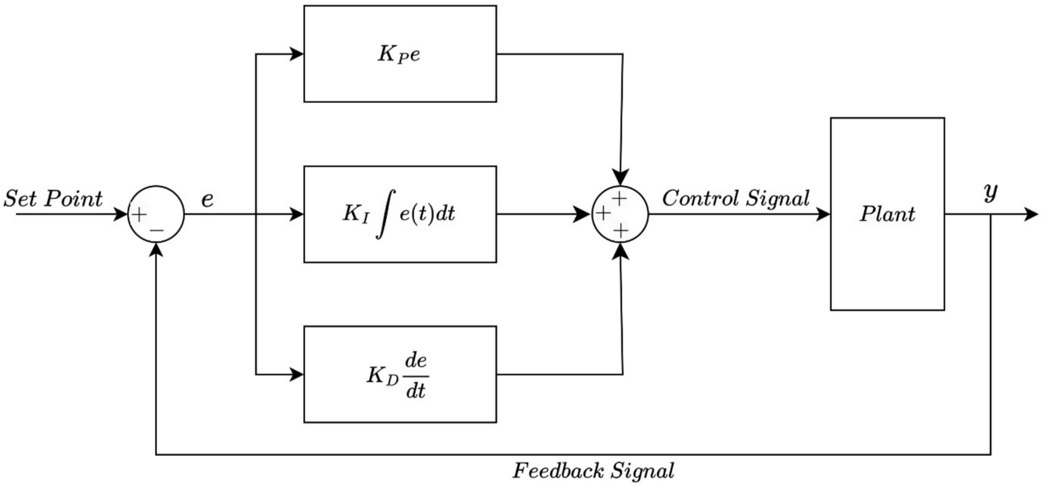


Figure 3.1: PID controller block diagram

The selection of K_p, K_i, K_d controller parameters is a critical task that requires careful consideration. A PID controller effectively combines the benefits of both PI and PD controllers, allowing it to operate efficiently across different frequency regions. Specifically, the PI component is more effective in the low-frequency range, where it improves the steady-state response by reducing steady-state error. Meanwhile, the PD component is more effective in the high-frequency range, where it enhances the transient response by anticipating the system's behavior and countering rapid changes [23].

As a result, the PID controller can be utilized in control system designs that require an optimal balance between transient and steady-state performance. By properly adjusting the PID parameters, the controller can simultaneously solve problems like overshoot, settling time, and steady-state error, resulting in a more robust and responsive system. PID tuning will be detailed in Section 3.3.1.

3.2 ADRC Controller

Active Disturbance Rejection Control (ADRC) is a modern control strategy designed to enhance the performance and robustness of various dynamic systems against uncertainties and external disturbances [7]. Its most significant feature is its ability to improve system robustness against model uncertainties and external disturbances through an error-driven approach [24]. Furthermore, its performance does not rely on precise mathematical models [2].

3.2.1 Linear ADRC Controller

Linear ADRC is designed for systems that can be modeled using linear dynamics. Unlike traditional control methods, ADRC does not need an accurate plant model and focuses on actively rejecting disturbances and compensating for system uncertainties. In linear ADRC, the assumption of linear system behavior simplifies both the observer design and the control law, making it less complex than its nonlinear version.

Linear ADRC consists of three key components: the Extended State Observer (ESO), a linear feedback controller, and a disturbance rejection mechanism.

The ESO is critical to linear ADRC, as it estimates both the system states and the total disturbance in real-time. For linear systems, the observer is designed using linear equations, which enhances computational efficiency and ease of implementation. The estimated disturbance is used to adjust the control signal, keeping the system on its desired trajectory.

Linear ADRC often uses a Proportional (P) or Proportional-Derivative (PD) controller as the feedback mechanism. These controllers use the system's observed states and estimated disturbances to generate the appropriate control signal. The simplicity of the linear feedback structure enables faster response times and easier tuning for systems exhibiting linear behavior.

A key advantage of linear ADRC is its ability to reject both external disturbances and internal model uncertainties. The ESO continuously estimates the total disturbance, and the controller adjusts the control signal accordingly to actively mitigate its effects. This leads to enhanced robustness and improved performance, even when there are inaccuracies or uncertainties in the model.

Linear ADRC is more straightforward to implement and tune than the nonlinear version, making it particularly well-suited for systems that can be approximated as linear or operate within a range where linear dynamics dominate.

First-order linear ADRC block diagram is given in Figure 3.2.

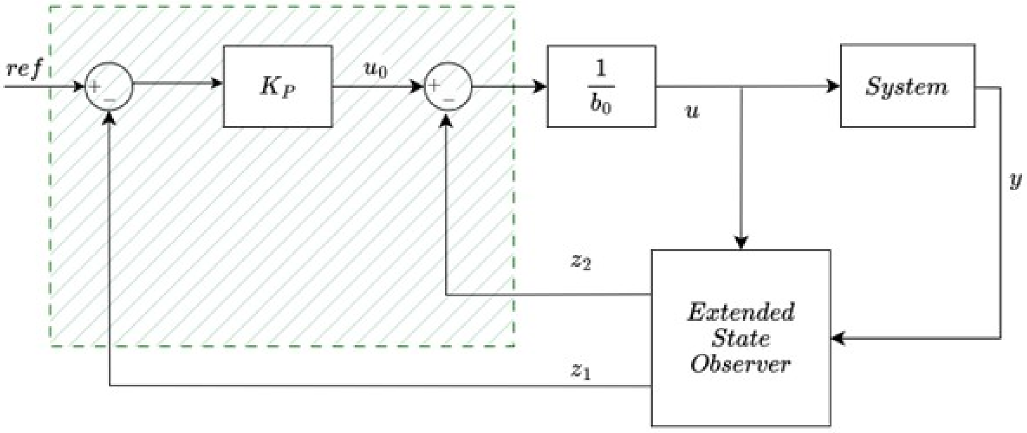


Figure 3.2: First-order linear ADRC block diagram

Second-order linear ADRC block diagram is given in Figure 3.3. This system is classified as second-order due to the incorporation of a derivative term, which allows the controller to account for both the system's state and the rate of change. This addition enhances the controller's ability to respond effectively to disturbances and dynamic changes in the system.

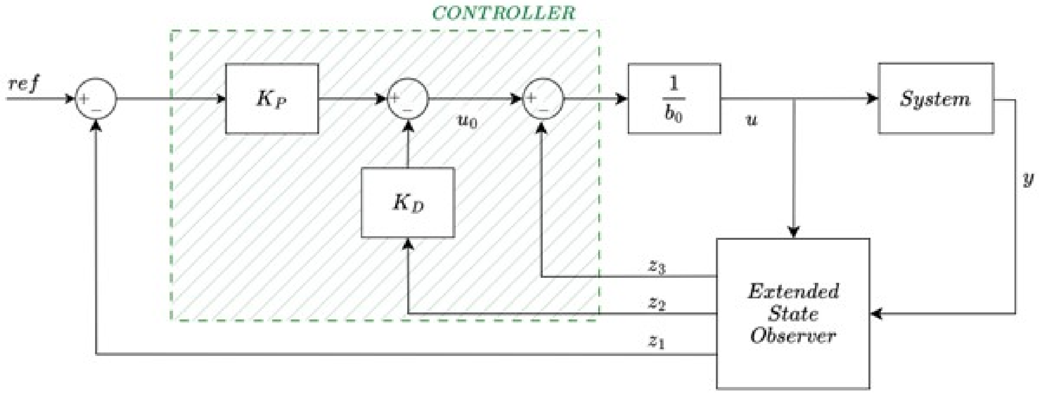


Figure 3.3: Second-order linear ADRC block diagram

3.2.2 Second-Order Nonlinear ADRC Controller

Given the highly nonlinear nature of the selected missile model, linear ADRC has proven insufficient to achieve the desired performance. Therefore, to enhance the stability and robustness of the system under abrupt uncertainties and disturbances, second-order nonlinear ADRC has been developed.

A typical second-order nonlinear ADRC comprises of three primary components: the tracking differentiator (TD), the extended state observer (ESO), and the nonlinear state error feedback (NLSEF).

The second-order nonlinear ADRC closed-loop system block diagram is depicted in Figure 3.4.

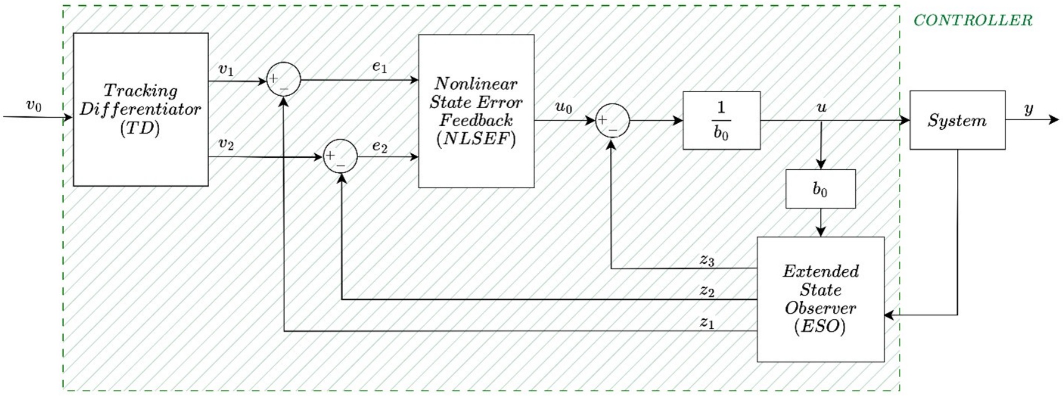

CONTROLLER
v_0
Tracking Differentiator (TD)
v_1
v_2
e_1
e_2
Nonlinear State Error Feedback (NLSEF)
u_0
$\frac{1}{b_0}$
u
b_0
System
y
Extended State Observer (ESO)
z_3
z_2
z_1

Figure 3.4: Second-order nonlinear ADRC block diagram

ADRC initially employs a TD to obtain the derivative signal of the input signal from the system's reference signal, enabling accurate and smooth tracking of both the reference signal and its derivative. TD part of ADRC is shown as:

$$\begin{aligned} e &= v_1(k) - v_0(k) \\ fh &= fhan(e, v_2(k), r_0, h_0) \\ v_1(k+1) &= v_1(k) + hv_2(k) \\ v_2(k+1) &= v_2(k) + hfh \end{aligned} \tag{3.2}$$

fhan(x_1, x_2, r_0, h_0), is a key function in ADRC for managing transition dynamics. v_0, v_1 and v_2 are the reference signal, tracking signal and the derivative of the tracking signal, respectively. h represents the sampling period, while r_0 and h_0 are two adjustable controller parameters. r_0, acting as the speed factor, enhances response speed and reduces the transition time. However, an excessively large value may cause the tracking signal to closely resemble the expected input signal, thereby nullifying the significance of the transition process. On the other hand, h_0 functions as a filter factor, enhancing filtration efficacy. Nevertheless, increasing h_0 can introduce a more substantial phase delay.

fhan(x_1, x_2, r_0, h_0) function is defined as:

$$
\begin{aligned}
d &= r_0 h_0^2 \\
a_0 &= h_0 x_2 \\
y &= x_1 + a_0 \\
a_1 &= \sqrt{d(d+8|y|)} \\
a_2 &= a_0 + \frac{\text{sign}(y)(a_1 - d)}{2} \\
s_y &= \frac{\text{sign}(y+d) - \text{sign}(y-d)}{2} \\
a &= (a_0 + y - a_2)\, s_y + a_2 \\
s_a &= \frac{\text{sign}(a+d) - \text{sign}(a-d)}{2} \\
fhan &= -r_0 \left[\frac{a}{d} - \text{sign}(a)\right] s_a - r_0\, \text{sign}(a)
\end{aligned}
\tag{3.3}
$$

The results of the *fhan*$(x_1,\ x_2,\ r_0,\ h_0)$ function for $r_0 = 5$, $h_0 = 0.001$ and $r_0 = 1$, $h_0 = 1$ are shown in Figures 3.5 and 3.6, respectively, for illustration purposes.

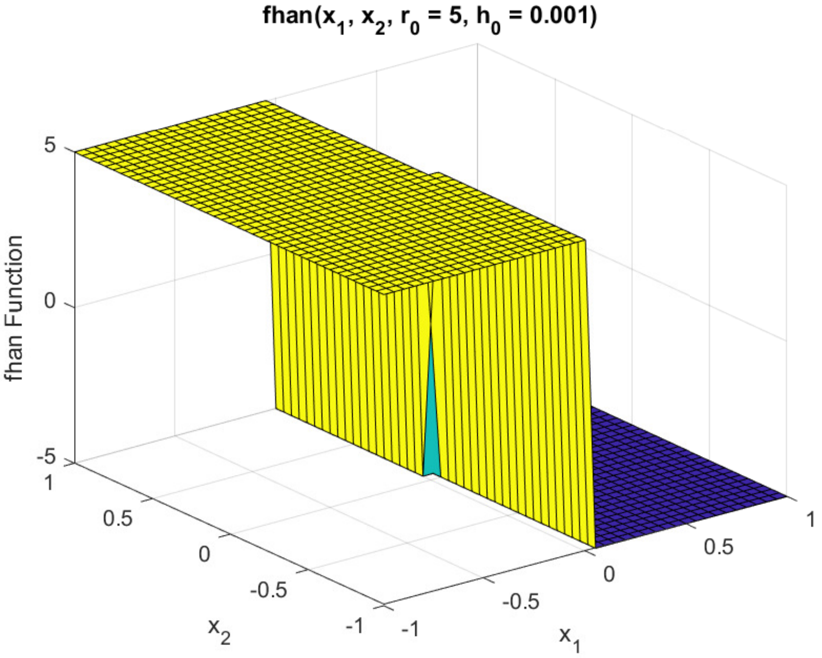


Figure 3.5: *fhan*$(x_1,\ x_2,\ r_0,\ h_0)$ function for $r_0 = 5$, and $h_0 = 0.001$

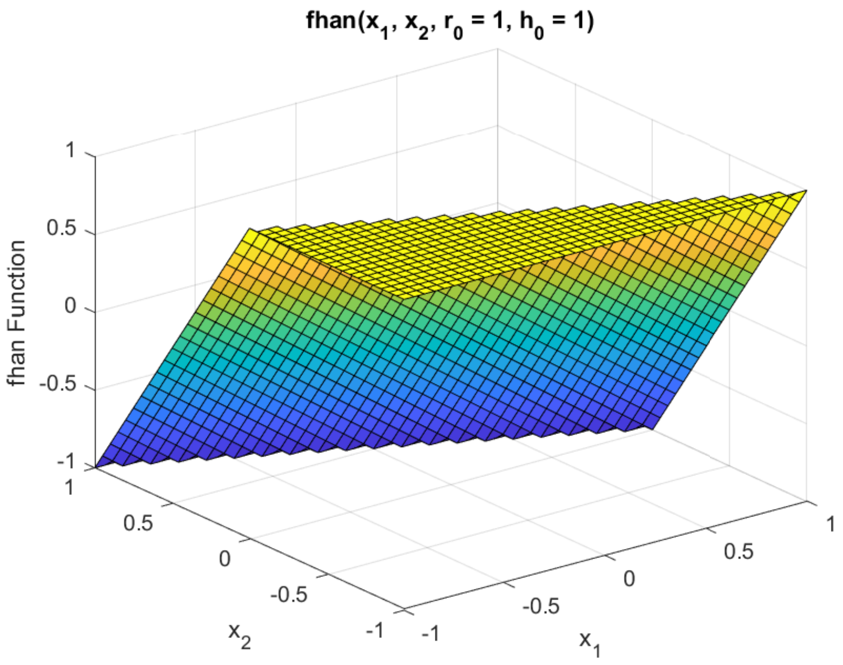


Figure 3.6: *fhan*$(x_1,\ x_2,\ r_0,\ h_0)$ function for $r_0 = 1$, and $h_0 = 1$

ESO, which extends the concept of state observation by including the system's total disturbance as an additional state for estimation. The ESO dynamically estimates both the system's internal states and the combined impact of internal dynamics and external disturbances. ESO part of ADRC algorithm is shown as:

$$\begin{aligned}
\epsilon &= z_1(k) - y(k) \\
fe_0 &= fal(\epsilon, \alpha_{01}, \delta) \\
fe_1 &= fal(\epsilon, \alpha_{02}, \delta) \\
z_1(k+1) &= z_1(k) + h(z_2(k) - \beta_{01}\epsilon) \\
z_2(k+1) &= z_2(k) + h(z_3(k) - \beta_{02} fe_0 + b_0 u) \\
z_3(k+1) &= z_3(k) + h(-\beta_{03} fe_1)
\end{aligned} \tag{3.4}$$

z_1, z_2 and z_3 represent the output signal, its derivative signal, and the total interference signal estimated by ESO, respectively.

β_{01}, β_{02} and β_{03} are generally proportional with h as:

$$\beta_{01} = \frac{k_1}{h}, \quad \beta_{02} = \frac{k_2}{h^2}, \quad \beta_{03} = \frac{k_3}{h^3} \tag{3.5}$$

$fal(x, \alpha, \delta)$ is a nonlinear function, introduces nonlinearity to ESO, and defined as:

$$fal(e, \alpha, \delta) = \begin{cases} \frac{e}{\delta^{1-\alpha}}, & \textit{if}\, |e| \leq \delta, \\ |e|^{\alpha}\,\mathrm{sign}(e), & \textit{if}\, |e| > \delta, \end{cases} \quad \delta > 0, \alpha > 0 \tag{3.6}$$

δ is linear interval width, and α is the function control parameter.

The results of the $fal(e,\ \alpha,\ \delta)$ function for $\delta = 0.1$ and $\delta = 0.5$ are shown in Figures 3.7 and 3.8, respectively, for illustration purposes.

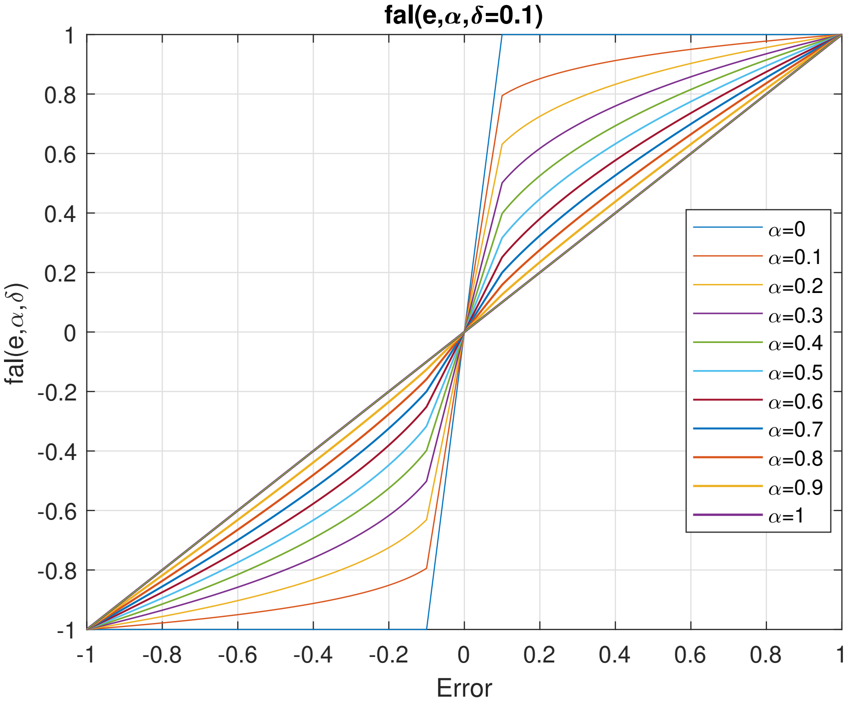


Figure 3.7: $fal(e,\ \alpha,\ \delta)$ function for $\delta = 0.1$

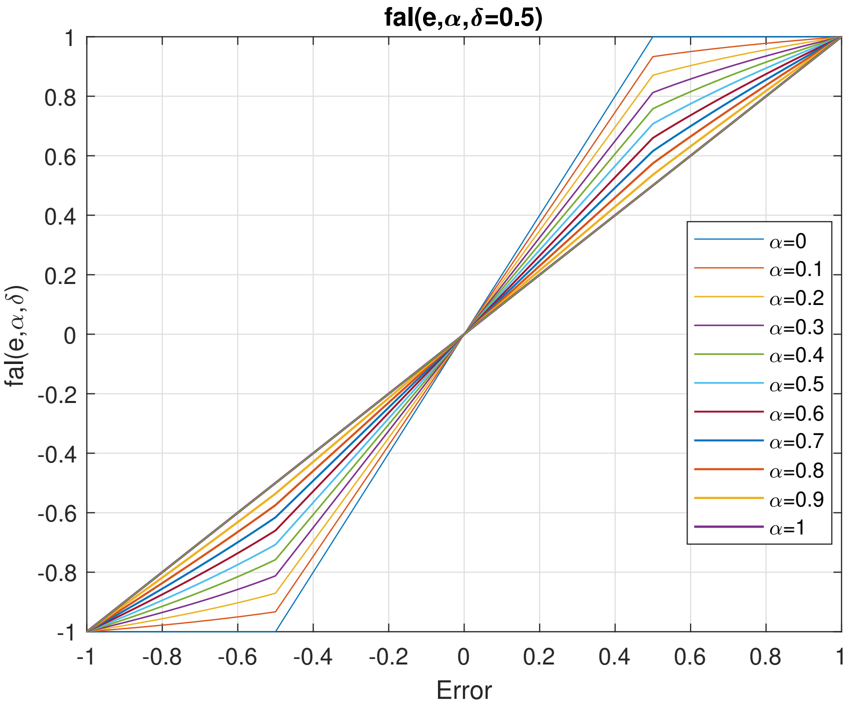


Figure 3.8: $fal(e,\ \alpha,\ \delta)$ function for $\delta = 0.5$

Using the state and disturbance estimates provided by the ESO, NLSEF part employs a feedback control law to calculate the required control inputs. The objective is to neutralize the estimated disturbances and steer the system toward the needed trajectory or state. NLSEF part of ADRC algorithm is shown as:

$$\begin{aligned} e_1 &= v_1(k) - z_1(k) \\ e_2 &= v_2(k) - z_2(k) \\ u_0 &= \beta_1 \mathit{fal}(e_1, \alpha_1, \delta) - \beta_2 \mathit{fal}(e_2, \alpha_2, \delta) \end{aligned} \tag{3.7}$$

The control input gain used in ADRC determines how quickly the system responds to the reference. These gains control the acceleration or deceleration of the system. This compensation equation is shown as:

$$u = \frac{u_0 - z_3(k)}{b_0} \tag{3.8}$$

As a control variable, u is divided into two parts:

Nonlinear feedback for controlling:

$$u = \frac{u_0}{b_0} \tag{3.9}$$

Compensation for disturbances:

$$u = \frac{z_3(k)}{b_0} \tag{3.10}$$

Second-order nonlinear ADRC tuning will be detailed in Section 3.3.2.

3.3 Controller Tuning

3.3.1 Tuning PID Controller

PID tuning is the process by which the optimal gains for a PID controller are determined to provide stable and responsive control of a dynamic system. Common methods such as the Ziegler-Nichols method and manual tuning are often used as initial steps to approximate these gains [22], [25].

Initially, a step input is applied to the system, and its response is observed. The proportional gain (K_p) is then increased until the system begins to oscillate, providing an initial value. Unfortunately, steady state error is never zero for a P-only controller.

Secondly, integral action is added in parallel with proportional action, and the integral gain (K_i) is adjusted to eliminate any steady state error. Integral action can introduce the risk of closed-loop instability. Excessive accumulation of integral error can lead to the process variable to overshoot the set point. Therefore, an anti-windup part is incorporated to maintain stability.

Finally, the derivative gain (K_d) is fine-tuned to dampen the oscillations. Careful consideration is given to the selection of K_d to avoid excessive overshoot and sensitivity to disturbances. Also, it helps to reduce settling time.

While manual tuning provides a reasonable starting point for PID gains, it may fail to reach the global optimum, as it can become limited by subjective judgment and system-specific behavior. Therefore, to further improve the tuning process, a two-step optimization approach is implemented:

Initially, the manually tuned PID gains were used as starting points, and a neighborhood region around these gains is explored using the Particle Swarm Optimization (PSO) algorithm. PSO is a population-based global optimization method that explores a wide search space to approximate the global optimum. In this study, a 20% neighborhood around the manually tuned parameters is selected as the initial search space. This percentage neighborhood ensures that the algorithm searches close to the manually tuned values while still having flexibility to explore other promising regions in the parameter space.

After the global search using PSO, the resulting PID parameters were further refined using a gradient-based constrained optimization method. This method fine-tuned the parameters by locally minimizing a defined cost function subject to constraints on the PID gains. The gradient-based method provides a more precise adjustment of the gains by leveraging local gradients to converge in on a local minimum, ensuring improved control performance. Constraints are imposed to prevent parameters from reaching unrealistic or unstable values, ensuring robust operation.

By combining both global and local optimization techniques which are given in detail in Section 3.3.3, this approach allows the controller to first explore the global solution space for potential optima and then perform precise local adjustments to fine-tune the parameters.

Once the PID parameters are chosen, the controller's performance is tested under various conditions to verify that it meets the desired specifications and provides robust control.

3.3.2 Tuning ADRC Controller

In a linear ADRC system, typically three main parameters need to be tuned. These are TD bandwidth, observer bandwidth and controller gain. Tuning the parameters of linear ADRC is a challenging task, as highlighted in [26], [27].

Furthermore, in a second-order nonlinear ADRC, the tuning process becomes even more complex due to the significantly larger number of parameters that must be tuned. Since there are too many parameters to be tuned, the starting point and the steps should be carefully chosen. Proper parameter tuning can significantly improve the performance and robustness of the autopilot system.

In this scope of thesis, the optimal tuning guide for the second-order nonlinear ADRC is proposed.

As given detailed formulation in Section 3.2, second-order nonlinear ADRC consists of fourteen parameters to be optimized which are given in Table 3.1. These parameters, significantly influence the controller's ability to reject disturbances, manage nonlinearities, and achieve precise control under varying operational conditions and disturbances.

Table 3.1: Second-order nonlinear ADRC parameters

Parameter	**Symbol**
TD speed factor	r_0
TD filtering factor	h_0
ESO gain coefficient-1	β_{01}
ESO gain coefficient-2	β_{02}
ESO gain coefficient-3	β_{03}
ESO linear interval width	δ_{ESO}
ESO function control parameter-1	α_{01}
ESO function control parameter-2	α_{02}
NLSEF gain coefficient-1	β_1
NLSEF gain coefficient-2	β_2
NLSEF linear interval width	δ_{NLSEF}
NLSEF function control parameter-1	α_1
NLSEF function control parameter-2	α_2
ADRC control input gain	b_0

3.3.2.1 Tuning TD Part

First of all, the TD part should be set. Since the parameter r_0 represents the tracking bandwidth, a relatively large value for this parameter should be chosen as a starting point. Then, change the value until a smooth reference is found. The greater the r_0, the faster the system response, but it also increases the overshoot. h_0 denotes the filtering factor. The bigger the h_0, the better the filtering effect. Typically, h_0 is chosen to be four to five times the step size. Since the considered system is a missile, step size is selected as 0.001 to capture the rapid dynamics accurately. However, for slower systems where the response time is less critical, larger step sizes can be used.

3.3.2.2 Tuning ESO Part

The second part is tune the ESO parameters. The main role of the ESO is to estimate the system states and disturbances. It has the observer gain parameters, the interval width and function control parameters. Firstly, observer gain coefficients should be chosen. These values are responsible for how fast the ESO responds. The higher the observer gain coefficients, the faster the estimation. However, this leads to the noise amplification. Therefore, it is crucial to find optimal observer coefficient values. Generally, β_{01}, β_{02}, and β_{03} are proportional to h_0, h_0^2, and h_0^3, respectively. Moreover, the relationship between the parameters follows $\beta_{01} < \beta_{02} < \beta_{03}$. However, determining the coefficients for these values based on the magnitude of the error is crucial for accurate results. If the observer coefficient values are selected too high, ESO will be very fast and responsive; however, excessively high gains can increase the system's sensitivity to noise, potentially resulting in a loss of control.

3.3.2.3 Tuning Function Control Parameters

Function control parameters, α, for both the ESO and NLSEF are selected within the range of 0 to 1. These parameters are used in the error feedback functions and play a critical role in shaping the slope of the nonlinear feedback functions, directly influencing the system's responsiveness and stability. In simulations, they are typically chosen to be greater than 0.5 to achieve the desired response characteristics.

3.3.2.4 Tuning Linear Interval Width

Linear interval width parameters, δ, are used for both the ESO and NLSEF. These define the linear range of $fal(x, \alpha, \delta)$ function and must be adjusted based on the magnitude of the error taking into account the type of controller, such as whether it is an angle control or angular velocity control.

3.3.2.5 Tuning NLSEF Part

The next part is to find the NLSEF gain coefficient values. This part is responsible for generating the control inputs based on the error values, state and disturbance estimates. Small values should be chosen as a starting point and the values should be gradually increased while taking into account the system's response. This part behaves somewhat similarly to proportional and derivative gains, and it has a significant direct impact on the output of the ADRC. Increasing β_1 reduces the system's static error and improves tracking accuracy, but it can also cause the system to oscillate. Therefore, careful attention must be given to its design and tuning to ensure optimal system performance.

3.3.2.6 Tuning Control Law Part

Lastly, the ADRC control input gain, b_0, must be tuned. This gain plays a crucial role in determining how the control signal is compensated by the ESO and how quickly the system responds to the reference. Therefore, careful attention should be given to this parameter value during all tuning steps.

3.3.3 Hybrid Optimization Approach for Controller Tuning

In this section, the structure of the controller parameter optimization algorithm is presented. The optimization process integrates the Particle Swarm Optimization (PSO) algorithm with a gradient-based refinement technique to achieve an optimal set of controller parameters.

The structure of controller parameter optimization algorithm is illustrated as in Figure 3.9.

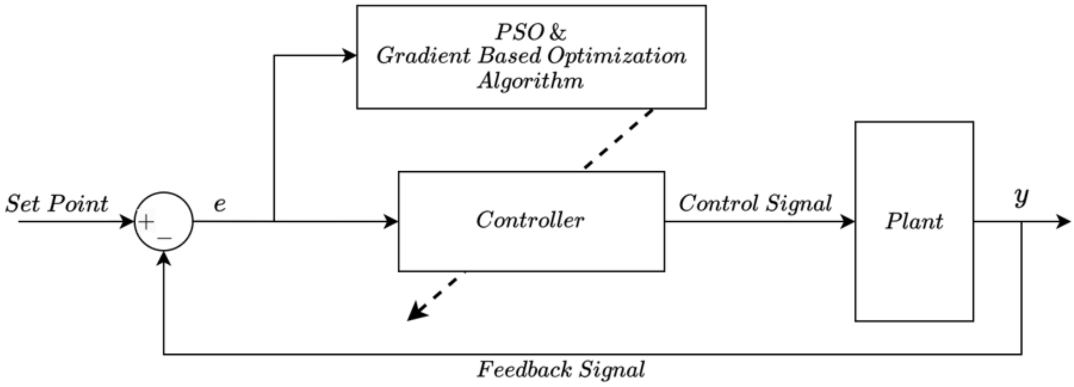


Figure 3.9: Structure of controller parameter optimization

3.3.3.1 Particle Swarm Optimization (PSO)

Particle Swarm Optimization (PSO) is an algorithm used to find the approximate solutions to optimization problems. The technique is inspired by the social behavior of birds or fish. In the PSO algorithm, a population of particles navigates through the search space, guided by its own best experience and the experience of other particles with the help of iterative updates [28].

PSO algorithm begins with the initialization of a predetermined number of particles, each representing a potential solution of the optimization problem. Each particle is characterized by its position and velocity. The fitness of each particle is calculated based on its position in this space, providing an indication of how well the particle's position solves the problem [29]. Afterwards, each particle is updated to its own personal best position, reflecting the best solution it has achieved thus far, and also the global best solution is updated which is the best solution with the highest fitness among all particles. The algorithm iteratively updates positions and velocities of the particles until a predetermined terminal condition is met. In this study, maximum iteration number is selected as the stopping criterion. The final value obtained is called as the global best position which represents the optimal solution to the optimization problem.

For fitness calculation, objective function is defined to evaluate the performance of a solution to the optimization problem. In the context of optimization algorithms like PSO, the objective function quantitatively measures how well a particular solution meets the requirements under consideration. The objective function generally takes the solution as input and returns a scalar value that represents the quality of that solution. In this study, the goal of PSO is to find solutions that minimize the value of the objective function[30]. After the controller parameters were determined using the Particle Swarm Optimization (PSO) algorithm, these parameters were subsequently refined through a gradient-based constrained optimization method to further improve system performance by minimizing a specified cost function, subject to constraints on controller gains.

There are generally four main performance indices used in control theory that assess system performance or model's response. These are referred to as Integral of Squared Error (ISE), Integral of Absolute Error (IAE), Integral of Time-Weighted Absolute Error (ITAE), and Integral of Time-Weighted Squared Error (ITSE), defined as follows:

$$\begin{aligned} \text{ISE} &= \int_0^{\infty} e^2(t)\,dt \\ \text{IAE} &= \int_0^{\infty} |e(t)|\,dt \\ \text{ITAE} &= \int_0^{\infty} t|e(t)|\,dt \\ \text{ITSE} &= \int_0^{\infty} te^2(t)\,dt \end{aligned} \tag{3.11}$$

In this context, t represents time, and $e(t)$ denotes the instantaneous error between the reference signal and the controlled variable at time t. The Integral of Time-Weighted Absolute Error (ITAE) algorithm has been selected among these indices to quantify the overall deviation of the system's response from the desired behavior across the entire time interval. To enhance the controller's anti-disturbance accuracy, the cost function incorporates both system error and overshoot, yielding the following cost function:

$$J = \int_0^\infty (a_1 t|e(t)|\, dt) + a_2\eta \tag{3.12}$$

where η represents overshoot, while a_1 and a_2 are the weighting parameters.

There are three important aspects for a particle i in an n-dimensional space. First, there is its current location, denoted as $x^i = \{x^i_1, \ldots, x^i_n\}$. Second, there is its historically best location, p^i, which is determined based on the evaluation of the cost function. Finally, there is the global best solution, p^g, within the population in m particles.

The velocity update is calculated using the formula below:

$$v^i_j(k+1) = c_1\, \text{rand}_1(p^i_j - x^i_j(k)) + c_2\, \text{rand}_2(p^g_j - x^i_j(k)) \tag{3.13}$$

where k is the iteration number, rand_1 and rand_2 are uniformly distributed random numbers $U(0,1)$, $c_1 > 0$ and $c_2 > 0$ are acceleration coefficients controlling the influence of a particle's historical best position and the global best location of the population, respectively. The first part of the right side of the equation is called "cognitive component" and the second part is called "social component". Then, the particle location is updated using the formula below:

$$x^i_j(k+1) = x^i_j(k) + v^i_j(k+1) \tag{3.14}$$

As the swarm approaches convergence, the personal and the global best solutions become closer to their current location, resulting in a decrease in velocity. This decrease in velocity also reduces the step size, indicating that PSO adaptively adjusts the step size.

3.3.3.2 Gradient-Based Constrained Optimization

The gradient-based optimization method is used to fine-tune the controller gains after the global search using PSO. This local optimization is performed using fmincon, a gradient-based solver available in MATLAB [21] designed to solve constrained nonlinear optimization problems.

In this phase, the ITAE algorithm is once again employed as the cost function for optimizing the controller parameters. The ITAE is selected for its ability to prioritize early error correction while effectively minimizing long-term system deviations. This metric not only reduces the overall error but also enhances system stability by penalizing persistent errors over time. Using the fmincon algorithm, a gradient-based constrained optimization method, the parameters are further refined in a local region around the initial values obtained from the PSO algorithm. This local optimization process improves the controller's performance by building upon the global search results from PSO, while ensuring that the final solution adheres to system performance criteria and stability constraints. Additionally, constraints on the coefficients are enforced to prevent the parameters from reaching unrealistic or unstable values, thereby resulting in a robust and precise tuning process that enhances overall control performance.

3.4 Autopilot Architectures

3.4.1 Attitude Autopilot Architecture

Given the nonlinear complexity of the missile model, a cascade control approach is employed, where multiple interconnected loops enhance overall system performance. These cascaded loops consist of an outer loop (primary) and an inner loop (secondary). The outer loop is responsible for regulating the primary process variable according to a predetermined setpoint or reference signal. Its primary role is to ensure that the process variable remains at the desired level, thereby optimizing the overall system performance. Meanwhile, the inner loop (or slave), representing the faster dynamics, responds swiftly to changes in the system, while the outer loop (or master), reflecting slower dynamics, provides broader system adjustments.

3.4.1.1 Two-Loop PID-PID Autopilot Architecture

A two-loop cascaded PID-PID attitude controller is designed and this controller structure will be referred to as PID^2. The inner loop is chosen to enhance the damping of the angular rate dynamics and to improve the performance of the outer angle loop. The cascade controller design begins with tuning the secondary controller. Subsequently, the primary controller is adjusted based on the closed-loop transfer function of the secondary loop, which is interconnected in series, as depicted in Figure 3.10. In the attitude controller block diagram illustration, the upper side represents the outer loop, while the second part corresponds to the inner loop.

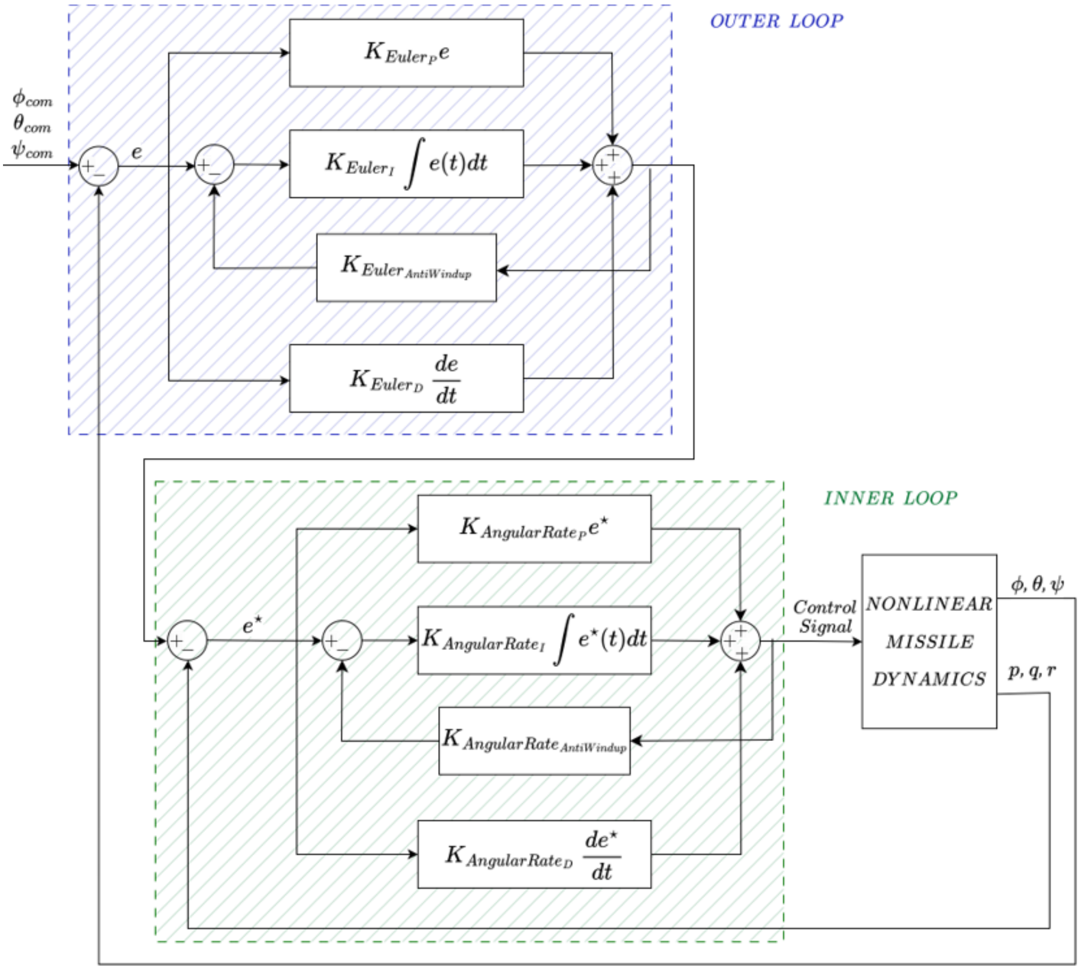

OUTER LOOP
ϕ_{com}
θ_{com}
ψ_{com}
e
$K_{Euler_P} e$
$K_{Euler_I} \int e(t)dt$
$K_{Euler_{AntiWindup}}$
$K_{Euler_D} \frac{de}{dt}$
INNER LOOP
$e^\star$
$K_{AngularRate_P} e^\star$
$K_{AngularRate_I} \int e^\star(t)dt$
$K_{AngularRate_{AntiWindup}}$
$K_{AngularRate_D} \frac{de^\star}{dt}$
Control Signal
NONLINEAR MISSILE DYNAMICS
ϕ, θ, ψ
p, q, r

Figure 3.10: PID^2 block diagram

PID^2 autopilot architecture block diagram is given in Figure 3.11. As shown in the figure, attitude is controlled by the outer loop, while angular velocities are included in the inner loop.

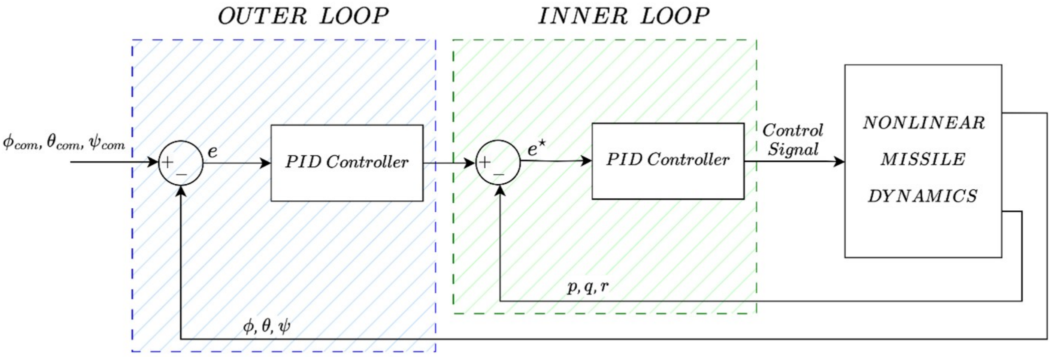


Figure 3.11: PID^2 autopilot architecture block diagram

3.4.1.2 Two-Loop ADRC-ADRC Autopilot Architecture

The two-loop ADRC-ADRC controller structure will be referred to as ADRC^2 and its block diagram is given in Figure 3.12. As depicted in the figure, attitude commands serve as reference inputs for the ADRC block. Additionally, system state variables are fed into the ADRC block.

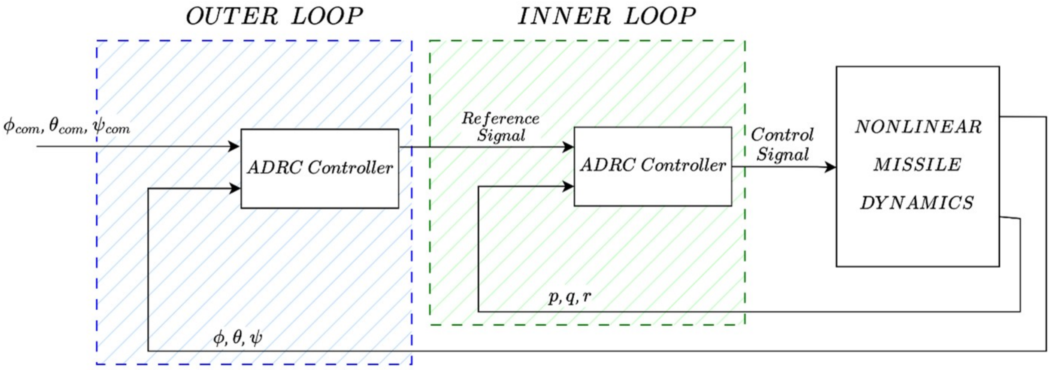


Figure 3.12: ADRC^2 autopilot architecture block diagram

3.4.1.3 Cascaded PID with Parallel ADRC (Outer Loop) Autopilot Architecture

Cascaded PID with parallel ADRC (outer loop) autopilot architecture will be referred to as ADRC || PID^2. The cascaded PID-PID controller provides a robust hierarchical control strategy, effectively handling both high-level and low-level system dynamics. By incorporating the parallel ADRC, the system gains enhanced disturbance rejection capabilities and improved adaptability to nonlinearities and uncertainties. This combination aims to leverage the strengths of both control methodologies, ensuring precise and reliable performance in complex and variable operating conditions. The weighting and switching between the cascaded PID and ADRC are managed using a weighting function, enabling smooth transitions and optimizing overall system behavior.

ADRC || PID^2 block diagram is given in Figure 3.13. This diagram illustrates the integration of the cascaded PID controllers with the parallel ADRC controller and the weighting and switching algorithm managed by a weighting function is also represented.

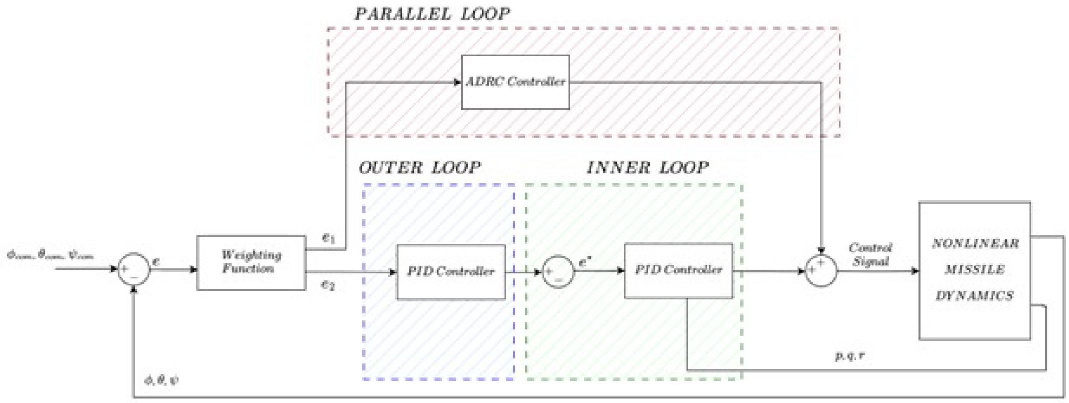


Figure 3.13: ADRC || PID^2 autopilot architecture block diagram

3.4.1.4 Cascaded PID with Parallel ADRC (Inner Loop) Autopilot Architecture

Cascaded PID with parallel ADRC (inner loop) autopilot architecture will be referred to as PID^2 || ADRC. The cascaded PID-PID controller provides a robust hierarchical control strategy, effectively handling both high-level and low-level system dynamics. By incorporating the parallel ADRC, the system gains enhanced disturbance rejection capabilities and improved adaptability to nonlinearities and uncertainties. This combination aims to leverage the strengths of both control methodologies, ensuring precise and reliable performance in complex and variable operating conditions. The weighting and switching between the cascaded PID and ADRC are managed using a weighting function, enabling smooth transitions and optimizing overall system behavior.

PID^2 ‖ ADRC block diagram is given in Figure 3.14. This diagram illustrates the integration of the cascaded PID controllers with the parallel ADRC controller and the weighting and switching algorithm managed by a weighting function is also represented.

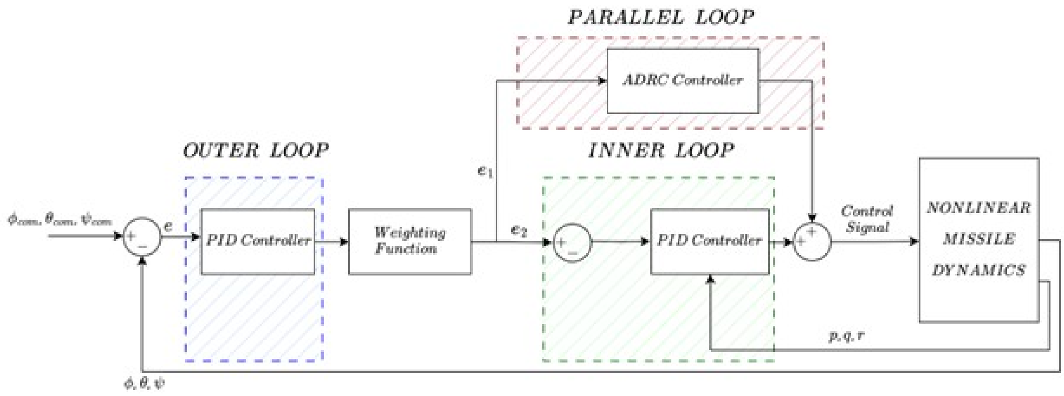


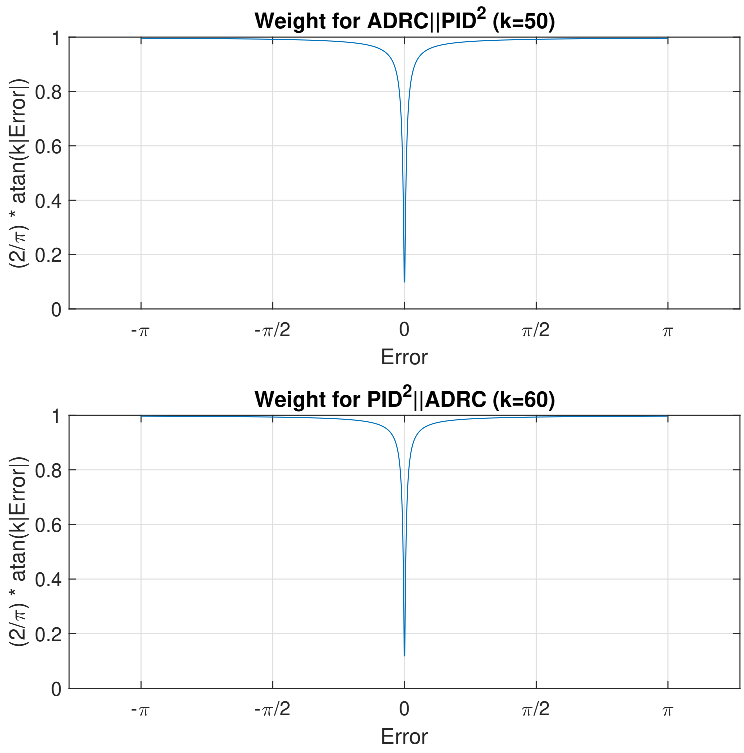
Figure 3.14: PID^2 ‖ ADRC autopilot architecture block diagram

3.4.1.5 Weighting Function

Scaled and modified arctan function is used for weighting and switching between ADRC and cascaded PID controllers.

$$\sigma(x) = \frac{2}{\pi} \arctan(k|x|) \tag{3.15}$$

x denotes the error value, while k represents the gain parameter for the weighting function. To scale the arctan function to the [0, 1] range $\frac{2}{\pi}$ term is added. Also, an appropriate value of k is needed so that when the error is at maximum expected value, the argument of arctan gives the output close to 1. For the maximum error value for the attitude π value is considered and the value of k is chosen as 50. For the angular rate error switching function, k value has been selected as 60. Weighting function weights versus error relation for ADRC || PID^2 and PID^2 || ADRC are given in Figure 3.15.


Weight for ADRC||PID^2 (k=50)

Weight for PID^2||ADRC (k=60)

Figure 3.15: Weighting function weights vs error for ADRC || PID^2 and PID^2 || ADRC

In the distribution of error to the controllers, the following equation is used; the error is weighted and transmitted to the controllers as e_1 and e_2:

$$e = \sigma(x)e_2 + (1 - \sigma(x))e_1 \tag{3.16}$$

$\sigma(x)$ smoothly transitions between 0 and 1 based on the value x and effectively switches the dominance between parallel controllers. However, during sudden small changes, there can be abrupt jumps between the ADRC ve PID weights. To prevent this issue, a smoothing function has been applied. Illustrating the transition between ADRC and PID coefficients is provided in Figure 3.16 as an example.

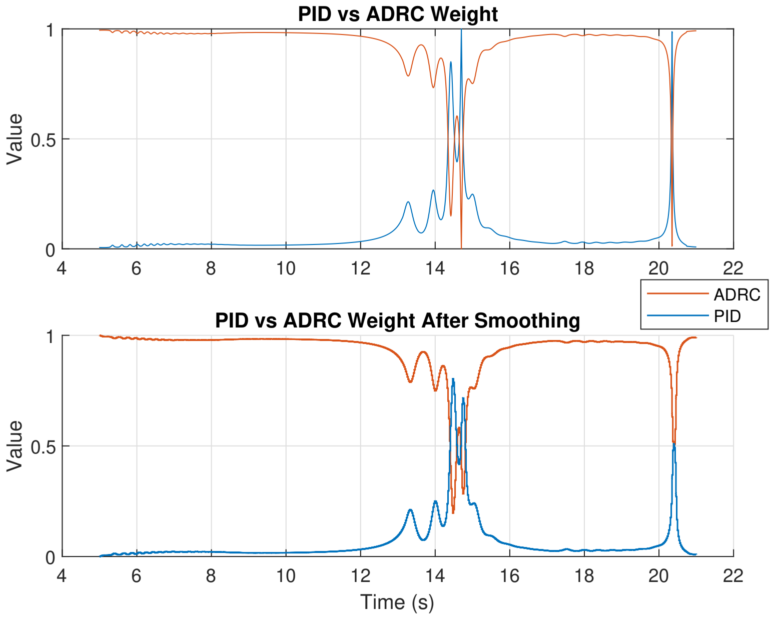


Figure 3.16: PID vs ADRC weights

3.4.2 Acceleration Autopilot Architecture

To efficiently track and engage a moving target, relying only on attitude autopilots is insufficient. Attitude autopilots ensure the missile's orientation, however, they might not be able to respond quickly enough to the dynamic behaviour of the target. Therefore, a more detailed control system is necessary, leading to the selection of a three loop autopilot system for the final autopilot architecture.

The innermost loop is responsible for controlling the missile's angular rate as stated before and middle loop controls the missile's attitude. Acceleration loop is selected as outer loop for the acceleration control. Its purpose is to control the missile's acceleration towards the target and enhance the responsiveness of the missile. This loop uses the commands from the missile's guidance algorithm which are generally acceleration based commands. The output from this loop feeds to the middle loop, providing the necessary inputs to adjust the missile's attitude to achieve the needed acceleration.

Three-loop autopilot architecture block diagram is given by Figure 3.17.

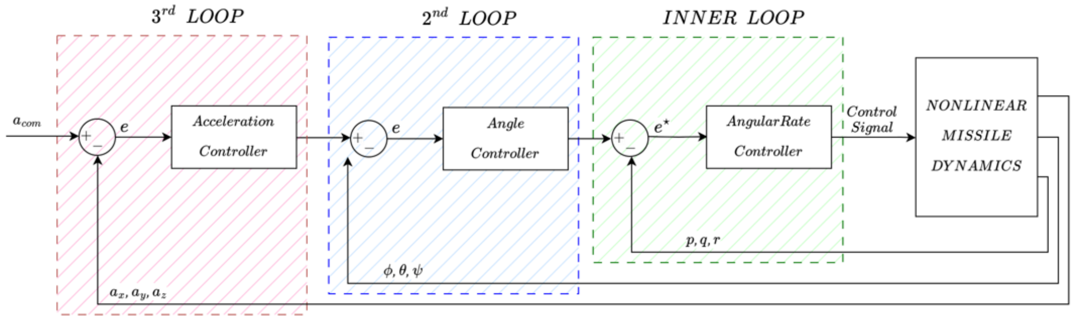


Figure 3.17: Three-loop autopilot architecture block diagram

3.5 Proportional Navigation Guidance (PNG)

For short to medium range tactical homing missiles, PNG is widely used for tracking and guiding missile towards the target. The core principle behind PNG is if two objects are closing on each other, they will inevitably intercept if the line of sight (LOS) between them remains stationary relative to an inertial reference frame [31], [32]. By maintaining a constant LOS, the missile continuously adjusts its trajectory, ensuring precise alignment with the target, even as both move dynamically. The seeker generally plays a critical role in accurately measuring the line-of-sight (LOS) rate. Moreover, a Doppler radar system often complements this by providing critical information on the closing velocity between the missile and its target. This combination of LOS rate and closing velocity data allows the missile's guidance system to make precise adjustments during flight, enhancing its ability to intercept fast-moving or maneuvering targets effectively.

PNG provides guidance based on acceleration commands. It uses the missile's closing velocity and the rate of change of the target's relative angular position. Mathematical formulation is given by (3.17)

$$a_n = VN\dot{\lambda} \tag{3.17}$$

N : Navigation constant (usually in the range of 3-5),

$\dot{\lambda}$: LOS rate,

a_n : Normal acceleration command,

V : Missile-target closing velocity.

In simulations, N is taken as 5.

The geometric framework for deriving PNG involves analyzing the relative motion between a missile and its target. For better visualization, the missile-target engagement geometry is illustrated in Figure 3.18.

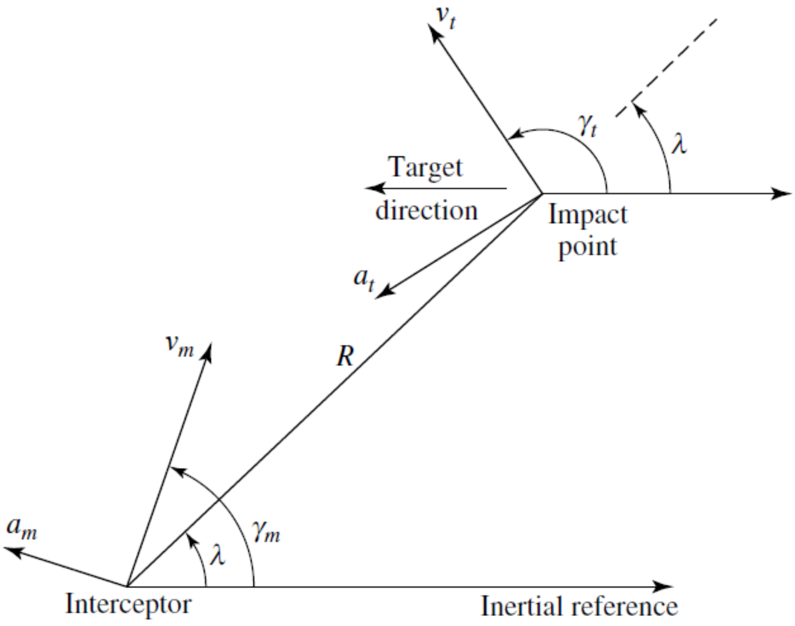


Figure 3.18: Missile-target engagement geometry [11]

R : Range between the missile and target,

v_m : Missile velocity,

v_t : Target velocity,

λ : LOS angle,

γ_m : Missile flight path (or heading) angle,

γ_t : Target flight path angle.

3.6 Energy and Power Analysis

For the comparison of modeled autopilots, an energy and power analysis are conducted. Since some controller architectures respond more rapidly than others, it becomes essential to assess their energy consumption in relation to these faster dynamics. To do this, current, power, and energy consumption are calculated throughout the simulations. Faster response times, while beneficial for control performance, often lead to higher energy demands due to the increased actuation efforts required by the control surfaces. These energy requirements directly impact the battery capacity. By comparing the energy profiles of the different autopilot architectures, the analysis provides insights into battery requirements, highlighting potential trade-offs between response speed and power efficiency.

To calculate power and energy consumption in electrical circuits, particularly in the context of the control actuation systems, the following fundamental equations are used:

$$P = IV = I^2R \tag{3.18}$$

P : Power,

I : Current,

V : Voltage,

R : Resistance.

$$E = Pt = IVt = I^2Rt \tag{3.19}$$

E : Energy,

t : Time.

Additionally, the potential and kinetic energy of the missile are computed. The kinetic energy is derived using the missile velocity as:

$$E_k = \frac{1}{2} m v^2 \tag{3.20}$$

E_k : Kinetic energy,

m : Mass,

v : Velocity.

The potential energy is calculated based on the missile's altitude using below formula:

$$E_p = mgh \tag{3.21}$$

E_p : Potential energy,

g : Acceleration due to gravity,

h : Height above a reference point.

To account for energy losses due to aerodynamic drag, the drag force is also computed. This force contributes to the overall energy dissipation of the missile, particularly during high-speed flight. The energy loss resulting from drag can be calculated using the following formula:

$$F_{\text{drag}} = \frac{1}{2} C_d \rho A v^2 \tag{3.22}$$

F_{drag} : Drag force,

C_d : Drag coefficient (a dimensionless number),

ρ : Fluid density,

A : Reference area (cross-sectional area),

v : Velocity of the body relative to the fluid.

Drag energy loss can be calculated using the drag force and the distance over which the force is applied.

$$E_{\text{drag}} = \int F_{\text{drag}}\, dx = \int \frac{1}{2} C_d \rho A v^2\, dx \tag{3.23}$$

E_{drag} : Drag energy,

dx : Infinitesimal distance element.

CHAPTER 4

SIMULATION RESULTS

This chapter presents a comprehensive comparison of various autopilot designs that utilize different controller architectures. The comparative analysis is conducted through nonlinear simulations, allowing for an in-depth evaluation of the controllers' performance across several criteria. Initially, different attitude autopilot architectures are assessed based on their control performance, settling time, and energy consumption.

Following autopilot performance evaluation, two out of the four initial architectures are selected for further examination. These selected architectures are then incorporated into a three-loop autopilot structure, and their performance is compared using PNG algorithm. In addition, the effects of internal and external disturbances on the system are thoroughly investigated.

For the simulation studies, the Runge-Kutta 4 numerical solver is employed with a step size of 1 ms. This small step size is adequate to capture the complex dynamics of the missile, ensuring accuracy in the simulations.

The atmospheric conditions, gravitational forces, inertial measurement unit and the armature-controlled DC servomotor system as a control actuator system are modeled analytically, providing a realistic representation of the missile and environment in which the missile operates. Meanwhile, the missile's aerodynamic properties are represented through a series of coefficients, which are stored in look-up tables. Linear interpolation is used to estimate the aerodynamic coefficients during the simulation, allowing for a smooth transition between the data points. This modeling approach ensures a precise and efficient simulation of the missile's behavior under various flight conditions.

In the simulation results, power and energy comparisons are also conducted to assess the efficiency and performance of the different autopilot designs. These calculations are fundamental in comparing the energy efficiency of the different autopilot architectures, offering insights into the trade-offs between control performance and power consumption.

4.1 Attitude Autopilot Simulation Results

Surface-to-air missiles typically need to ascend to high altitudes and adjust their direction towards the target. The pitch channel is responsible for raising and lowering the missile's nose. It must remain stable against atmospheric conditions, gravitational forces, and aerodynamic instabilities, and it needs to counteract gravity to maintain the desired trajectory. As the missile accelerates, aerodynamic loads on the pitch channel increase. This makes it difficult to adjust the missile's nose to the desired trajectory, especially at higher Mach number values. To reach the target, the margin for error in the pitch angle is usually very small. Thus, control errors in the pitch channel can cause significant trajectory deviations. Therefore, especially pitch control is of particular importance and pitch commands are considered when comparing various autopilot architectures.

As given the detailed architectures in Chapter 3, PID^2, ADRC^2, ADRC || PID^2 and PID^2 || ADRC architectures are compared. These architectures are analysed based on key performance metrics, such as control accuracy, response time, energy consumption, and robustness to external disturbances.

4.1.1 Attitude Autopilot Simulation Results with Step Input

Controllers are activated 0.1 s after launch, and the missile is accelerated for 8 s. The thrust profile is chosen as a rectangle, meaning it remains constant at 12500 N throughout the 8 s period.

Since the missile is initially fired vertically, the pitch angle is $90°$. In order to compare the responses of the controllers to the abrupt commands, a command is sent to the pitch angle to $0°$ 5 s after the launch. This represents a typical step command and the illustration of this maneuver is given in the 4.1.

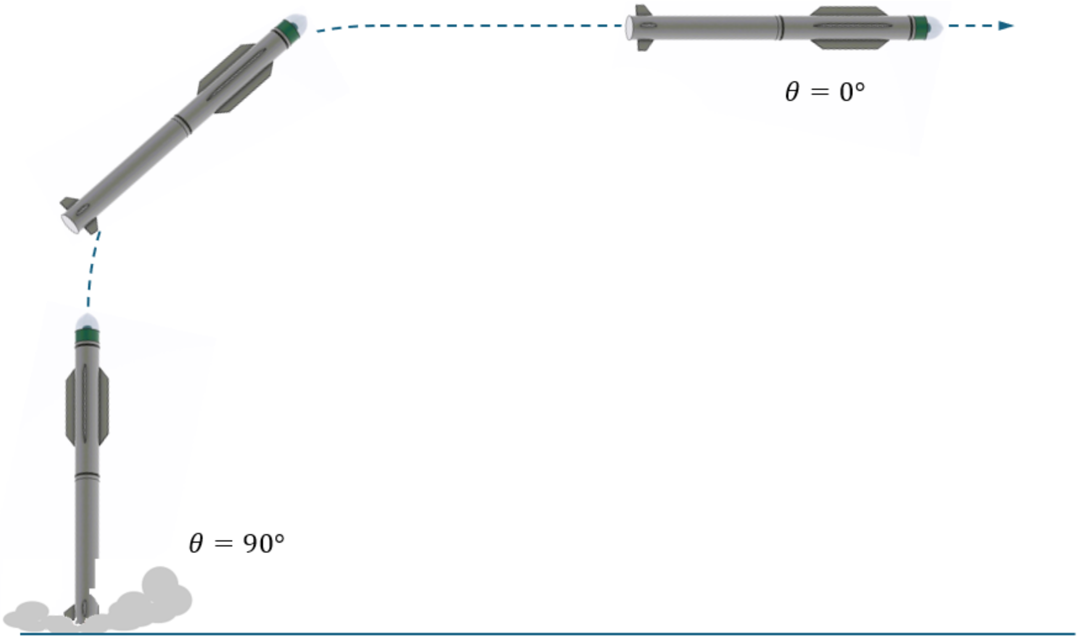


Figure 4.1: A representative launch profile transitioning into horizontal maneuvering

The responses of the four controller architectures are given in Figure 4.2. As can be seen from the figure, ADRC^2 and PID^2 || ADRC provide the fastest response. When the performance criterion is defined as the point where the error falls within a $0.5°$ range, ADRC^2 and PID^2 || ADRC meet this criterion at approximately 4.5 s, while ADRC || PID^2 reaches it in 5.84 s, and PID^2 in 6.8 s. This shows that the command response of the architectural structures using ADRC in the inner loop is 51% faster. This is because the inner loop has faster dynamics and the fast response ability of the ADRC structure is more suitable for the inner loop. However, as can be seen from the zoomed-in graph, the oscillations in controllers using ADRC in the inner loop are higher compared to the others. Moreover, as observed, there is no overshoot in any of the four controller architectures.

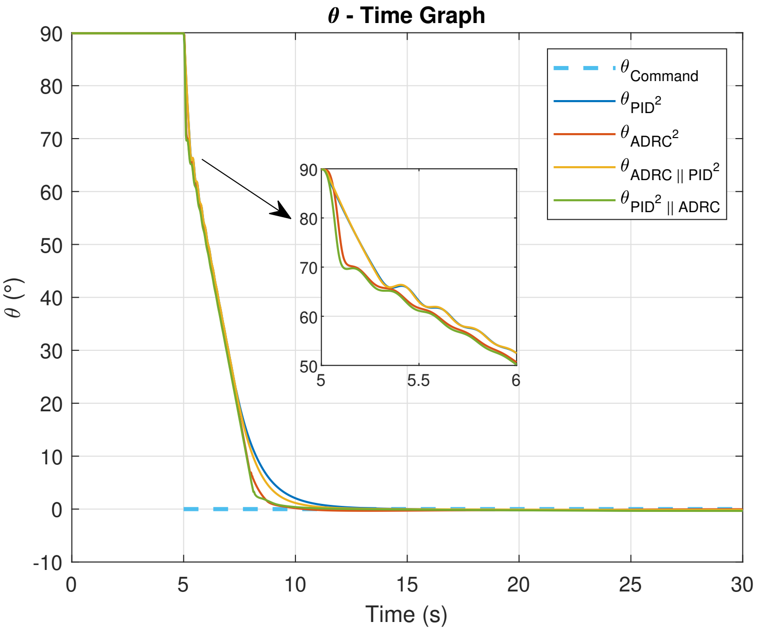


Figure 4.2: θ - time graph

The pitch angle command of 0° means that the missile will become parallel to the ground. As shown in Figure 4.3, the expected altitude change over time is achieved most rapidly in configurations where ADRC is used in the inner loop. Although faster responses are positive in terms of control performance; it should be taken into account that fast responses will cause an increase in the flight load on the missile.

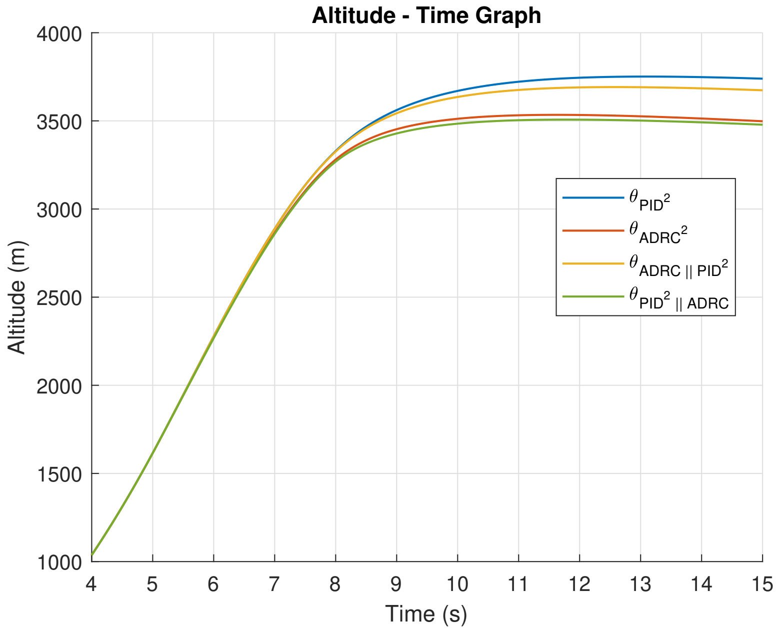


Figure 4.3: Altitude - time graph

The fast responses of the controller structures are as important as the difficulty of the commands they give to the control drive system and the energy they consume. For this purpose, the current, power and energy consumption of the control actuation system in the four controller architectures are examined.

As indicated in Figure 4.4, power consumption is higher in architectures where ADRC is used in the inner loop. It should be taken into account that as the energy requirement increases, the battery requirement will increase and this will lead to additional weight.

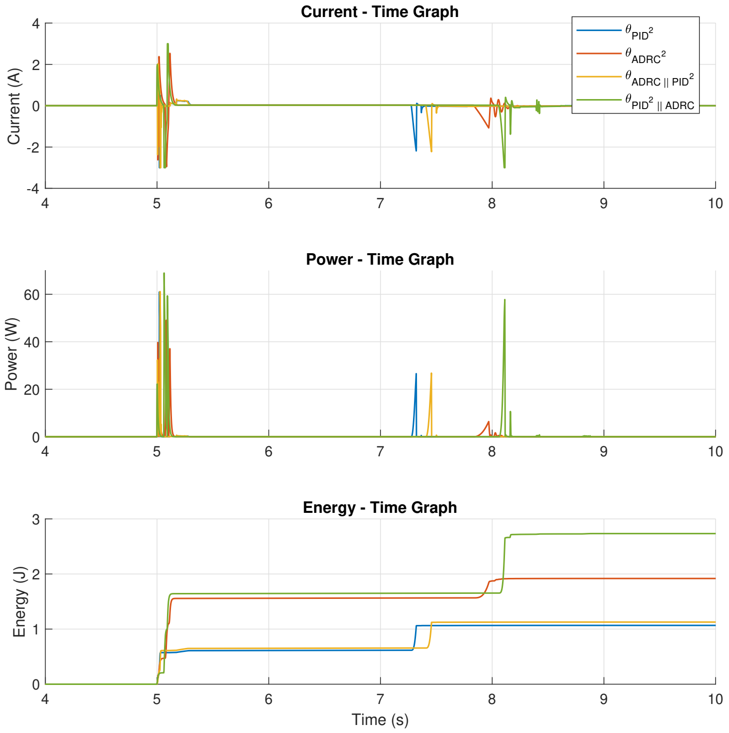


Figure 4.4: Current, power, and energy - time graph

The time-dependent graph of drag-induced energy consumption due to aerodynamic effects is presented in Figure 4.5. In this figure, it can be seen that architectures using ADRC in the inner loop lose more energy due to drag. The more aggressive response of ADRC in the inner loop and the faster response ability of the system and also energy losses are increased because of the more drag.

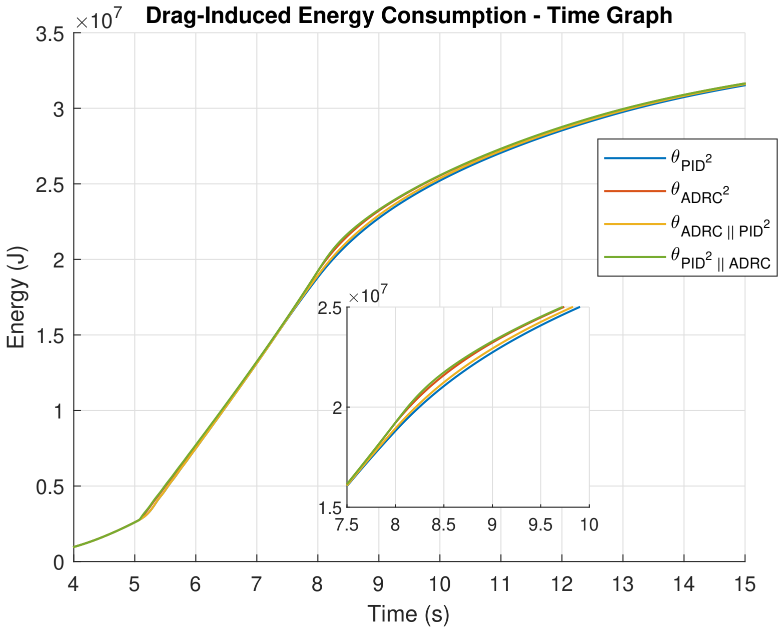


Figure 4.5: Drag-induced energy consumption - time graph

The instantaneous kinetic energy, potential energy and total energy graph of the missile is given in Figure 4.6. When the kinetic and potential energy of the missile are examined totally, it can be seen that the total energy of the architectures using ADRC in the inner loop is less than the others. In the PID^2 structure, the kinetic energy is 4%, the potential energy is 5.5% and the total energy is 4.3% more than the ADRC^2 structure. The main reason for this difference is the decrease in kinetic energy caused by drags in the architectures using ADRC in the inner loop and the earlier decrease in altitude and potential energy loss due to faster response to the commands. When the PID^2 and ADRC || PID^2 and ADRC^2 and PID^2 || ADRC architectures are compared among themselves, it is observed that the difference in energy consumption is below 1%.

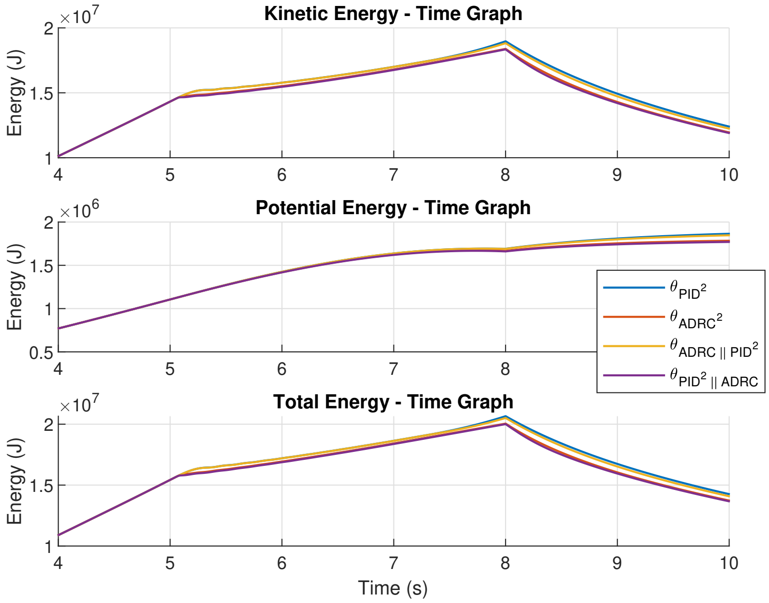


Figure 4.6: Kinetic, potential, and total energy - time graph

4.1.2 Attitude Autopilot Simulation Results with Ramp Input Under Disturbances

In order to examine the disturbance effects, thrust uncertainties and gust models are added to the 6-DOF simulation model.

In solid fuel engines, uncertainties such as fuel distribution, combustion characteristics, and pressure fluctuations cause thrust uncertainties. Since controllers are generally designed considering the nominal thrust profile, it is necessary to examine the performance of the controllers against these uncertainties. Therefore, the uncertainties are modeled and the thrust profile is updated. For this purpose, a normally distributed white noise, a sine wave profile with a frequency of 25 Hz, and a half-sine wave with a period of 1 s starting at 5.5 s are added. With this model, the total thrust value is increased to 103420 Ns; this corresponds to 3% increase in the total impact. The nominal thrust profile and the thrust profile with the addition of the disturbances are given in Figure 4.7.

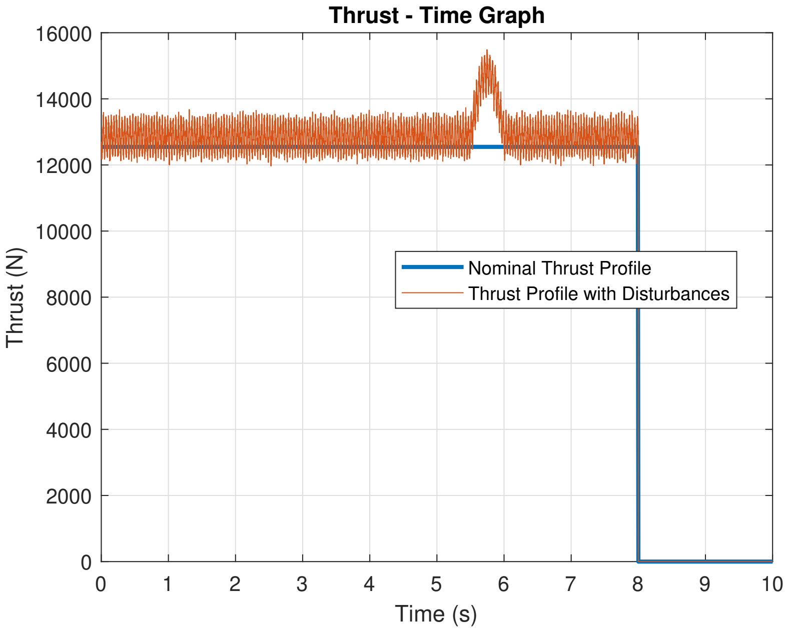


Figure 4.7: Thrust profile with disturbances

As the gust, the force corresponding to the half-sine wave profile with an amplitude of 8 m/s starting at 7 s and lasting for 3 s is added to the aerodynamic force block.

In order to compare the responses of the controllers to the different commands, a command similar to the ramp signal is send to the pitch angle. The ramp signal changes from 90° to 0° in 3 s and then holds at this value.

Figure 4.8 shows the responses of different autopilot architectures. When the given command is considered the error to be within the 0.5° range as a performance criterion, it is seen that the fastest response is achieved with the ADRC^2 architecture. The PID^2 || ADRC autopilot follows 0.48 s after ADRC^2; while ADRC || PID^2 enters the specified range after 1.15 s later, and PID^2 after 3.38 s. This situation shows that architectures using ADRC respond faster to commands and that especially architectures with ADRC in the inner loop respond faster than those with ADRC in the outer loop. This is because the faster response ability of ADRC can be used more effectively in the inner loop. However, as can be seen from the zoomed-in graph similar to the step input response, the oscillations in controllers using ADRC in the inner loop are higher compared to the others. Furthermore, its rapid response in the initial seconds is so fast that it overtakes the command itself.

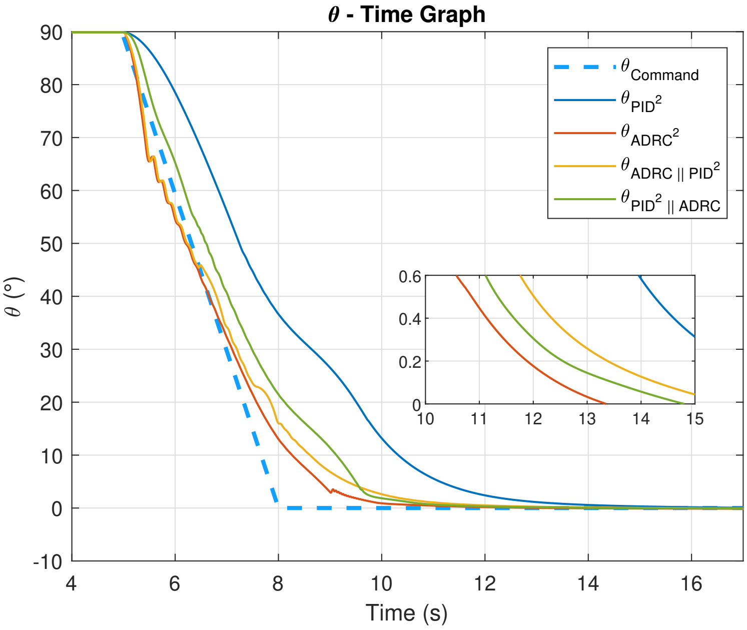


Figure 4.8: Inertial Euler angles - time graph under the effect of disturbance

The pitch angle command given in Figure 4.8 indicates that the missile is parallel to the ground. As shown in Figure 4.9, the fastest response in time-dependent altitude change is obtained with configurations where ADRC is used in the inner loop. Although faster responses are positive in terms of control performance, it should be kept in mind that these fast responses can increase the flight load on the missile.

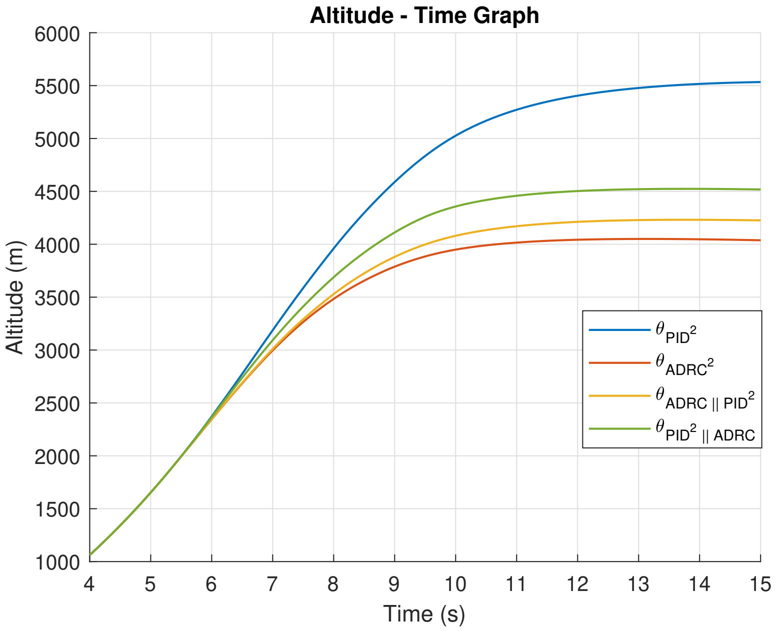


Figure 4.9: Altitude - time graph under the effect of disturbance

In addition to the fast response times of controller structures, the difficulty level and energy consumption of the commands they give to the control drive system are also of great importance. In this context, the current, power and energy consumptions in the control drive system of four different controller architectures are examined. As shown in Figure 4.10, the lowest power consumption is seen in PID^2 and PID^2 || ADRC architectures.

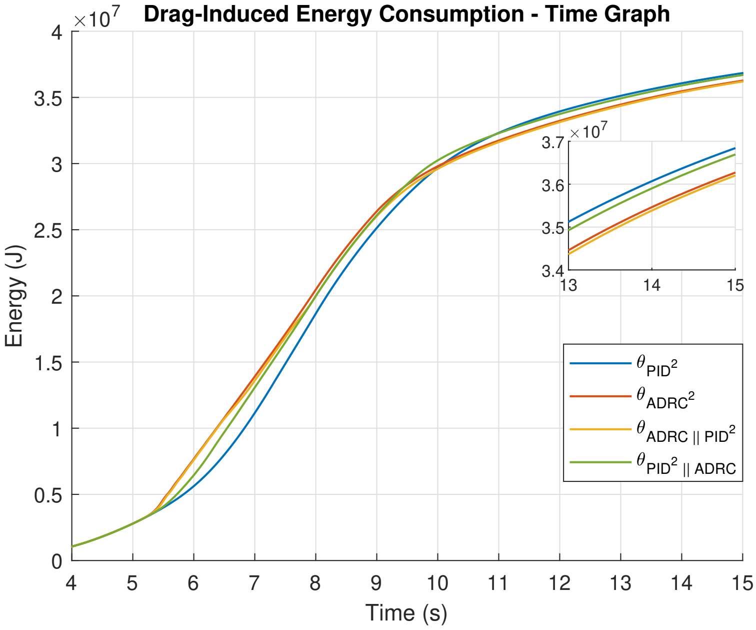


Figure 4.10: Drag-induced energy consumption - time graph under the effect of disturbance

The change in energy consumption over time due to drag caused by aerodynamic effects is shown in Figure 4.11. In this graph, it is observed that architectures using ADRC consume more energy due to drag effect while executing the command. In addition, when the total energy consumption due to drag is examined, it is seen that the highest energy consumption occurs in the PID^2 structure.

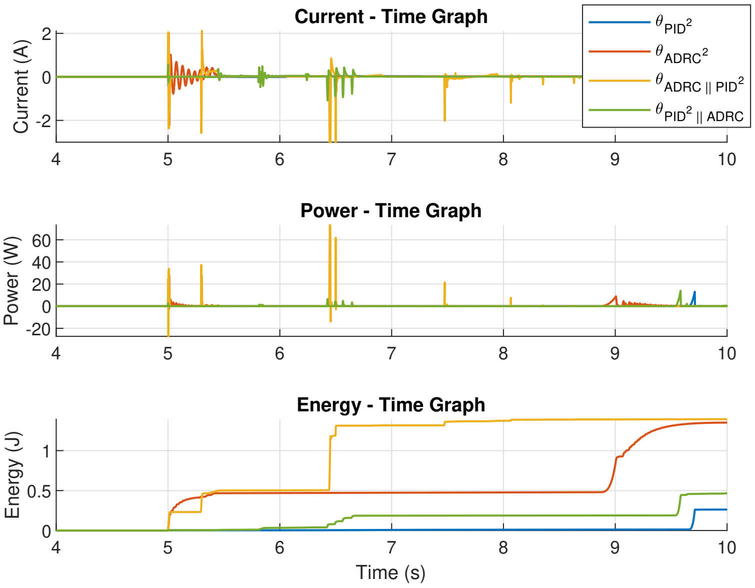


Figure 4.11: Current, power and energy - time graph under the effect of disturbance

When the kinetic and potential energies of the missile are examined collectively, it is seen that the energies of architectures using ADRC are lower than the architecture using only PID. In the PID^2 structure, the kinetic energy is 7.7%, the potential energy is 37% and the total energy is 30% more compared to the ADRC^2 structure. The reason for this is that structures using ADRC reduce the kinetic energy caused by drag more effectively. Additionally, the faster response to the given command leads to an earlier reduction in altitude, resulting in a loss of potential energy. The total energy difference between architectures with ADRC remains below 10%.

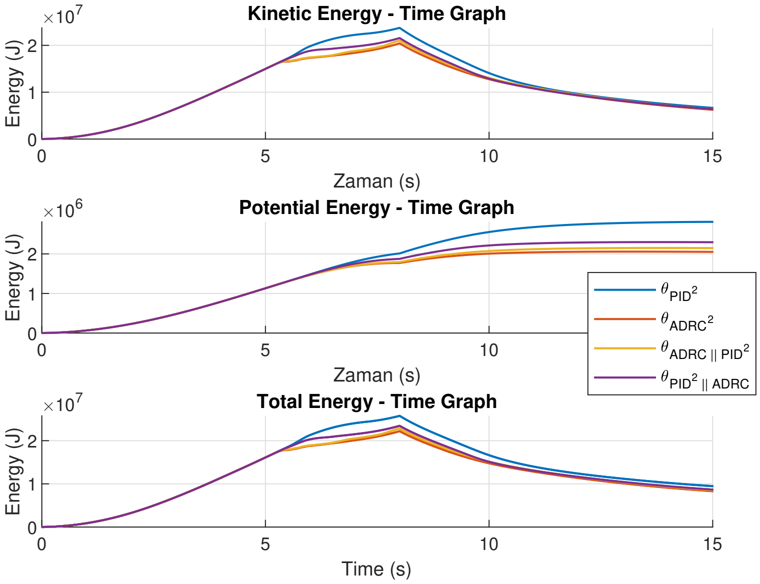


Figure 4.12: Kinetic, potential and total energy - time graph under the effect of disturbance

4.2 PNG Simulation Results

Following the analysis of the cascaded two-loop attitude autopilot architectures, the guidance performance of the controllers is also analyzed. In this part, a three-loop autopilot system is employed, where the outer loop is governed by an acceleration controller, while the inner loops consist of cascaded attitude autopilot structures. This configuration is particularly suited for missile control as the PNG commands generate acceleration demands, requiring a robust system that can translate these commands into precise trajectory adjustments. By incorporating the acceleration control in the outer loop, the system could respond to acceleration guidance commands. This configuration allows for a comprehensive evaluation of the controllers' ability to manage both attitude stabilization and dynamic flight path corrections under varying disturbances and different target maneuvers.

As discussed in the previous section, architectures that use ADRC as the angular rate controller in the innermost loop exhibit overly aggressive behavior, leading to significantly higher energy consumption. Although missile loads are not considered in this analysis, in real-world scenarios, this is a critical factor that must be carefully addressed, as excessive energy demands can impact missile performance and system longevity. Moreover, excessively rapid maneuvers can result in structural damage or failure of the missile under high flight loads. To mitigate these risks and focus on a more balanced evaluation of guidance performance, PID^2 and PID^2 ∥ ADRC architectures have been selected for comparison.

In this section, a low-pass filter is introduced to further improve system performance. Since the angular rates and accelerations contain various frequency components, a fourth-order Chebyshev Type II IIR low-pass filter is employed to mitigate the effects of high-frequency disturbances. This type of filter is selected for its effectiveness in noise reduction and its ability to maintain signal stability within the desired frequency range. The Chebyshev Type II filter is characterized by ripples in the stop-band but a completely flat response in the pass-band. The filter is designed using MATLAB [21] filter designer tool. Since delay is a critical factor to consider, both the stop-band frequency (Fstop) and the delay are taken into account when selecting the filter parameters. The flat pass-band response, as well as the magnitude and phase characteristics of the Chebyshev Type II filter, are shown in Figure 4.13.

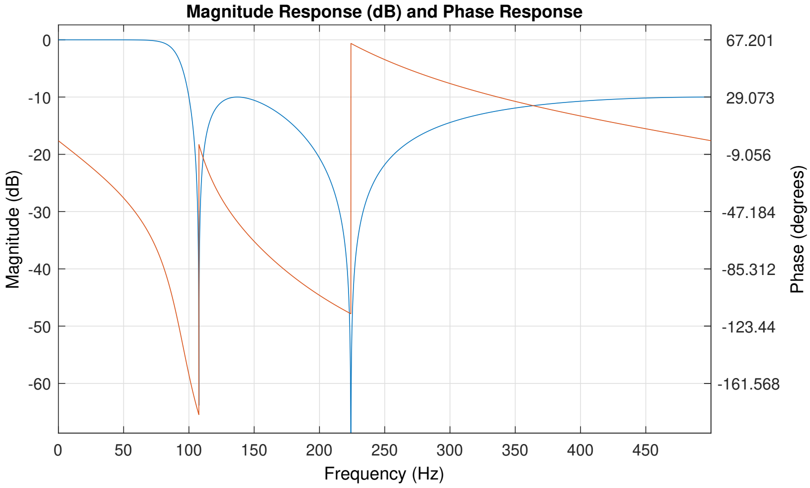


Figure 4.13: Magnitude and phase response of the filter

As can be seen in Figure 4.13, 10 dB attenuation begins at frequencies near 100 Hz and beyond which means the reduction of the amplitude by a factor of approximately 3. This effectively reduces high-frequency disturbances and components while maintaining stability in the desired frequency range with minimal delay. Group delay of the filter is illustrated in Figure 4.14. In this plot, it can be observed that the group delay peaks at around 100 Hz, reaching a maximum of approximately 12 ms, before gradually decreasing as the frequency increases. At 0 Hz, the delay is below 2 ms.

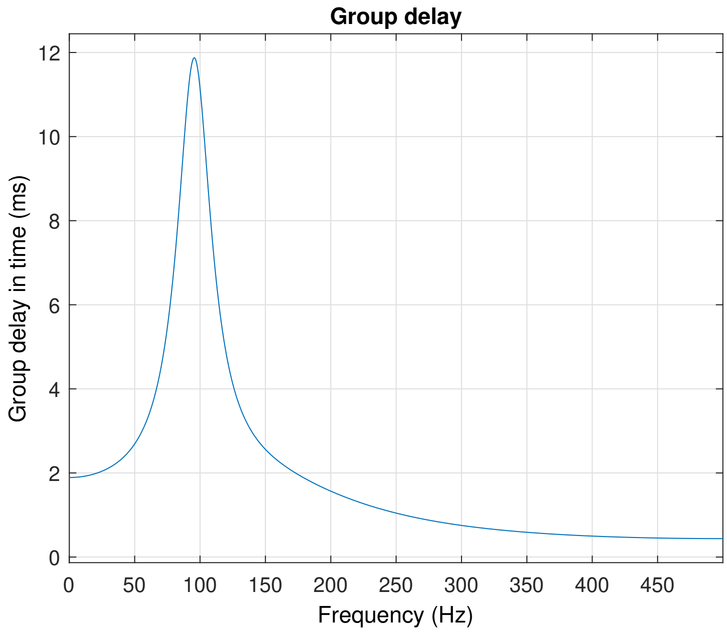


Figure 4.14: Group delay of the filter

4.2.1 PNG Simulation Results with Target-1

A representative target profile is provided in Figure 4.15. As shown, the missile is initially launched vertically and then follows a curved trajectory towards the target.

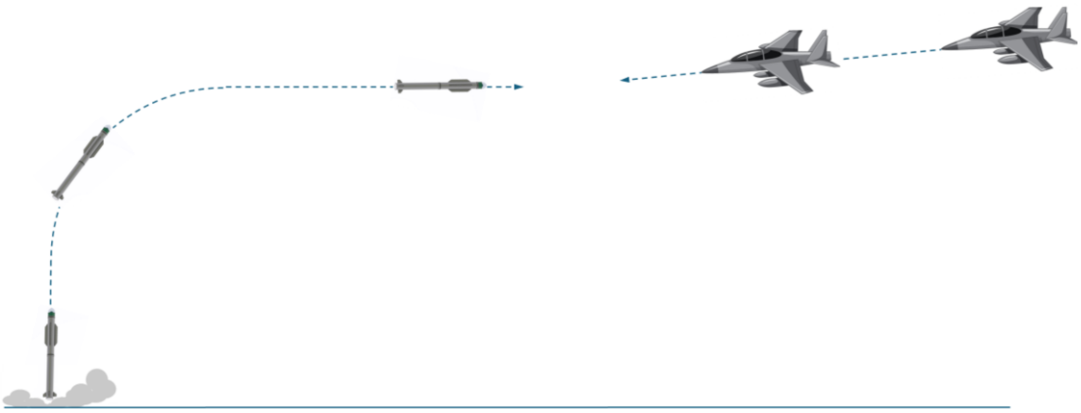


Figure 4.15: A representative Target-1 profile

Target-1 initial conditions are given in Table 4.1.

Table 4.1: Target-1 initial conditions

	Unit	Target-1
Initial Position	m	[6000 8000 0]
Initial Velocity	m/s	[200 0 0]
Initial Acceleration	m/s^2	[20 20 0]

Target-1 acceleration command and response time graph changing with time graph is given in Figure 4.16. The figure illustrates the acceleration command and response time graphs for the Target-1 scenario, comparing two controller architectures: PID^2 and ADRC || PID^2. Both the PID^2 and ADRC || PID^2 graphs, the acceleration command exhibits a relatively smooth response initially, but near 12 s, there is a significant spike in the response. Despite the small differences in behavior between the two controllers, both are able to successfully intercept the target with a miss distance of less than 1 meter. This is particularly noteworthy given that in PNG guidance, the system often requires rapid maneuvers during the final moments of engagement. Although the ADRC || PID^2 controller issues more aggressive commands, especially in the first seconds, it still manages to guide the missile to a successful intercept. Both architectures enable the system to handle these abrupt, last-second maneuvers effectively, ensuring a successful hit on the target, despite the increased dynamic demands.

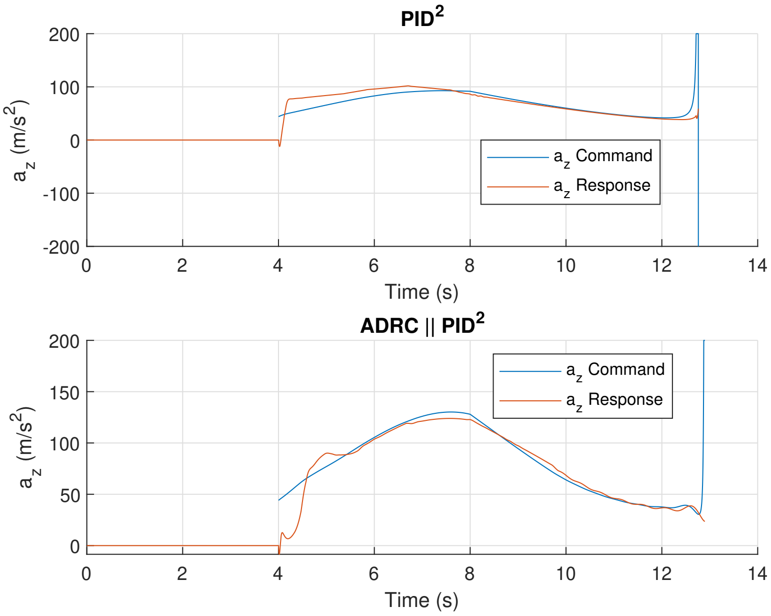


Figure 4.16: Target-1 acceleration command and response time graph

Target-1 interception results are given in Figure 4.17. The figure presents illustrating the interception dynamics for two different control architectures: PID^2 and ADRC || PID^2. The position X - time graph shows the horizontal position of the missile and the target over time. Both control strategies successfully guide the missile to intercept the target at a point marked by the stars, indicating successful interception at nearly the same time. The altitude - time graph shows the altitude of the missile and the target changing with time. The target maintains a relatively constant altitude, while both control architectures can raise the missile's altitude. As a result, both control strategies achieve successful interception at around 6600 m height. The altitude - position X graph highlights the relationship between altitude and horizontal position. These results confirm that despite differences in control behavior, both architectures effectively achieve the interception goal.

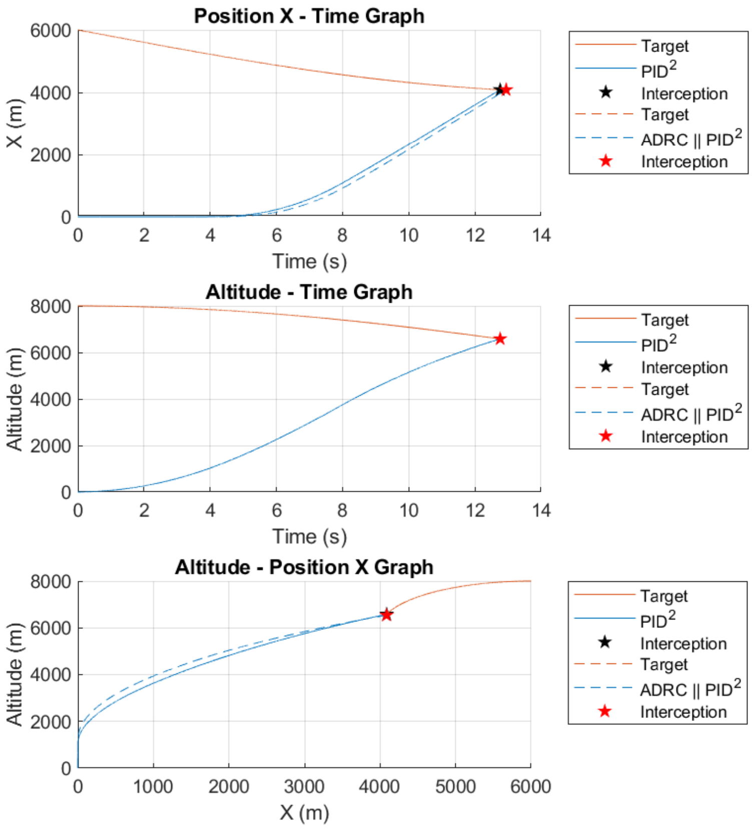


Figure 4.17: Target-1 interception results

Target-1 interception results are given in Table 4.2. Time of flight, impact velocity and miss distance values are very similar for both architectures.

Table 4.2: Target-1 interception results

	Unit	**PID^2**	**ADRC || PID^2**
Time of Flight	s	12.76	12.94
Impact Velocity	Mach	2.46	2.36
Miss Distance	m	0.7	0.1

Target-1 angular velocity changing with time graph is given in Figure 4.18.

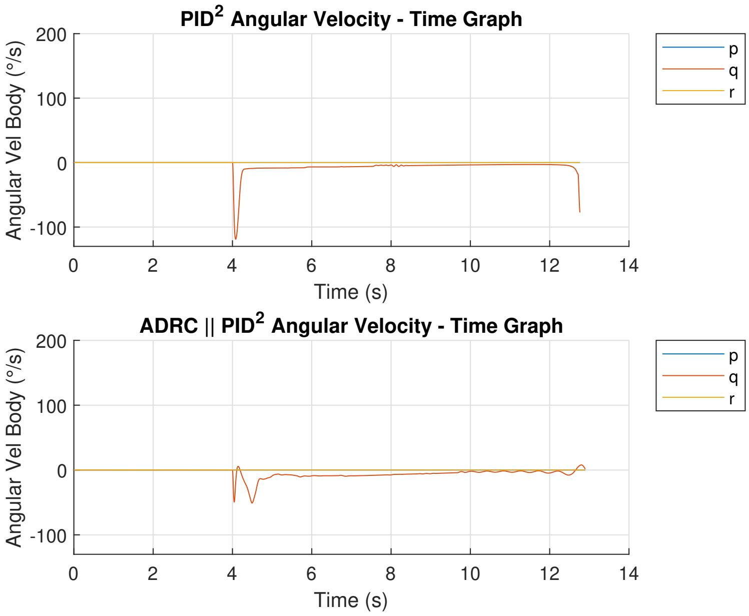


Figure 4.18: Target-1 angular velocity vs time graph

Target-1 Mach number and alpha changing with time graph is given in Figure 4.19.

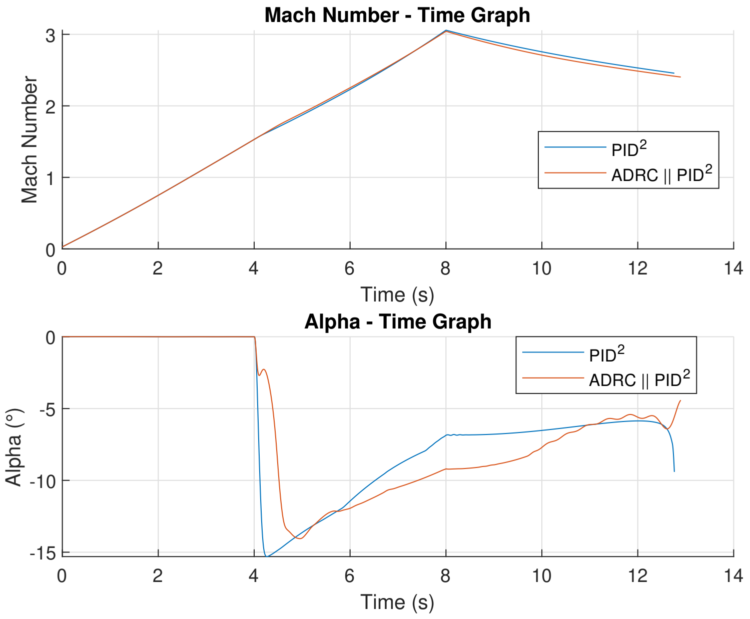


Figure 4.19: Target-1 Mach number and alpha vs time graph

Target-1 delta elevator changing with time graph is given in Figure 4.20.

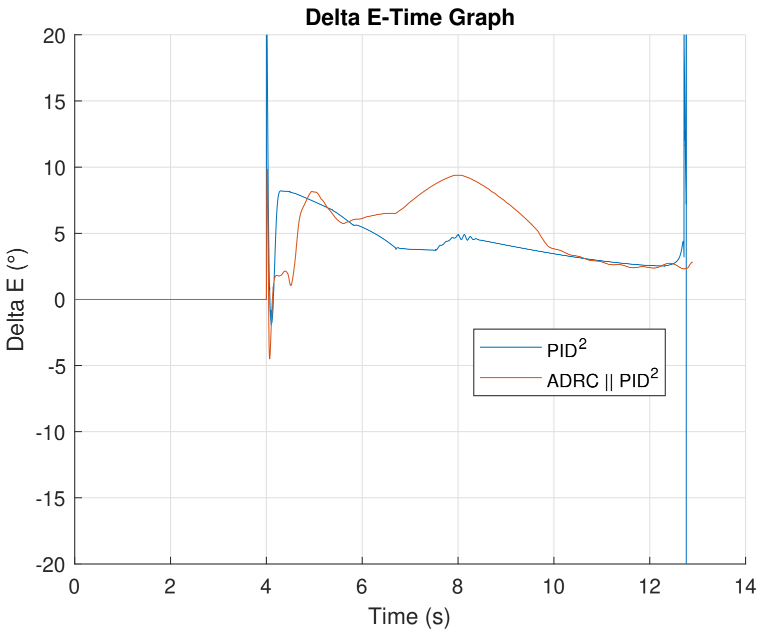


Figure 4.20: Target-1 delta elevator vs time graph

4.2.1.1 Thrust Disturbances and Gust Analysis

Controllers are typically designed based on nominal operating conditions, therefore, it is crucial to evaluate their performance under the effect of uncertain conditions. For this analysis, gust and thrust disturbances are selected as the primary sources of uncertainty. In solid fuel engines, variations in thrust can arise from factors such as fuel distribution, combustion characteristics, and pressure fluctuations, which can differ from one production batch to another. To simulate these uncertainties, the thrust curve is modified by introducing normally distributed white noise, a sine wave with a frequency of 25 Hz, and a half-sine wave with a 0.4 s period starting at 7.8 s and lasting for 0.2 s is used to represent a sudden drop in thrust. As a result, the total thrust output increased to 101254 Ns, corresponding to approximately 1.3% increase in the overall impact. The nominal thrust profile, along with the updated profile incorporating the disturbances are given in Figure 4.21.

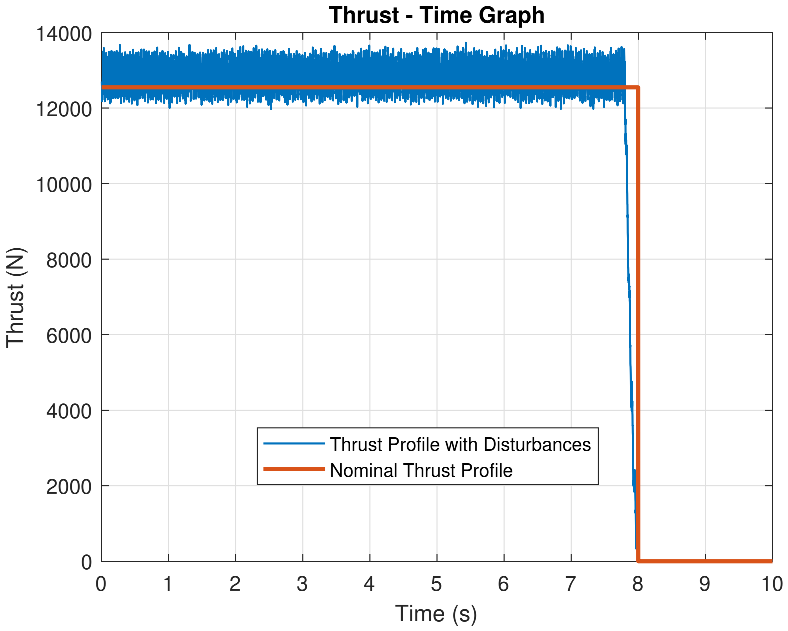


Figure 4.21: Thrust profile with disturbances

A gust disturbance, similar to the profile presented in the previous section, is modeled as a half-sine wave with an amplitude of 8 m/s, starting at 7 s and lasting for 3 s. This force is added to the aerodynamic force block. With the consideration of two disturbances, analysis results demonstrate the robustness of the controllers under real-world uncertainties.

Target-1 interception under the effect of gust and thrust disturbances results are given in Figure 4.22. As illustrated, PID^2 architecture cannot achieve the the interception goal. As examining Figure 4.23, Figure 4.24 and Figure 4.25, it is observed that the missile experiences oscillations, particularly during the periods when the gust is active.

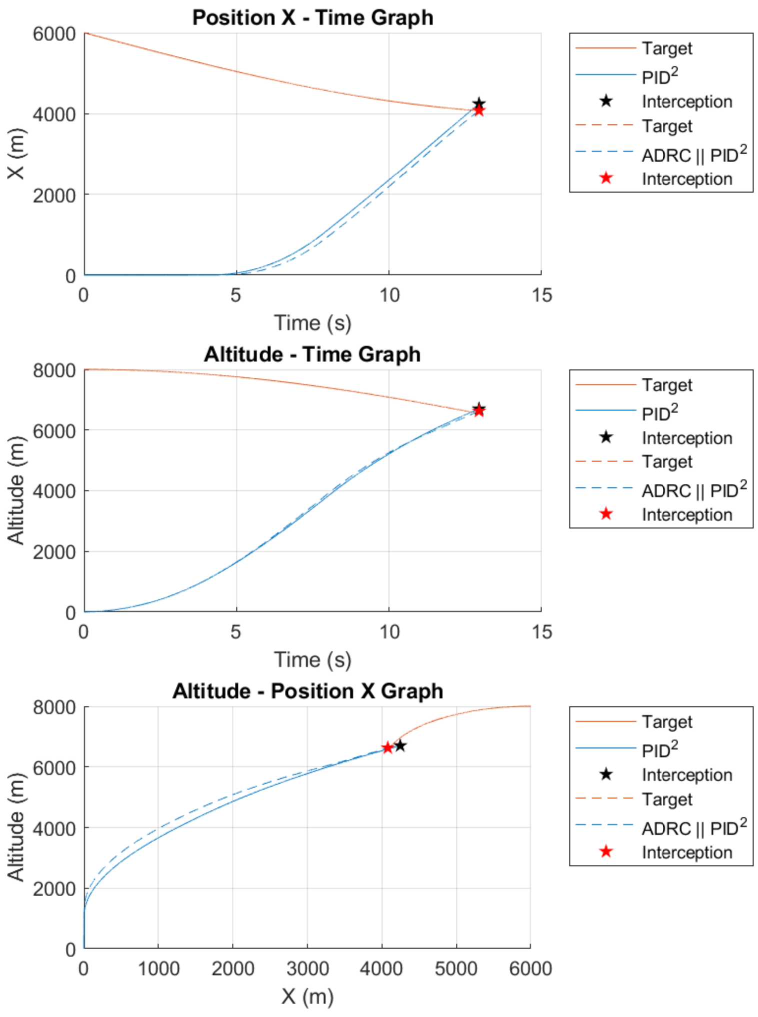


Figure 4.22: Target-1 interception under the effect of gust and thrust disturbances

Target-1 angular velocity changing with time graph under the effect of gust and thrust disturbances is given in Figure 4.23.

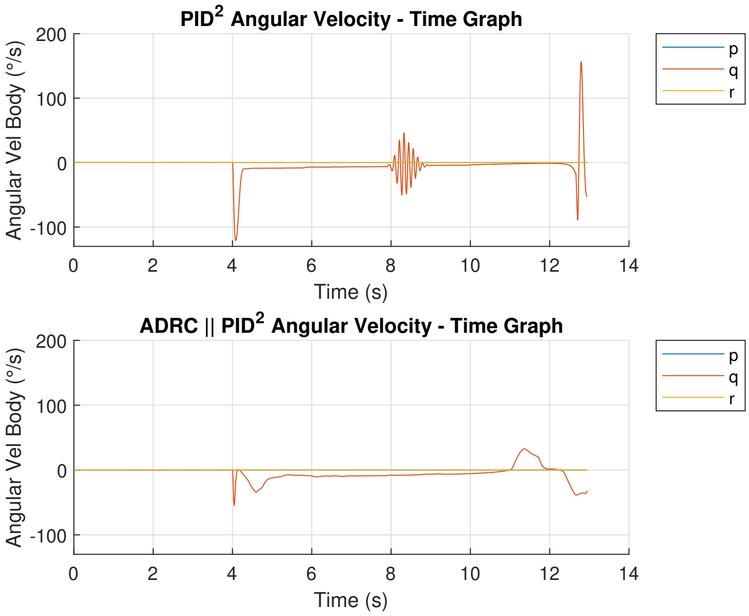


Figure 4.23: Target-1 angular velocity vs time graph under the effect of gust and thrust disturbances

Mach number and alpha changing with time graph under the effect of gust and thrust disturbances is given in Figure 4.24.

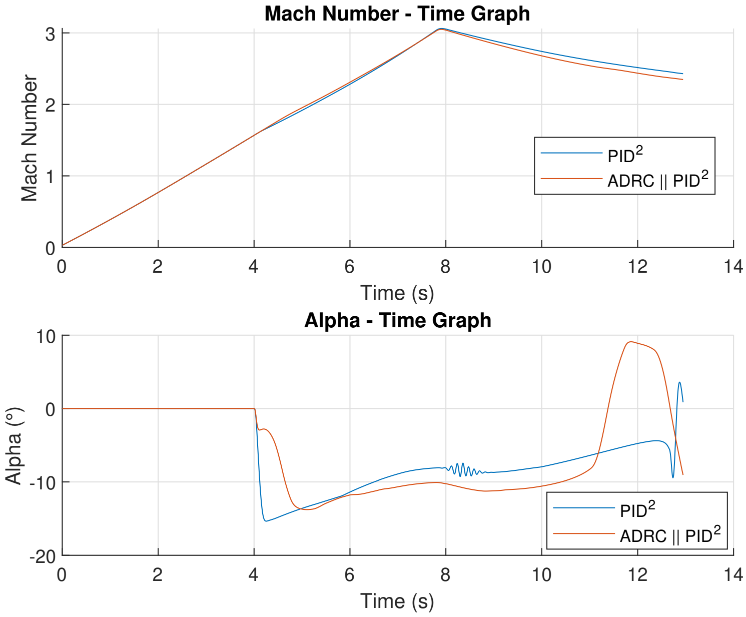


Figure 4.24: Target-1 Mach number and alpha vs time graph under the effect of gust and thrust disturbances

Delta elevator changing with time graph under the effect of gust and thrust disturbances is given in Figure 4.25.

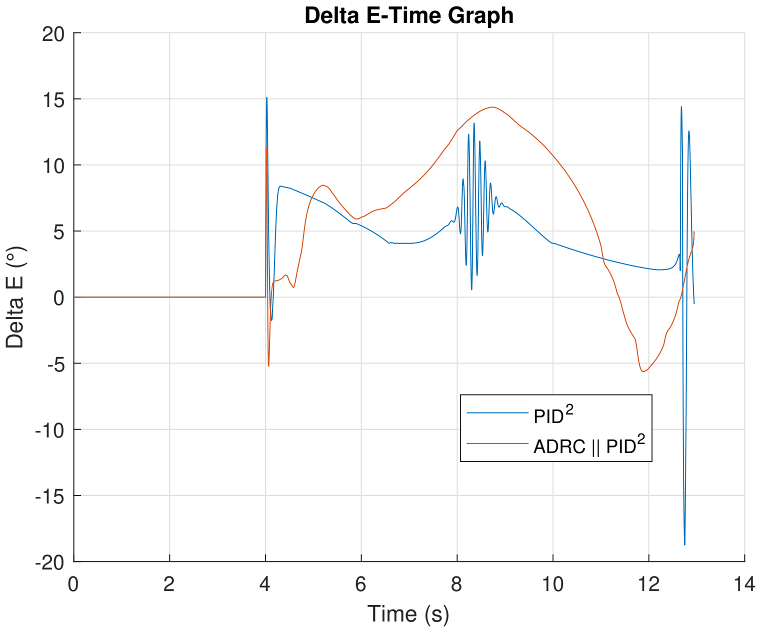


Figure 4.25: Target-1 delta elevator vs time graph under the effect of gust and thrust disturbances

4.2.1.2 Sensor and Measurement Delay Analysis

For the delay analysis, sensor and measurement delays are also considered which can be defined as the lag between the actual physical state of the missile and the time when this information is received by the autopilot system. Sensor processing time, data transmission latency, and computational delays, among other factors, can contribute to these delays. Such delays can affect the feedback loop, leading the system to function with outdated data, which in turn negatively impact precise control. This type of delays must be carefully analyzed that the missile maintains precise guidance and control throughout its flight. This is particularly crucial during high-speed engagements, where even minor timing errors can result in significant performance degradation and missed targets.

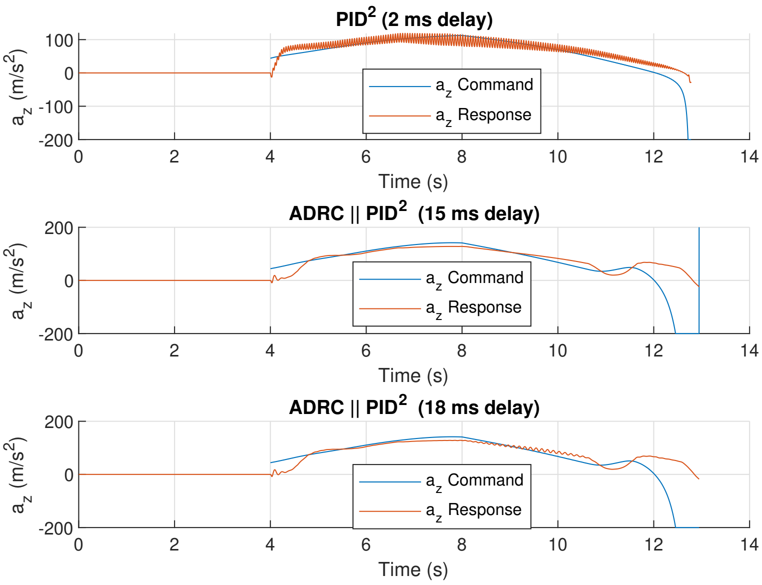


Figure 4.26: Target-1 command and response time graph with autopilot command delay

Target-1 angular velocity changing with time graph with autopilot command delay is given in Figure 4.27.

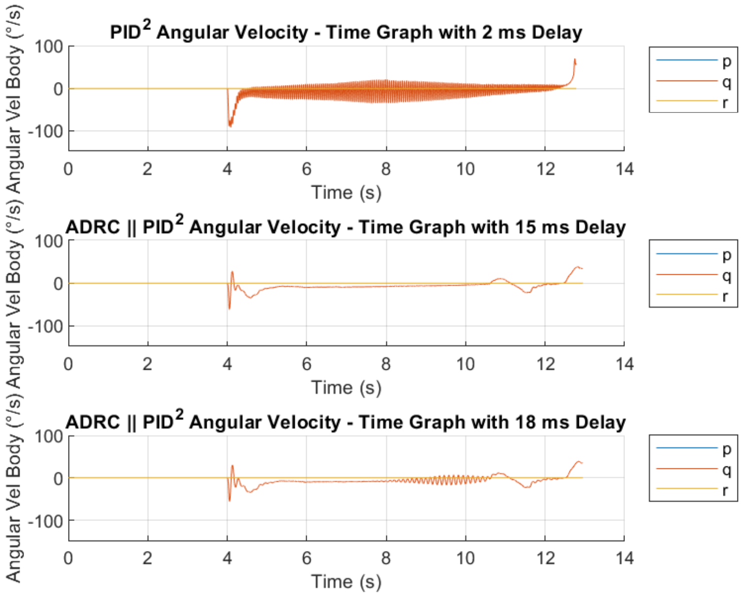


Figure 4.27: Target-1 angular velocity vs time graph with autopilot command delay

Target-1 Mach number and alpha changing with time graph with autopilot command delay is given in Figure 4.28.

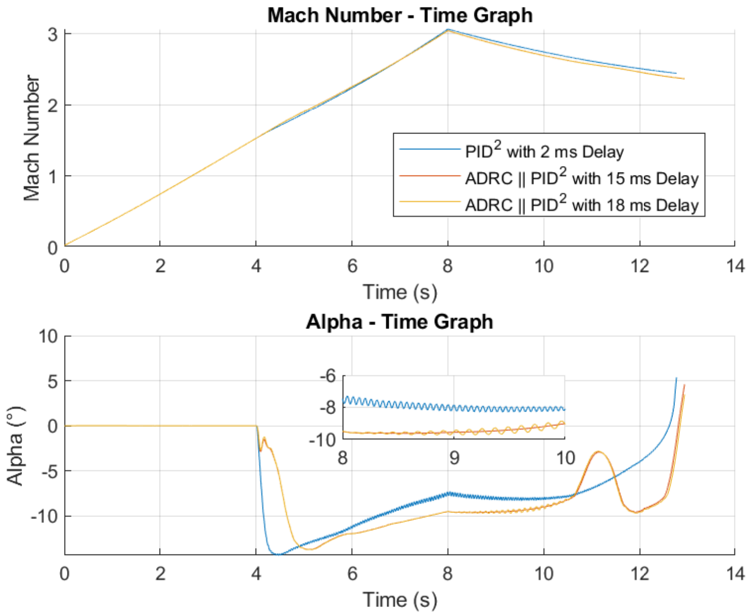


Figure 4.28: Target-1 Mach number and alpha vs time graph with autopilot command delay

Target-1 delta elevator changing with time graph with autopilot command delay is given in Figure 4.29.

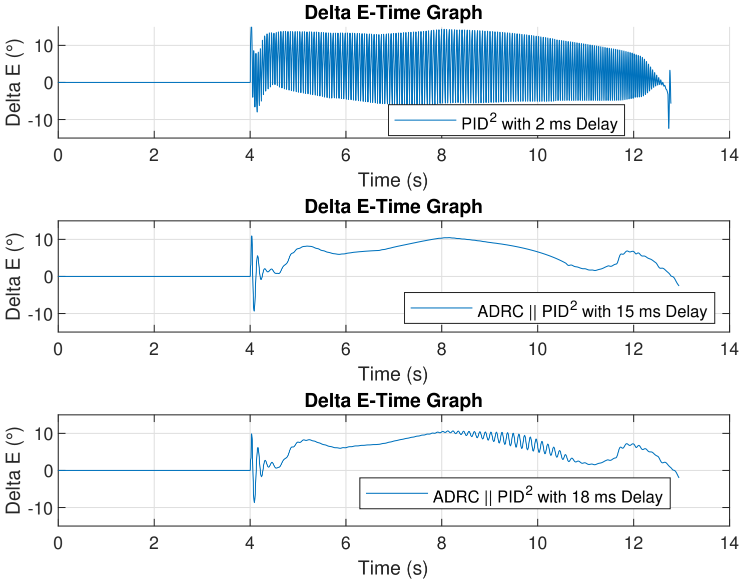


Figure 4.29: Target-1 delta elevator vs time graph with autopilot command delay

As shown in Figure 4.27, Figure 4.28 and Figure 4.29, PID^2 architecture is again highly sensitive to sensor and measurement delay, with significant oscillations starting at 2 ms delay. In contrast, the ADRC || PID^2 configuration operates smoothly up to a delay of 15 ms, with oscillations only beginning to appear at 18 ms.

4.2.1.3 CAS Command Delay Analysis

In the design and evaluation of missile autopilot systems, delays in command execution is critical factor that can significantly affect performance. The autopilot command delay analysis examines the time lag between the control commands and their execution by the actuators. These delays can lead to slower response times and reduced accuracy, especially during rapid maneuvers. In this analysis, delays are systematically introduced at the input of the actuators and incrementally increased for both autopilot configurations. The process continued until noticeable oscillations occurred, impacting system stability. At this point, the corresponding miss distance values are evaluated to assess the performance degradation due to the delays. This approach provides a clear understanding of how increasing delays can affect both control accuracy and overall mission success.

Target-1 angular velocity changing with time graph with CAS command delay is given in Figure 4.30.

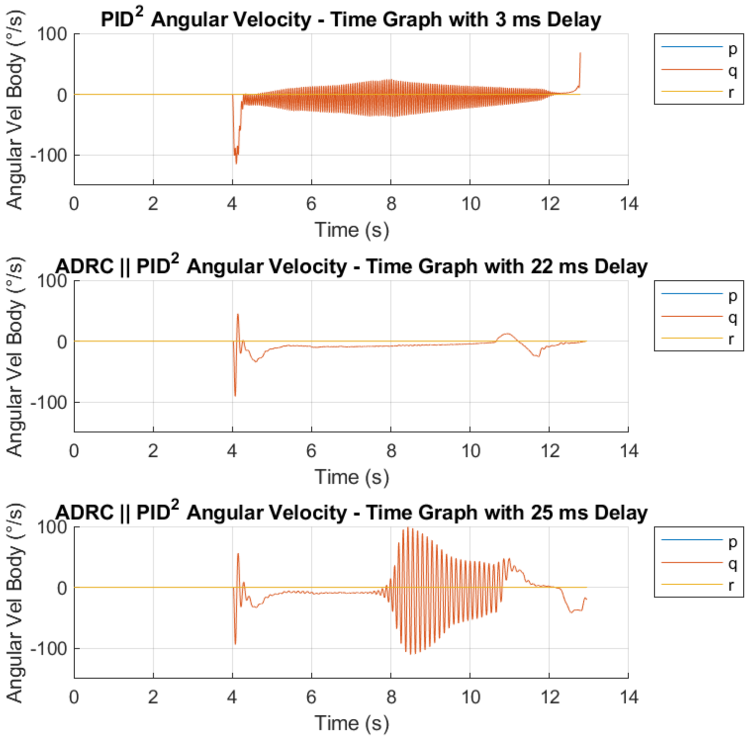


Figure 4.30: Target-1 angular velocity vs time graph with CAS command delay

Target-1 Mach number and alpha changing with time graph with CAS command delay is given in Figure 4.31.

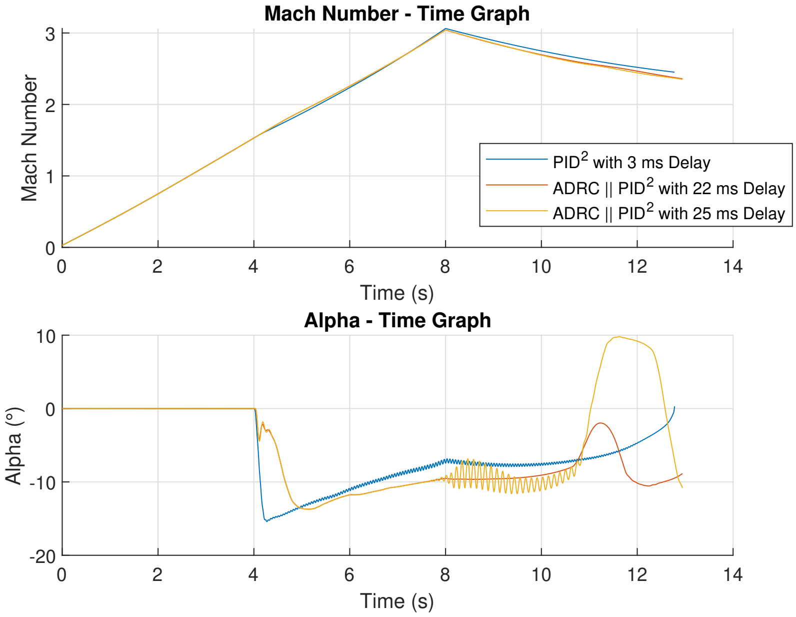


Figure 4.31: Target-1 Mach number and alpha vs time graph with CAS command delay

Target-1 delta elevator changing with time graph with CAS command delay is given in Figure 4.32.

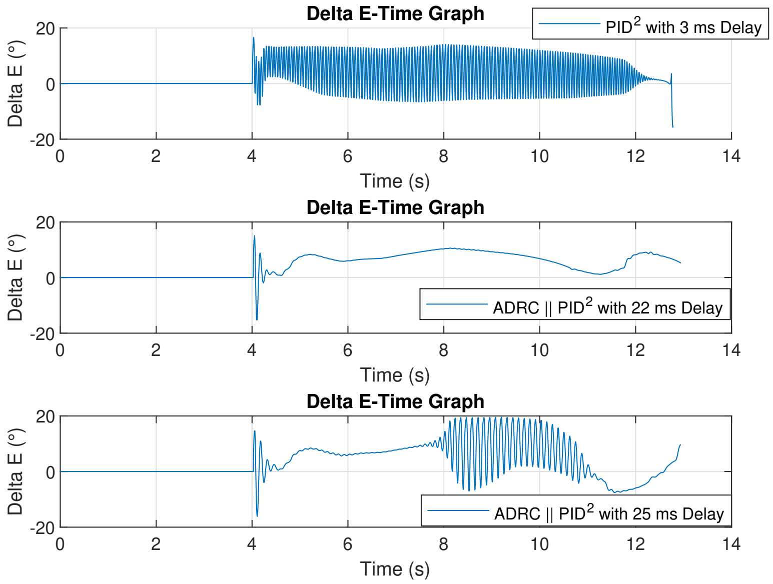


Figure 4.32: Target-1 delta elevator vs time graph with CAS command delay

As shown in Figure 4.30, Figure 4.31 and Figure 4.32, PID^2 architecture is highly sensitive to CAS command delay, with significant oscillations starting at 3 ms delay. In contrast, the ADRC || PID^2 configuration operates smoothly up to a delay of 22 ms, with oscillations only beginning to appear at 25 ms.

Upon comparing the CAS command delay results with the sensor measurement delay results, it is evident that both controllers exhibit slightly more sensitivity to delays originating from sensor measurements. However, the delay analysis results indicate that the ADRC || PID^2 configuration is significantly more robust against delays compared to the PID^2 architecture.

4.2.2 PNG Simulation Results with Target-2

A representative Target-2 profile is provided in Figure 4.33. As illustrated, the target performs a defensive maneuver by raising its nose as the missile closes in. This type of maneuver is particularly challenging for the missile, as it should quickly raise its nose to maintain the proper intercept trajectory. The difficulty increases when the missile's engine is off, as there is insufficient thrust to generate the required lift and maneuverability, making it harder to follow the target's evasive actions. This highlights the critical importance of maintaining sufficient control, maneuver capability and thrust power during the terminal phase of engagement.

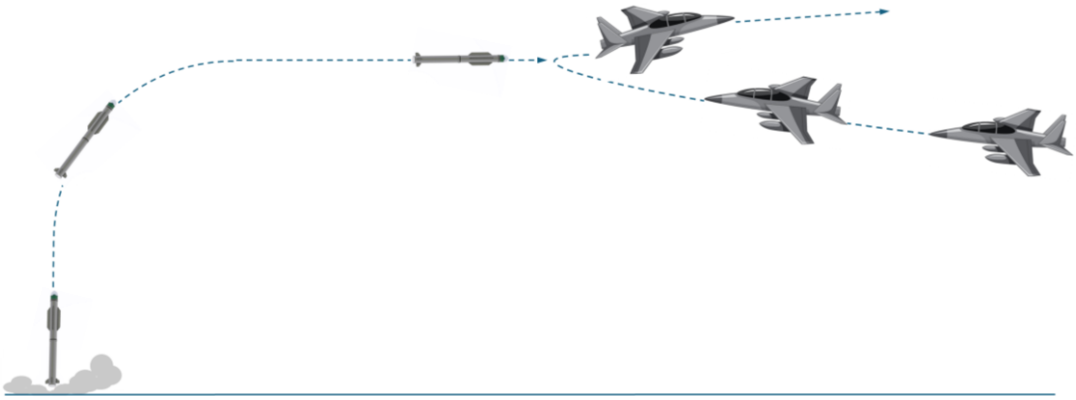


Figure 4.33: A representative Target-2 profile

Target-2 initial conditions are given in Table 4.1.

Table 4.3: Target-2 initial conditions

	Unit	**Target-2**
Initial Position	m	[2500 3000 0]
Initial Velocity	m/s	[200 0 0]
Initial Acceleration	m/s^2	[50 50 0]

Target-2 acceleration command and response time graph changing with time graph is given in Figure 4.34. As shown in the figure, ADRC || PID^2 architecture follows the given commands more closely than the PID^2 architecture.

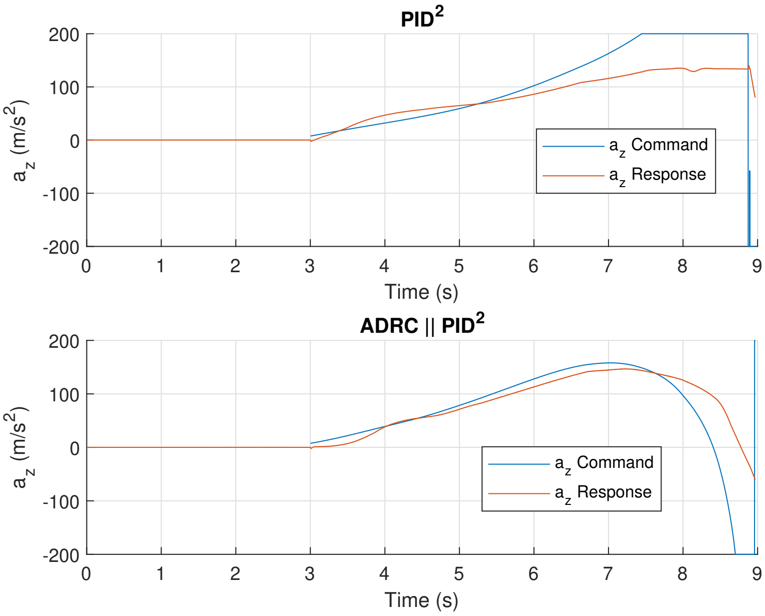


Figure 4.34: Target-2 command and response time graph

Target-2 interception results are given in Figure 4.35

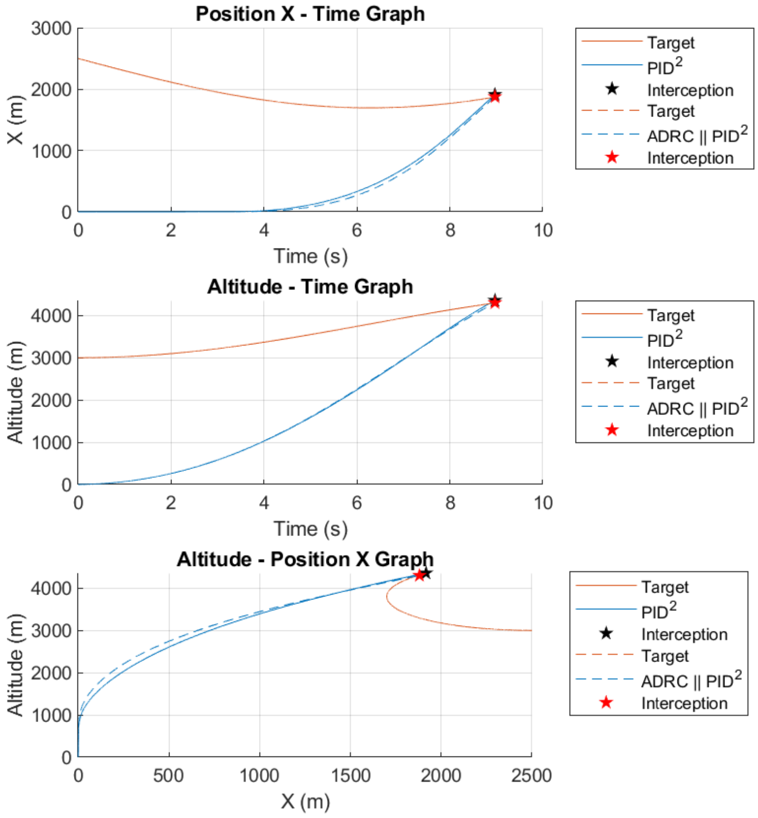


Figure 4.35: Target-2 interception results

Target-2 interception results are given in Table 4.4. The figure shows illustrating the interception dynamics for two different control architectures: PID^2 and ADRC || PID^2 for the Target-2. As clearly shown in the altitude - position X graph, there is a height difference. The PID^2 architecture failed to follow the missile as it raising its nose.

Table 4.4: Target-2 interception results

	Unit	PID^2	ADRC || PID^2
Time of Flight	s	8.88	8.97
Impact Velocity	Mach	2.86	2.84
Miss Distance	m	19.5	1.2

Target-2 angular velocity changing with time graph is given in Figure 4.36.

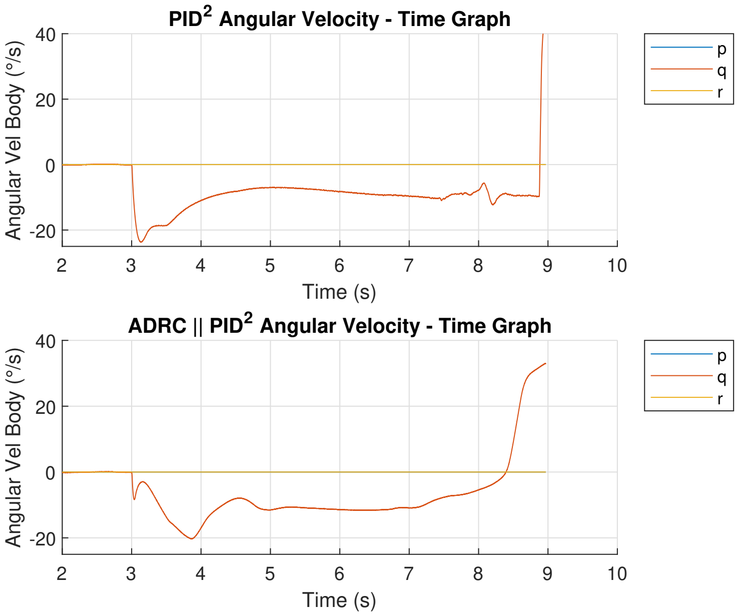


Figure 4.36: Target-2 angular velocity vs time graph

Target-2 Mach number and alpha changing with time graph is given in Figure 4.37.

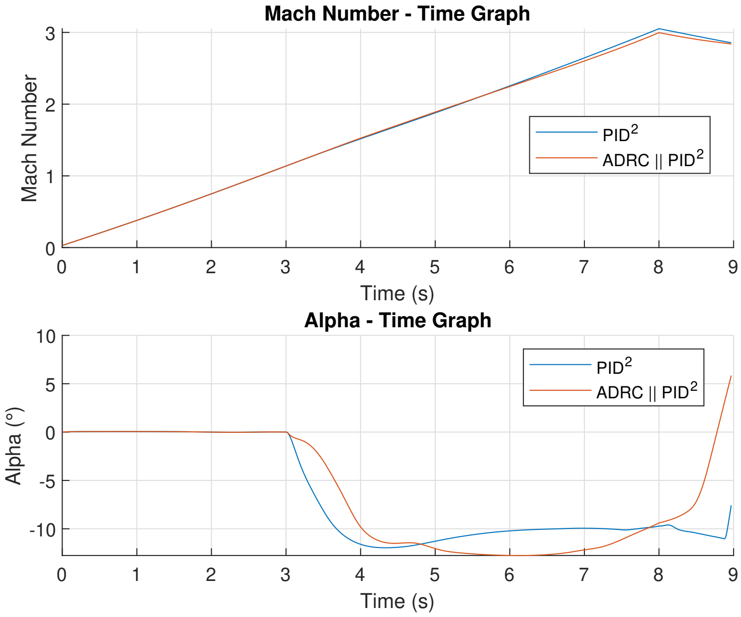


Figure 4.37: Target-2 Mach number and alpha vs time graph

Target-2 delta elevator changing with time graph is given in Figure 4.38.

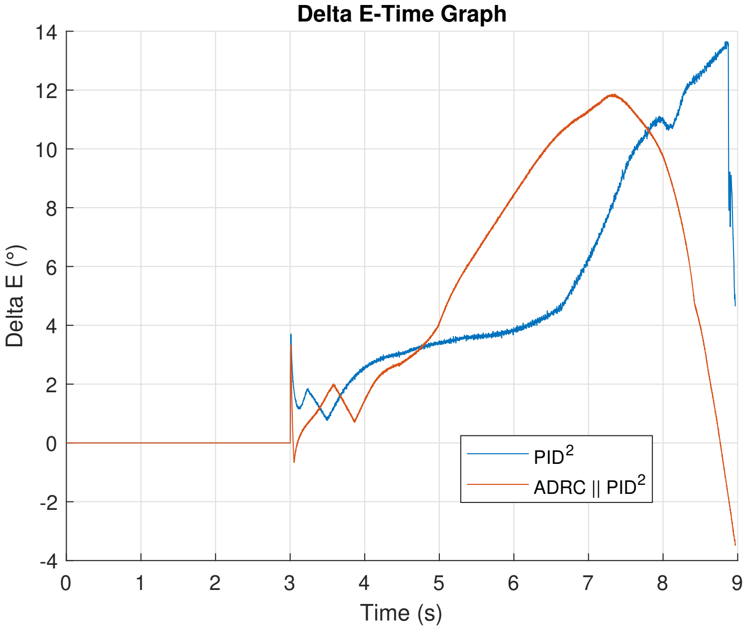


Figure 4.38: Target-2 delta elevator vs time graph

4.2.2.1 Mass Uncertainty Analysis

The influence of mass variations on missile performance is also investigated. The graph in Figure 4.39 shows the Target-2 acceleration command and the corresponding response over time under the effect of the mass variations. Changes in mass can result from differences in payload configuration, manufacturing tolerances, or structural modifications. As shown in Figure 4.39, the system is analyzed under various mass conditions, with mass shifts ranging from -4% to +4%. As mass increases, the system becomes less responsive. This indicates that with a heavier mass, the missile's dynamics are slower, requiring more control effort to follow the command accurately.

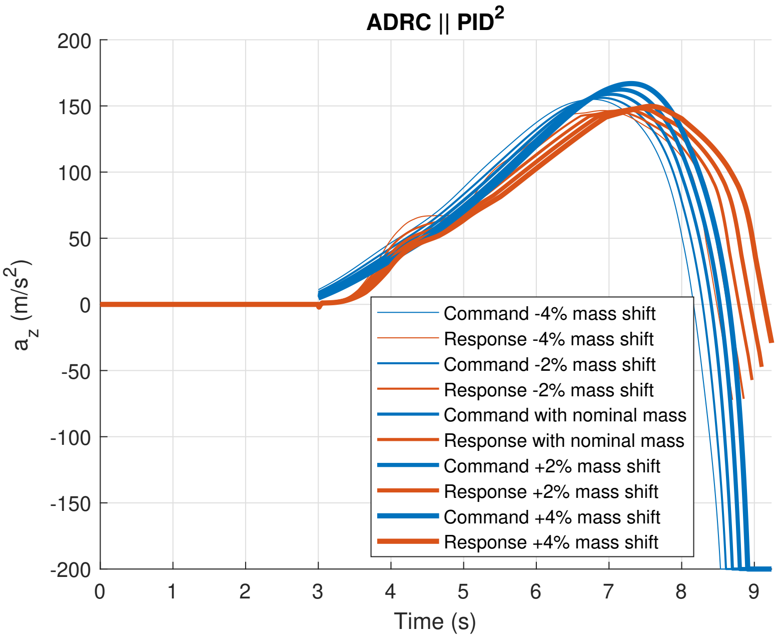


Figure 4.39: Influence of mass variations

Target-2 interception results under the effect of mass variations are given in Table 4.5. The time of flight increases as the mass increases. For the +4% mass shift, the time of flight increases to 9.23 seconds, while for the -4% mass shift, it decreases to 8.72 seconds. This is expected, as a heavier missile will generally take longer to reach the target due to its inertia, while a lighter missile may have a faster response. A -4% mass shift results in the lowest miss distance, showing that the lighter missile performs well in terms of accuracy. As the mass increases, the miss distance also increases, with a +4% mass shift yielding a miss distance of 1.9 meters. This demonstrates that the missile's targeting accuracy degrades as the mass increases, likely due to slower dynamics. However, mass uncertainty in missiles typically does not exceed 2-3%, which is within manageable limits. As a result, the ADRC || PID^2 control architecture exhibits robustness, maintaining effective control performance under mass uncertainties.

Table 4.5: Target-2 interception results under the effect of mass variations

	Unit	-4%	-2%	Nominal	+2%	+4%
Time of Flight	s	8.72	8.84	8.97	9.10	9.23
Miss Distance	m	0.6	0.9	1.2	1.6	1.9

4.2.2.2 CG Uncertainty Analysis

In this study, the impact of CG shifts on the missile's control performance is also investigated. Variations in the CG can occur due to several factors, such as payload distribution, or structural changes, all of which can affect the stability and control of the missile. By introducing CG shifts ranging from -4% to +4%, the robustness of the autopilot system and its ability to maintain accurate control despite these changes are evaluated. This analysis is critical for understanding the missile's performance under real-world operational conditions, where such shifts may lead to deviations from the desired command response. Target-2 acceleration command and response changing with time under the effect of CG variations graph is given in Figure 4.40.

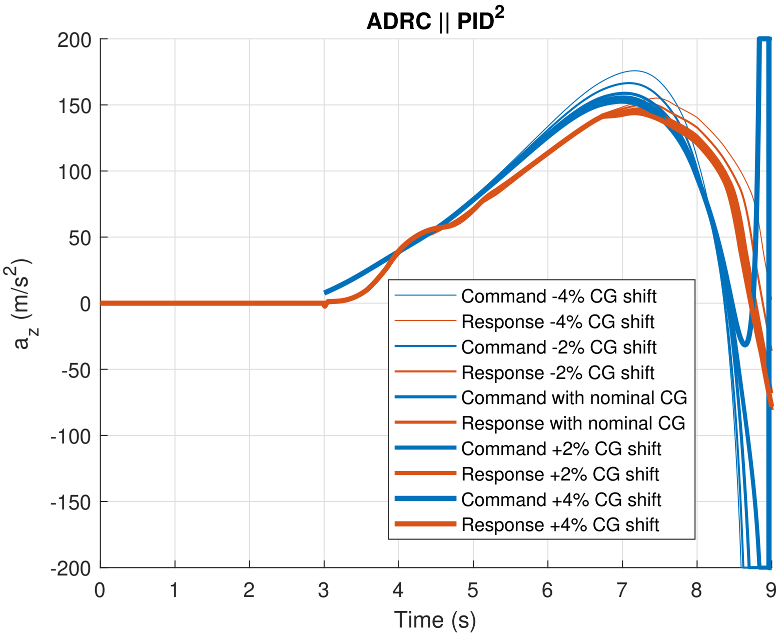


Figure 4.40: Influence of CG variations

Target-2 interception results under the effect of CG variations are given in Table 4.6. The table presents the results of Target-2 interception under varying CG conditions, with shifts of -4%, -2%, nominal, +2%, and +4% from the baseline CG position.

The time of flight exhibits only minor variations across different CG shifts, ranging from 8.96 to 8.99 seconds. This indicates that the missile's overall flight duration remains relatively consistent despite changes in CG. This small fluctuation means that while CG variation may affect dynamic control, it has a limited impact on the missile's overall speed and time to intercept the target.

The miss distance is more sensitive to CG variations. With the nominal CG configuration, the miss distance is 1.2 meters, indicating precise targeting. However, as the CG shifts, the miss distance changes as can be seen from the table. These results indicate that positive CG shifts (+2% and +4%) enhance missile accuracy, reducing miss distance, while negative CG shifts (-4% and -2%) cause a degradation in targeting performance. This is likely due to the effect of the CG moving closer to the CP as the CG shifts forward. As the CG approaches the CP, the missile's stability decreases, which affects its control effectiveness and targeting accuracy. This shift in the CG-CP relationship can lead to degraded performance in maintaining a precise trajectory. However, it is important to note that in missile systems, CG uncertainties typically do not exceed 1%. Even with a 4% CG shift, the results indicate that the ADRC-based architecture demonstrates robustness under CG variations, maintaining acceptable performance. This highlights the system's ability to handle larger-than-expected CG shifts effectively, further validating the robustness of the ADRC || PID^2 controller.

Table 4.6: Target-2 interception results under the effect of CG variations

	Unit	**-4%**	**-2%**	**Nominal**	**+2%**	**+4%**
Time of Flight	s	8.99	8.98	8.97	8.97	8.96
Miss Distance	m	3.3	2.1	1.2	0.4	1.5

CHAPTER 5

CONCLUSIONS

In this thesis, a novel and fully detailed implementation of a nonlinear second-order ADRC for missile control systems has been presented, introducing an innovative approach to missile guidance and control. This new methodology enhances the robustness and precision of missile performance, offering a promising alternative to traditional control strategies in complex and dynamic environments.

Within the scope of this thesis, a comprehensive 6-DOF nonlinear missile model has been developed, representing a tactical surface-to-air defense missile. This model has been analytically formulated to provide a highly realistic simulation of both the missile's dynamics and its operational environment. Additionally, the effects of disturbances, delays, and model uncertainties have been incorporated into the model.

The design and implementation of various autopilot algorithms for selected air defense missiles are thoroughly examined in this thesis. Also, the tuning strategies are provided for the controllers, with particular focus on a second-order nonlinear ADRC. Four distinct types of ADRC and PID autopilot architectures, referred to as PID^2, ADRC^2, ADRC || PID^2, and PID^2 || ADRC, have been implemented in different architectural configurations. These include both parallel and cascaded hybrid configurations. A comparative analysis is conducted to evaluate their respective strengths and weaknesses in missile control systems, focusing on important performance metrics such as control performance, system stability, and response times. The analyses show that architectures with ADRC has a faster response capability compared to architectures with PID controllers. In dual-loop structures, the inner loop is faster, allowing ADRC to be used in the inner loop for quicker responses. However, it has also been shown that as the dominance of ADRC increases, it leads to higher energy consumption. The results highlight the fast response characteristics of ADRC while indicating that energy efficiency considerations must be taken into account. Moreover, analyses performed under external factors such as disturbances, especially thrust uncertainties and sudden gust disturbances, revealed that ADRC responded faster to such situations. Also, the energy consumptions are higher, too. These findings emphasize that control architectures should be carefully selected to optimize the performance and energy efficiency of missile systems.

Finally, an acceleration controller is developed and integrated as a third loop, cascaded with the attitude controllers. In addition, the implementation of PNG with selected two autopilot architectures is thoroughly examined. When the target is maneuvering while the missile is approaching, the architecture with ADRC is able to respond rapidly and successfully intercept with the target, whereas the generic PID-based system exhibits slower response times, resulting in missed intercept. This highlights the adaptability and responsiveness of ADRC in dynamic scenarios compared to conventional PID control. Furthermore, various disturbance effects and delays are modeled and various autopilot architectures are compared in terms of the rejection capability and precision of missile guidance performance. The results from CAS command delay, and sensor measurement delay analyses indicate that the ADRC-based architecture is significantly more robust in handling delays compared to the PID-based architecture. Moreover, ADRC-based control architecture demonstrates strong robustness in handling CG shifts, mass uncertainties, and model inaccuracies. It effectively maintains control performance and stability, even in the presence of substantial variations in system dynamics.

REFERENCES

[1] Devaud, E., Siguerdidjane, H. and Font, S., “Some control strategies for a high angle of attack missile autopilot,” *Control Engineering Practice*, Vol. 8, 2000.

[2] Han, J., “From PID to Active Disturbance Rejection Control”, *IEEE Transactions on Industrial Electronics*, Vol. 56, 2009.

[3] Price, Charles F., *Adaptive Control and Guidance for Tactical Missiles*, The Analytic Science Corporation, 1970.

[4] Lavretsky, E., Wise, K., *Robust and Adaptive Control: with Aerospace Applications*, Springer, 2013.

[5] Chai, R., Tsourdos, A., Savvaris A., "Review of Advanced Guidance and Control Algorithms for Space/Aerospace Vehicles", *Progress in Aerospace Sciences*, Volume 122, 2021.

[6] Smeur, E., Croon, G., Chu Q., "Cascaded Incremental Nonlinear Dynamic Inversion Control for MAV Disturbance Rejection", *Control Engineering Practice*, Volume 73, 2018.

[7] Guo, B.Z. and Zhao, Z.L. *Active Disturbance Rejection Control for Nonlinear Systems*, John Wiley & Sons, 2016.

[8] Fu B., Qi H., Xu J., Yang Y., Wang S., Gao Q., “Attitude Control in Ascent Phase of Missile Considering Actuator Non-Linearity and Wind Disturbance”, *Applied Sciences*, 2019.

[9] Liu, S., Xue, M., Qui, Y., Zhou, X., Zhao, Q., “Design of the Missile Attitude Controller Based on the Active Disturbance Rejection Control”, *Journal of Aerospace Technology and Management*, 2022.

[10] Qiao, H., Meng, H., Ke, W., Gao, Q. and Wang, S. “Adaptive Control of Missile Attitude Based on BP-ADRC”, *Aircraft Engineering and Aerospace Technology*, Vol. 92 No. 10, pp. 1475-1481, 2020.

[11] Siouris, G.M., *Missile Guidance and Control Systems*, Springer-Verlag New York, Inc., 2004.

[12] McLean, D., *Automatic Flight Control Systems*, Prentice Hall Inc., 1990.

[13] Roskam, J., *Airplane Flight Dynamics and Automatic Flight Controls, Part I.*, Design, Analysis and Research Corporation, 2001.

[14] Boiffier, J.L., *The Dynamics of Flight The Equations*, 1st ed., John Wiley & Sons, 1998.

[15] Schmidt, D.K., *Modern Flight Dynamics*, McGraw-Hill, 2011.

[16] Tewari, A., *Automatic Control of Atmospheric and Space Flight Vehicles*, Birkhäuser, 2011.

[17] Cook, M.V., *Flight Dynamics Principles*, 2nd ed., Elsevier, 2007.

[18] Honeywell HG1930 Inertial Measurement Unit. Retrieved July 28, 2024 from https://aerospace.honeywell.com/us/en/products-and-services/product/hardware-and-systems/sensors/hg1930-inertial-measurement-unit

[19] Titterton, D.H.and Weston, J.L., *Strapdown Inertial Navigation Technology*, 2nd ed., The Institution of Electrical Engineers, 2004.

[20] Hernández-Guzmán, V.M. and Silva-Ortigoza, R., *Automatic Control with Experiments*, Springer, 2019.

[21] MATLAB 2024a, MathWorks. Retrieved July 28, 2024 from https://www.mathworks.com/products/matlab.html

[22] Visioli, A., *Advances in Industrial Control*, Springer, 2006.

[23] Ogata, K., *Modern Control Engineering*, 5th ed., Pearson, 2010.

[24] Sira-Ramírez, H. and Luviano-Juárez, A., *Active Disturbance Rejection Control of Dynamic Systems*, Elsevier, 2017.

[25] Yu, C.C., *Autotuning of PID Controllers: A Relay Feedback Approach*, Springer, 2006.

[26] Ran, M., Wang, Q., Dong, C., “Stabilization of a class of nonlinear systems with actuator saturation via active disturbance rejection control”, *Automatica*, Vol. 63, 2016.

[27] Leblebicioğlu, D., *Learning Based Control Compensation for Multi-Axis Gimbal Systems Using Inverse and Forward Dynamics*, MSc Thesis, Bilkent University, Turkey, 2021.

[28] Parsopoulos, K., *Particle Swarm Optimization and Intelligence*, Information Science Reference, 2010.

[29] Clerc, M., *Particle Swarm Optimization*, Wiley-ISTE, 2006.

[30] Kennedy, J. and Eberhart, R., “Particle Swarm Optimization”, *Proceedings of ICNN’95 - International Conference on Neural Networks*, Perth, WA, Australia, pp. 1942-1948, 1995.

[31] Yanushevsky, R., *Modern Missile Guidance*, Taylor & Francis Group, 2007.

[32] Zarchan, P., *Tactical and Strategic Missile Guidance*, American Institute of Aeronautics and Astronautics, 1997.