# INVESTIGATION OF DIFFERENT SYSTEM IDENTIFICATION METHODS FOR LAUNCH VEHICLES

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

BY

HAZEL DEMİREL BAYRI

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
AEROSPACE ENGINEERING

JANUARY 2024

Approval of the thesis:

**INVESTIGATION OF DIFFERENT SYSTEM IDENTIFICATION METHODS FOR LAUNCH VEHICLES**

submitted by **HAZEL DEMİREL BAYRI** in partial fulfillment of the requirements for the degree of **Master of Science in Aerospace Engineering Department, Middle East Technical University** by,

Prof. Dr. Halil Kalıpçılar
Dean, Graduate School of **Natural and Applied Sciences** _____

Prof. Dr. Serkan Özgen
Head of Department, **Aerospace Engineering** _____

Assoc. Prof. Dr. Halil Ersin Söken
Supervisor, **Aerospace Engineering** _____

**Examining Committee Members:**

Prof. Dr. Ozan Tekinalp
Aerospace Engineering, METU _____

Assoc. Prof. Dr. Halil Ersin Söken
Aerospace Engineering, METU _____

Assist. Prof. Dr. Serkan Sarıtaş
Electrical and Electronics Engineering, METU _____

Assist. Prof. Dr. Yakup Özkazanç
Electrical and Electronics Engineering, Hacettepe University _____

Assist. Prof. Dr. Demet Çilden Güler
Astronautical Engineering, Istanbul Technical University _____

Date:17.01.2024

**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: HAZEL DEMİREL BAYRI

Signature :

# ABSTRACT

## INVESTIGATION OF DIFFERENT SYSTEM IDENTIFICATION METHODS FOR LAUNCH VEHICLES

Demirel Bayrı, Hazel
M.S., Department of Aerospace Engineering
Supervisor: Assoc. Prof. Dr. Halil Ersin Söken

January 2024, 102 pages

This thesis focuses on identifying the aerodynamic stability and control derivatives of vehicles, specifically launch vehicles. Such vehicles' dynamics and control exhibit unusual characteristics, including as quick and large changes in inertial and aerodynamic parameters, as well as interactions between low and high-frequency modes. Additionally, accurate vehicle modeling is difficult since, for example, dedicated flight testing with specific maneuvers are disallowed due to stringent safety and financial constraints. The Pegasus launch vehicle's aerodynamic database and first flight test data were used as a case study to evaluate the effectiveness of system identification methods. First, outcomes of a three degree-of-freedom model were used in the least square estimation method to predict aerodynamic derivatives. Furthermore, feed-forward neural network approach was used to predict total aerodynamic force and moment coefficients, and the least square method was used to estimate aerodynamic derivatives from the outputs of the neural model. On the other hand, the output error approach has been developed as a different method to obtain aerodynamic derivative coefficients. Lastly, the predictability of the parameters for these three techniques was compared by using mean-squared errors.

Keywords: Aerodynamic Parameters, System Identification, Least-Square, Neural Network, Output-Error, Pegasus Launch Vehicle

# ÖZ

## FIRLATMA ARAÇLARI İÇİN FARKLI SİSTEM TANILAMA YÖNTEMLERİNİN İNCELENMESİ

Demirel Bayrı, Hazel
Yüksek Lisans, Havacılık ve Uzay Mühendisliği Bölümü
Tez Yöneticisi: Doç. Dr. Halil Ersin Söken

Ocak 2024 , 102 sayfa

Bu tez, özellikle fırlatma araçları için, araçların aerodinamik kararlılık ve kontrol türevlerini tanılama üzerine odaklanmaktadır. Bu tür araçların dinamikleri ve kontrol sistemleri, ataletsel ve aerodinamik parametrelerde hızlı ve büyük değişiklikler, ayrıca düşük ve yüksek frekanslı modların etkileşimi gibi olağandışı özellikler sergilemektedir. Ek olarak, belirli manevralarla uçuş testi yapma imkanı sıkı güvenlik ve finansal sınırlamalar nedeniyle kısıtlıdır. Pegasus fırlatma aracının aerodinamik veri tabanı ve ilk uçuş test verileri, sistem tanımlama yöntemlerinin verimliliğini değerlendirmek için bir örnek çalışma olarak kullanılmıştır. Öncelikle, üç serbestlik dereceli bir modelin sonuçları ile en küçük kareler tahmin yöntemi kullanılarak aerodinamik türevler tahmin edilmiştir. Ayrıca, toplam aerodinamik kuvvet ve moment katsayılarını tahmin etmek için ileri beslemeli sinir ağı yaklaşımı kullanılmış ve sinir modelinin çıktılarından aerodinamik türevleri tahmin etmek için en küçük kareler yöntemi kullanılmıştır. Öte yandan aerodinamik türev katsayılarını elde etmek için farklı bir method olarak çıktı hatası yaklaşımı geliştirilmiştir. Son olarak, bu üç teknik ile parametrelerin tahmin edilebilirliği, ortalama hataların kareleri kullanılarak

karşılaştırılmıştır.

Anahtar Kelimeler: Aerodinamik Parametreler, Sistem Tanımlama, En Küçük Kareler, Yapay Sinir Ağları, Çıktı-Hatası, Pegasus Fırlatma Aracı

To my loving parents...

## ACKNOWLEDGMENTS

I am grateful to Assoc. Prof. Halil Ersin Söken, my thesis advisor, for his exceptional mentorship, intelligent direction, and support during this journey. The direction of this effort has been greatly influenced by his knowledge and commitment.

I want to express my sincere thanks to my company, ROKETSAN, for giving me the chance to carry out this research. I extend my sincere gratitude to my manager, Fahrettin Kağan İpek, for his understanding and unwavering support throughout this project, along with all of my teammates for their valuable support and direction. Also, my teammates Merve Çelikbudak Altıntaş and Atakan Süslü deserve special thanks for their contributions to this thesis study.

I would like to express my appreciation to my wonderful husband, Berke Bayrı, for being a source of strength to me via his faith, support, and endless patience. His understanding and encouragement have kept me going in this endeavor.

I am grateful to my lovely parents, Serpil and Teoman, whose everlasting faith and support I constantly feel; I am grateful to my siblings, İrem and Nezir, who always motivate me with their unwavering love.

Finally, I would like to thank myself for the effort I have put forth and for the personal growth I have experienced throughout this study process.

The contributions of everyone who helped me along the way, whether by advice, support, or encouragement, were significant. This thank you reflects my heartfelt gratitude and appreciation.

## TABLE OF CONTENTS

# LIST OF TABLES

TABLES

## LIST OF FIGURES

FIGURES

## LIST OF ABBREVIATIONS & SYMBOLS

| | |
|---|---|
| BP | Back Propagation |
| CFD | Computational Fluid Dynamics |
| DOF | Degree of Freedom |
| EOM | Equations of Motion |
| FFNN | Feed-Forward Neural Network |
| IMU | Inertial Measurement Unit |
| LSE | Least-Squares Estimation |
| LV | Launch Vehicle |
| ML | Maximum Likelihood |
| MSE | Mean-Squared Error |
| NN | Neural Network |
| OEM | Output-Error Method |
| $a_x, a_y, a_z$ | translational accelerations in the the body frame |
| $p, q, r$ | angular rates in the the body frame |
| $u, v, w$ | translational velocity components in the the body frame |
| $V_{body}$ | resultant translational velocity in the body frame |
| $\phi, \theta, \psi$ | Euler angles (roll, pitch, yaw) |
| $\alpha, \beta$ | wind angles (angle of attack, side-slip angle) |
| $\delta_e, \delta_r, \delta_a$ | control surface deflections (elevator, rudder, aileron) |
| $F_x, F_y, F_z$ | forces in the body frame |
| $C_{F_x}, C_{F_y}, C_{F_z}$ | non-dimensional forces in the body frame |
| $M_x, M_y, M_z$ | moments in the body frame |
| $C_{M_x}, C_{M_y}, C_{M_z}$ | non-dimensional moments in the body frame |

| | |
|---|---|
| $C_{F_x\alpha}, C_{F_y\alpha}, C_{F_z\alpha}$ | aerodynamic force derivative coefficients with respect to the angle of attack in the body frame |
| $C_{F_x\delta_e}, C_{F_y\delta_e}, C_{F_z\delta_e}$ | aerodynamic force derivative coefficients with respect to elevator angle in the body frame |
| $F_t$ | thrust force |
| $C_L$ | non-dimensional lift force |
| $C_D$ | non-dimensional drag force |
| $C_{L\alpha}, C_{D\alpha}$ | lift and drag force derivative coefficients with respect to the angle of attack |
| $C_{L\delta_e}, C_{D\delta_e}$ | lift and drag force derivative coefficients with respect to elevator angle |
| $C_{M_y\alpha}$ | pitch moment derivative coefficient with respect to angle of attack |
| $C_{M_y\delta_e}$ | pitch moment derivative coefficient with respect to elevator angle |
| $Q$ | dynamic pressure |
| $\rho$ | air density |
| $P$ | ambient pressure |
| $P_0$ | sea level pressure |
| $T$ | temperature |
| $a$ | speed of sound |
| $h$ | altitude |
| $S_{ref}$ | wing reference area |
| $l_{ref}$ (or $\bar{c}$) | reference length (mean aerodynamic chord) |
| $b$ | wing span |
| $T_{net}$ | net thrust force |
| $T_{vacuum}$ | vacuum thrust |
| $g$ | gravitational acceleration |

| | |
|---|---|
| $G$ | universal gravitational constant |
| $M_e$ | mass of earth |
| $R_e$ | radius of earth |
| $m$ | mass |
| $I_{xx}, I_{yy}, I_{zz}$ | mass of inertias |
| $I_{xy}, I_{yz}, I_{xz}$ | mass of inertias cross terms |
| $x_{cg}$ | Center of Gravity |
| $x_{cp}$ | Center of Pressure |
| $J$ | cost function or Jacobian |
| $E$ | error |
| $I$ | identity matrix |
| $\mu$ | learning rate of neural model |
| $\Omega$ | momentum factor of neural model |
| $\nu$ | measurement error vector |
| $\Theta$ | unknown parameter vector |
| $R$ | measurement error covariance matrix |

# CHAPTER 1

# INTRODUCTION

## 1.1 Scope of the Research

Aerodynamic parameter estimation is an important study area in the field of aerospace engineering. Aerodynamic parameter estimation results have a wide range of applications, such as validating and improving computational fluid dynamic analyses or wind tunnel test results, improving flight models, and researching stability-control impacts.

The dynamics and control of rockets and launch vehicles differ enormously from that of ordinary aircraft, with rapid and major changes in inertia and aerodynamic properties. Furthermore, accurately modeling these vehicles can be difficult because of the restrictions enforced by strict safety standards and financial constraints, which disallow specialized or frequent flight tests as is standard with aircraft.

This thesis aims to identify the aerodynamic coefficients, stability, and control derivatives of a launch vehicle. The Pegasus air-launched rocket is used as a case study. The aerodynamic derivative coefficients, including the longitudinal force and moment coefficients and their variations concerning angles of attack and control surface deflections, have been identified. The least squares method, the neural network approach, and the output-error technique are different methods whose estimation performance is evaluated.

## 1.2 Proposed Methods and Models

A 3-degree-of-freedom model is created using the aerodynamic database and the Pegasus's characteristic features. The model's results are fed to the least squares algorithm, which estimates the aerodynamic derivative coefficients. The performance of the least squares method is evaluated for various control surface inputs.

The 3-degree-of-freedom model developed for the least squares algorithm is a basic framework that does not account for any uncertainties or complexities that may exist in the system. As a result of noticing significant imperfections in the estimations acquired through this approach, an artificial neural network model is built. A neural model was developed using the feed-forward method, with the inputs being accelerations, angular velocity, angular velocity rate, body speed, angle of attack, and control surface commands, and the outputs being aerodynamic total force and moment coefficients. The neural model is trained with the Levenberg-Marquardt method using many different data sets recorded from the 3-DOF model. Once the model is trained, it is used to obtain the aerodynamic force and moment coefficients. Then, aerodynamic derivative coefficients are obtained using the outputs of the neural model together with the least-squares method.

In addition to aerodynamic parameter estimation with the equation error approach using two different models, the aerodynamic parameter estimation is repeated with the output error method, which is a maximum-likelihood approach. The output error approach is aimed to find the parameters that will minimize the error between the outputs of the defined mathematical model and the measurements. The Gauss-Newton method is used as the optimization method to reduce the error. When the classical output error method (single-shooting) is used for high-precision and unstable systems such as the Pegasus launch vehicle, numerical divergence problems are experienced in the optimization, so the algorithm is modified with the multiple-shooting approach.

## 1.3 Literature Review

The paper [4], named "Estimation of Aerodynamic Angles and Wind Components for a Launch Vehicle", mentions the estimation of aerodynamic angles and wind components for a launch vehicle. In this study, the authors create an estimation model for aerodynamic angle dynamics and wind velocity combined by using an Iterated Extended Kalman Filter. They tested their methodology for atmospheric flight conditions by simulations for the Ares-I launch vehicle flight model. According to their tests, their method converged efficiently to estimate the angle of attack, sideslip angle, and wind speed at higher uncertainties. According to the authors, this technique can be used for online aerodynamic angle predictions in launch vehicles' guidance and control systems during flight.

In the paper "Aerodynamic Parameter Estimation of a Supersonic Air-to-Air Missile with Rapid Speed Variation" [7], the aerodynamic parameter estimation for a supersonic missile using an Adaptive Kalman Filter technique is discussed. This approach is intended to address the challenges of missile flight, such as rapid Mach number changes and inadequate flight test data. The research paper includes building a six-degree-of-freedom missile model, integrating nonlinear aerodynamics, and carrying out simulation tests. The method employs an adaptive Kalman filter to identify post-flight aerodynamic parameters while accounting for measurement noise and process unknowns. They proved the accuracy and efficiency of the method by running Monte Carlo simulations, which take into account the uncertainties from instrumentation and the aerodynamic model. Similarly, research [19] proposes a unique adaptive filter approach for aerodynamic parameter estimation utilizing dual Unscented Kalman Filters, exhibiting higher accuracy in parameter estimation under varied noise situations compared to traditional methods.

In the thesis study "Aerodynamic Parameter Estimation of a Missile" [2], Aksu introduces a work about the estimation of aerodynamic parameters of a missile under limited wind angle measurements by using the Output Error method. The output error method is well-suited for dealing with the complex factors and unknowns related to missile flight dynamics, particularly when direct wind angle data is rare or inaccessible. In-depth testing using simulated and actual flight test data is used in the thesis to illustrate the application of this approach successfully. To overcome the limitations of

insufficient direct measurements, the output error method is used to provide more precise and trustworthy predictions of the aerodynamic parameters. The application of this technique represents an important development in the field, providing improved accuracy in aerodynamic parameter estimation for missiles in limited measurement circumstances.

Another thesis work, similar to the one mentioned previously, is "Aerodynamic Parameter Estimation of a Missile in Closed Loop Control and Validation with Flight Data" [6]. Aydın developed an estimation algorithm for the aerodynamic parameter estimation of a missile by using closed-loop system data in his thesis. In particular, the thesis analyses the effect of the distinct surface excitations on autopilot rejection of disturbance outcomes and suggests a design procedure to generate them. This work uses two different estimation techniques, such as ordinary least squares and complex linear regression, to estimate aerodynamic parameters. These methods are compared to one another as well as to an existing aerodynamic database. Furthermore, the thesis ensures the usefulness of these methods by showing their importance and efficiency in real-world scenarios by implementing them on a real missile system.

The study "Aerodynamic Parameter Estimation for Launch Vehicles" [15] explores the development and implementation of an innovative technique to estimate the aerodynamic properties of launch vehicles. The E-optimal design concept and statistical approaches are used in this strategy to solve the challenge of accurately estimating aerodynamic parameters given crucial flight data, such as height, velocity, and distance traveled. The method's ability to greatly improve prediction precision and its resilience to initial state perturbations are highlighted in the paper, making it a viable tool to improve launch vehicle performance evaluation and design.

The study [12] focuses on establishing strategies for maximum likelihood estimation-based flight mechanics parameter estimation. It addresses the difficulties in nonlinear systems with nonlinear characteristics in control and state variables. Minimum search techniques and quasi-linearization are the two main approaches investigated. The study highlights the advantages and disadvantages of each approach and provides examples of how to use them to estimate aerodynamic derivatives and verify the kinematic consistency of flight test data. The study advances the field by offering computationally effective methods for estimating parameters in complex nonlinear systems.

A comprehensive analysis of the estimation of aerodynamic characteristics for the X-43A hypersonic research vehicle using data from its third flight is presented in the publication "Aerodynamic Parameter Estimation for the X-43A (Hyper-X) from Flight Data" [23]. Equation-error parameter estimation in the frequency domain is used in the study to extract control and stability derivatives from flight data. The study emphasizes how to provide dynamic excitation with multiple orthogonal phase-optimized sweep inputs; this method works well even when a feedback control system is in place. The results validate the correctness of the utilized aerodynamic models by showing a good fit between flight estimations and pre-flight aerodynamic database values.

The paper [18] presents an easy-to-use application for Gaussian process modeling that is specifically intended for predicting missile aerodynamic coefficients. This tool handles single and multi-fidelity aerodynamic datasets using both standard kriging and co-kriging models. The process of creating graphical user interfaces (GUIs) for these models is described in depth in the study. This makes the models more user-friendly and accurate for aerodynamic estimations in missile design and analysis, even for those without strong experience in statistical methods.

The study "Aerodynamic Coefficient Identification of a Space Vehicle from Multiple Free-Flight Tests" [3] is concerned with identifying aerodynamic coefficients for reentry space vehicles using data obtained from numerous free-flight tests. It provides a thorough method for estimating pitch damping coefficients, drag, and pitching moment slope using polynomial functions of incidence angles and Mach values. This technique greatly advances the field of aerodynamic research for space vehicles and improves the accuracy of aerodynamic coefficient estimations.

In the paper "Aircraft Parameter Estimation Using a New Filtering Technique Based Upon a Neural Network and Gauss-Newton Method" [29], it is mentioned that a method uses a feed-forward neural network for aerodynamic parameter estimation of an aircraft. By concentrating on the fitting of input and output data, this approach creates a neural model for time history forecasting based on measured initial conditions. The Gauss-Newton method is used to optimize aerodynamic parameter values by minimizing the error cost function. They proved their method with different flight test data. They made comparisons with the result of wind tunnel tests and different estimation methods, such as Maximum Likelihood and Filter Error methods. The im-

portant point of the usage of neural networks is that it does not require a description of the flight model, and it eliminates the procedure of solving equations of motion. Hence, it becomes beneficial for dealing with flight data from unsteady aircraft.
Similarly, the paper "Aircraft Parameter Estimation Using Neural Network Based Algorithm" [30] introduces a technique for aerodynamic parameter estimation by using a neural network and Gauss-Newton optimization. This paper mentioned a similar method that employs a black box approach to develop the flight dynamic model from the measured flight data. Hence, that method can simplify the procedure by eliminating the requirement to solve equations of motion in order to figure out motion parameters. It is also mentioned that flight data from the HANSA-3 aircraft from the Indian Institute of Technology Kanpur and lateral-directional flight data from the ATTAS aircraft from DLR Germany were used to validate the method. Furthermore, they compared their results using other estimation techniques, such as the Least Square and Filter Error methods. Moreover, real flight data from the HFB-320 test aircraft was used to evaluate the algorithm's capacity to handle flight data with atmospheric turbulence. The research culminates in a comparison of the approximations generated through several techniques on the flight data of three distinct experimental aircraft.
The references "Nonlinear Longitudinal Aerodynamic Modeling Using Neural Gauss-Newton Method" [16] and "Estimation of Longitudinal Aerodynamic Parameters Using Recurrent Neural Network" [31] contain very similar studies with the references [29] and [30] which are mentioned above.

## 1.4 The Contribution and The Outline of the Thesis

This research aims to use and compare three alternative system identification approaches (Least Squares Technique, Neural Network + Least Squares, and Output-Error Method) in identifying the aerodynamic parameters of a launch vehicle. Studies using similar techniques on more stable systems, such as airplanes, are often found in the literature. Furthermore, when studies in the literature are reviewed, it is discovered that aerodynamic parameter estimation is conducted based on data collected for a short period by conducting flight tests at a constant altitude and constant speed zone. The goal of this research is to estimate the aerodynamic characteristics of launch ve-

hicles and rockets, which have more sensitive and rapidly changing dynamics and cannot be tested under constant altitude and constant speed conditions.
It is thought that this research will contribute to the literature with the comparative analysis of methods used in aerodynamic parameter estimation for launch vehicles. The results of this study may guide similar future research by highlighting the advantages and limitations of different methods used in aerodynamic parameter estimation. In particular, this comprehensive comparative analysis to determine the most appropriate estimation method for launch vehicles with fast and sensitive system dynamics may be of significant importance in terms of design and performance optimization in the aerospace industry. In addition, it is believed that the methods in this study can be used in areas such as verifying CFD analyses of an aircraft with fast dynamics, such as a launch vehicle, modeling closer to reality, and optimizing autopilot algorithms.

In Chapter 1, the scope of the research is briefly mentioned, and the methods and models used in the study are introduced. Also, similar research studies and some applications, which are the inspiration for this thesis, in literature are explained in Chapter 1.
Chapter 2 introduces the technical essentials of the research by explaining the launch vehicle, which is taken as a case in the study, the mathematical model, and the reference frames. Firstly, some characteristics and trajectory of the Pegasus launch vehicle are briefly mentioned. Secondly, reference frames and some assumptions used in the vehicle modeling process are introduced in this chapter. Lastly, the mathematical modeling subsection explains each part of 3 degree-of-freedom modeling.
In Chapter 3, aerodynamic parameter estimation using the least squares method is explained. After introducing the least square theory, the control surface input design study, which is about the relation between the model input and the efficiency of the least square algorithm, is mentioned. Then, the application of the least squares method for the aerodynamic derivatives estimation of the Pegasus launch vehicle and results are discussed.
In Chapter 4, it is explained how to perform aerodynamic parameter estimation using the artificial neural network approach. Firstly, the feed-forward neural model structure is described along with important considerations for developing the model struc-

ture. Next, the procedures and techniques for training the neural model are described, along with crucial hyperparameters. After evaluating the neural model's performance, the estimate of aerodynamic derivatives with the least squares approach utilizing neural model outputs is discussed. Finally, the outcomes of this strategy are analyzed.

Chapter 5 describes the estimation of aerodynamic derivative coefficients using the output error method. After discussing the idea of the maximum likelihood approach, the output error method methodology for parameter estimation is detailed. Next, the use of the optimization technique Gauss-Newton is discussed. The points to be considered in the algorithm are mentioned, and the multiple-shooting modification is described. Finally, the outcomes of this procedure are investigated.

In Chapter 6, obtained results for three different methods for estimating aerodynamic parameters are examined in detail. The calculation of mean-squared errors employing these approaches on different data sets obtained through the 3-DOF model is described. Moreover, the performances of the three methods are evaluated and compared using mean-squared error values.

Chapter 7 summarizes the study with concluding remarks. Moreover, it also discusses potential research fields on this subject for future work.

# CHAPTER 2

# MATHEMATICAL MODEL

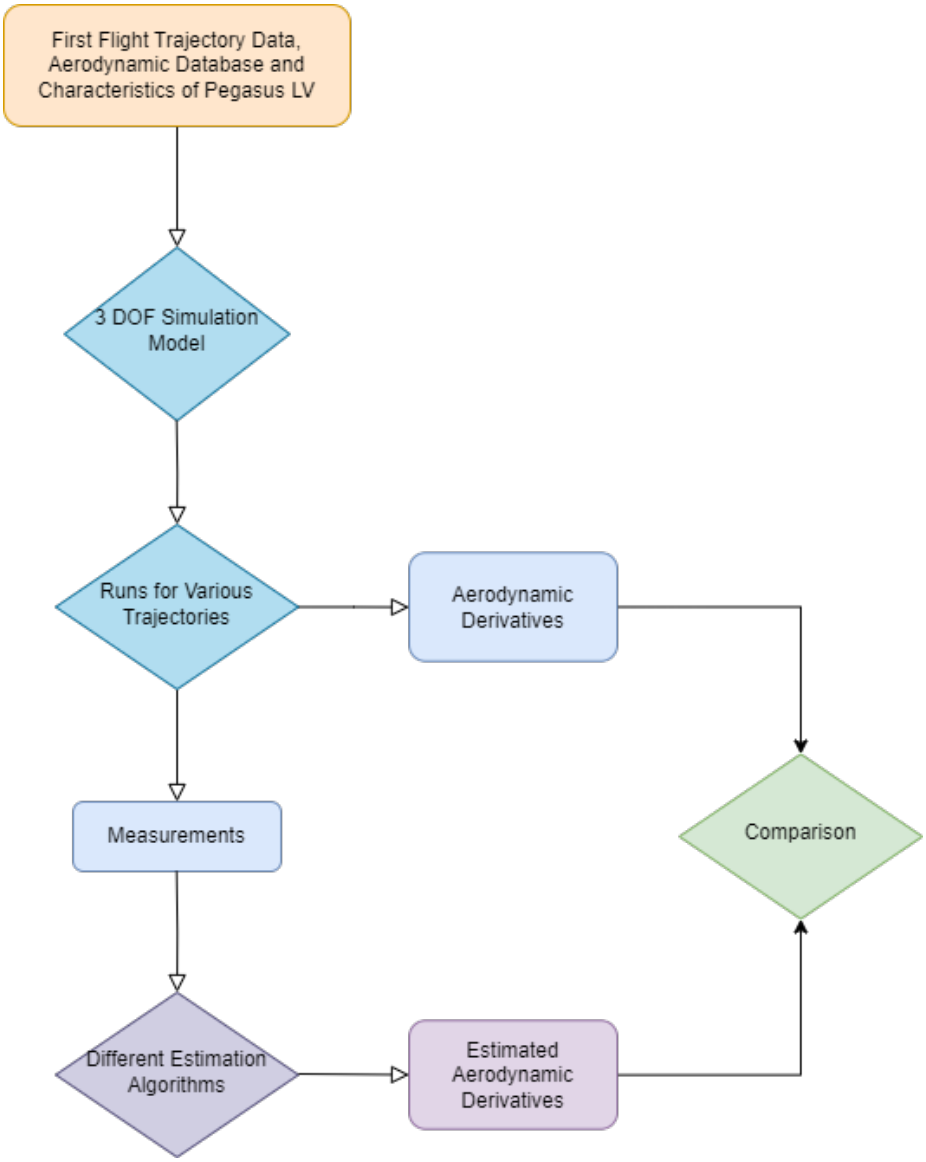



Figure 2.1: A Flow Chart for Aerodynamic Parameter Estimation Procedure

In Figure 2.1, a flow chart that represents the aerodynamic parameter estimation procedure for Pegasus LV is given. This flow chart can be explained by following steps:

1. The database, which contains Pegasus' first flight trajectory data, as well as the aerodynamic database and characteristic data, is gathered from the literature.

2. A three degrees of freedom model is built using this data.

3. Measurement data and aerodynamic derivatives for different trajectories are obtained using the 3-DOF model.

4. Three different approaches are used to estimate aerodynamic derivatives using the measurement data gathered from the 3 DOF model.

5. The estimated aerodynamic derivatives are compared with ones collected from the 3-DOF model as stated in Step 3.

This chapter will explain the Pegasus launch vehicle, the building of the 3-DOF model, and the collection of data by using this model.

## 2.1 Pegasus Air Launch Vehicle

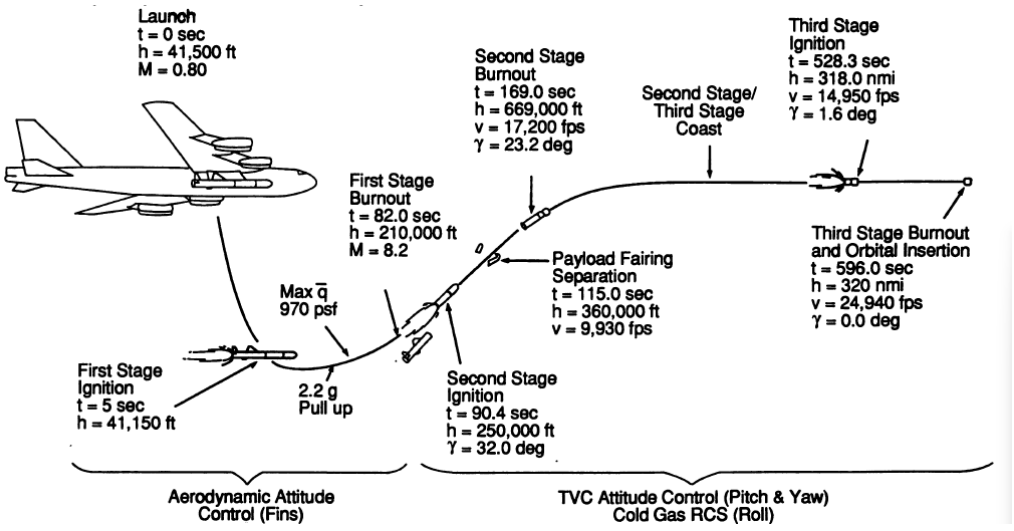



Figure 2.2: Pegasus Ascent Trajectory [33]

The Pegasus Launch Vehicle, a breakthrough in space launch technology, is a unique relationship between aeronautics and astronautics. The Pegasus, created by Northrop Grumman Innovation Systems (formerly Orbital Sciences Corporation), is notable for its unusual air launch technique, which has revolutionized conventional space launch paradigms. It is created with three stages and a winged booster to provide tiny satellites with quick and effective commercial launch services.
Unlike traditional vertical launches from Earth's surface, the Pegasus launch vehicle takes a different path. It is launched from a high altitude and carried underneath an aircraft, commonly the Lockheed L-1011 TriStar, until that starting altitude. The rocket can reach the desired orbit thanks to the air launch's beneficial starting position. In Figure 2.2, Pegasus' ascent trajectory can be seen.

The trajectory followed from the launch till the first stage separation is modeled in this thesis since the goal is to estimate the aerodynamic parameters for the first stage of Pegasus. The aerodynamic database utilized in the model is obtained from the [32]. Flight data until the initial separation, such as altitude, Mach, and angle of attack, are taken from the article [25], and data of acceleration change is collected from the article in [24]. [8] provides the geometric parameters which are shown in the Table 2.1.

Table 2.1: Physical Characteristics of the Pegasus [8]

| | |
|---|---|
| Wing Reference Area | 13.5 $m^2$ |
| Span | 6.7 $m$ |
| Mean Aerodynamic Chord | 2.48 $m$ |
| Root Chord | 3.69 $m$ |
| Tip Chord | 0.338 $m$ |
| Aspect Ratio | 3.33 |
| Wing Leading-Edge Sweep | 45 $^\circ$ |
| Wing Leading-Edge Radius | 2.54 $cm$ |
| Fuselage Length Overall | 15 $m$ |
| Fuselage Nominal Diameter | 1.27 $m$ |

The first flight trajectory of Pegasus' first stage is shared in the [25]. Changing altitude, Mach, and angle of attack throughout the trajectory until the first separation are given in Figures 2.3, 2.4, and 2.5 respectively. Also, one can see the changing of the dynamic pressure through the trajectory in Figure 2.6.

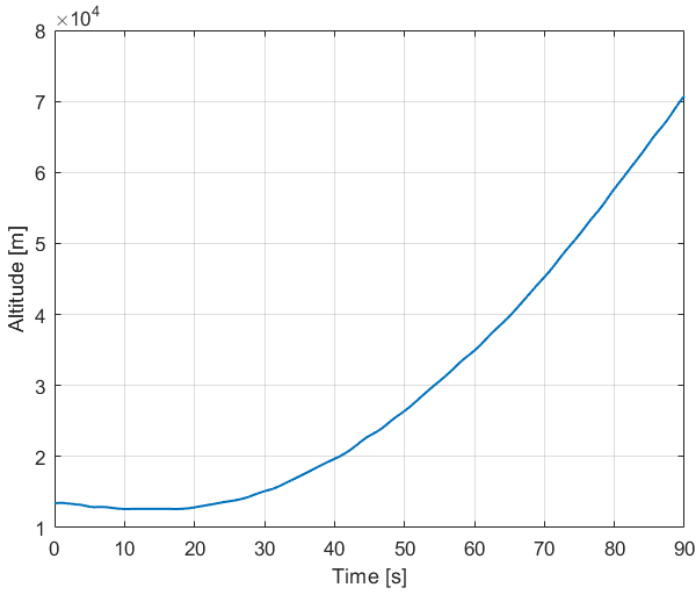



Figure 2.3: Pegasus's Pressure Altitude [25]

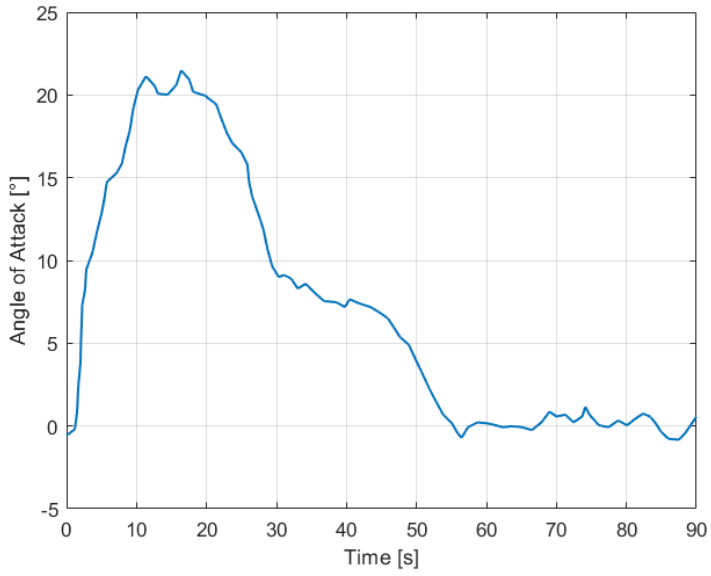



Figure 2.4: Pegasus's Angle of Attack [25]

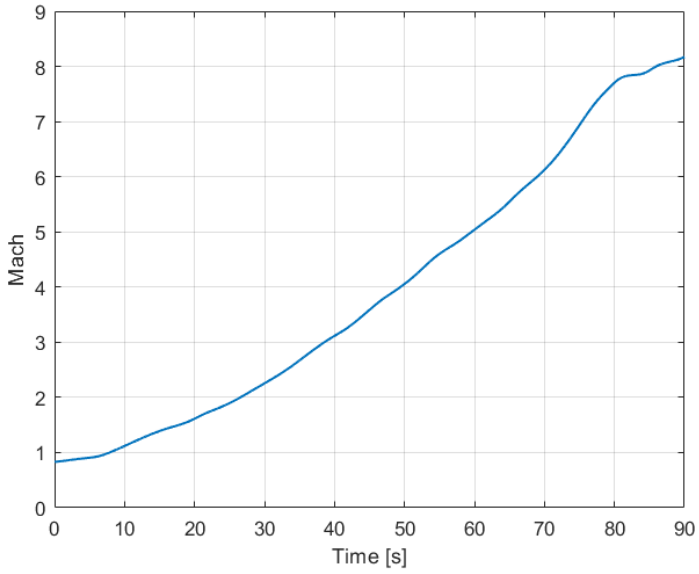



Figure 2.5: Pegasus's Free-stream Mach Number [25]

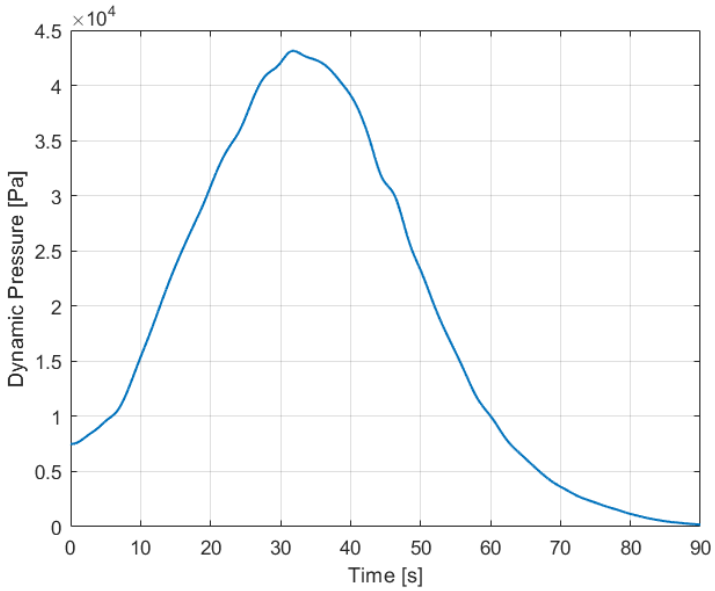



Figure 2.6: Pegasus's Dynamic Pressure

## 2.2 Reference Frames and Assumptions

In this work, two reference frame definitions are used to describe the attitude and motion of the Pegasus launch vehicle. The two reference frames are local inertial and body reference frames.

The inertial frame is set relative to the earth at the vehicle's released location from the aircraft, with the z-axis pointing toward the direction of the gravity vector, the x-axis pointing in the direction of the vehicle's trajectory, and the y-axis pointing to the right side of the vehicle's original orientation completing the orthogonal triad of unit axes. The inertial frame does not move with the vehicle and does not rotate.

The body frame is attached to the center of gravity of the launch vehicle. In the body frame, the x-axis points towards the nose of the vehicle, the z-axis points downwards in the plane of symmetry, and the y-axis completes the right-handed coordinate system. The body-fixed coordinate system moves and rotates with the vehicle.

The 3-2-1 Euler sequence is used to determine the rotation of the body frame from the fixed earth frame. Both reference frames are shown in Figure 2.7. X and z axes of both frames and pitch angle ($\theta$) are mainly focused, and a 3-DOF model is created for the longitudinal motion of the launch vehicle in this thesis study.

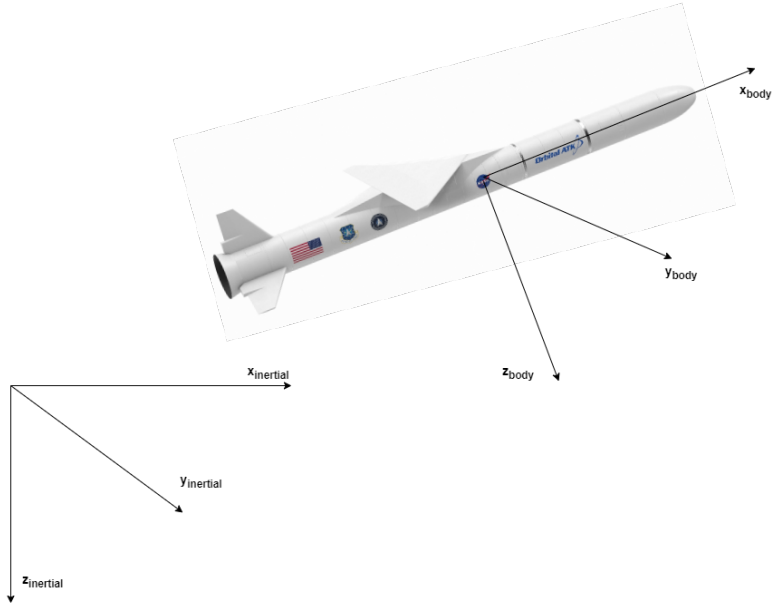



Figure 2.7: Inertial and Body Coordinate Frames [37]

The general assumptions used while preparing the 3-DOF model, which is used to collect data, are as follows:

- White noise is assumed to be the sole factor that affects the acceleration measurements through the IMU, and other IMU errors are considered negligible.

- It is assumed that there were no outside disturbances, such as wind or vibrations, in the simulation setting.

- The center of gravity and the center of pressure locations are assumed to vary only along the x-body axis.

- It is considered that the $C_{L0}$, $C_{D0}$, and $C_{M0}$ coefficients are pretty small because of the symmetric design of the launch vehicle. Hence, the aerodynamic force and moment equations were developed without considering these coefficients.

- Open-loop control surface commands are fed into the 3-DOF model using the recorded $\delta_e$ commands from Pegasus's first flight data.

## 2.3 Environment Model

To find atmospheric parameters, the 'atmosisa' function of MATLAB is used in the environment model. That function is based on The International Standart Atmosphere model, which is explained in [38]. This function gives temperature (T), speed of sound (a), pressure (P), and air density ($\rho$) according to the given altitude (h) input.
In that block, sea level pressure and dynamic pressure values are calculated. Sea level pressure is the output of the atmosphere model at sea level altitude, and it is used in the propulsion model to obtain net thrust from the vacuum thrust. Moreover, dynamic pressure is calculated by using the velocity and air density as follows:

$$Q = \frac{1}{2}\rho V^2 \tag{2.1}$$

## 2.4 Aerodynamic Model

The aerodynamic model of the Pegasus Launch Vehicle is explained in [32]. Longitudinal aerodynamic variables are taken from this reference. Aerodynamic variables $C_L$, $C_D$, $C_M$ and their derivatives $C_{L_\alpha}$, $C_{L_{\delta_e}}$, $C_{D_\alpha}$, $C_{D_{\delta_e}}$, $C_{M_\alpha}$, $C_{M_{\delta_e}}$ are given as a function of angle of attack and Mach number. According to this article, Mach breakpoints are 1.6, 2, 2.96, 4.63, and 6, and angle of attack breakpoints are [-8:4:24]. Since these breakpoints do not include all Mach numbers and angle of attack values seen in the trajectory of the Pegasus, the aerodynamic database is expanded using the neural network toolbox of MATLAB.

First, a neural network model is defined with 15 hidden layer size using the 'feedforwardnet' function. Afterward, 70 percent of the given data is set for training, 15 percent for validation, and 15 percent for testing. The angle of attack breakpoints and Mach breakpoints are used as inputs and given aerodynamic parameters $C_L$, $C_D$, $C_M$, $C_{L_\alpha}$, $C_{L_{\delta_e}}$, $C_{D_\alpha}$, $C_{D_{\delta_e}}$, $C_{M_\alpha}$, $C_{M_{\delta_e}}$ are used as outputs while training the neural model. Expanded angle of attack and Mach breakpoints are decided as including maximum and minimum values in trajectory. Furthermore, these expanded breakpoints are given to the trained neural model as inputs, and the model outputs are a new aerodynamic database that includes the trajectory.

Table 2.2: Aerodynamic Database Breakpoints

| | |
|---|---|
| $\alpha[^\circ]$ | [-1.84 : 2 : 26.48] |
| Mach | [0.3 : 1 : 9.2] |

In the 3-DOF model $C_{L_\alpha}$, $C_{L_{\delta_e}}$, $C_{D_\alpha}$, $C_{D_{\delta_e}}$, $C_{M_\alpha}$, $C_{M_{\delta_e}}$ are used as look-up tables to get the aerodynamic variables ($C_L$, $C_D$, and $C_M$) at each step time.

$$C_L = C_{L_\alpha}\alpha + C_{L_{\delta_e}}\delta_e \tag{2.2}$$

$$C_D = C_{D_\alpha}\alpha + C_{D_{\delta_e}}\delta_e \tag{2.3}$$

$$C_M = C_{M_\alpha}\alpha + C_{M_{\delta_e}}\delta_e \tag{2.4}$$

Moreover, forces on the x and z axes of the body frame $C_x$ and $C_z$ are calculated by

using the angle of attack. Body forces, lift and drag forces acting on the vehicle are shown in figure 2.8.

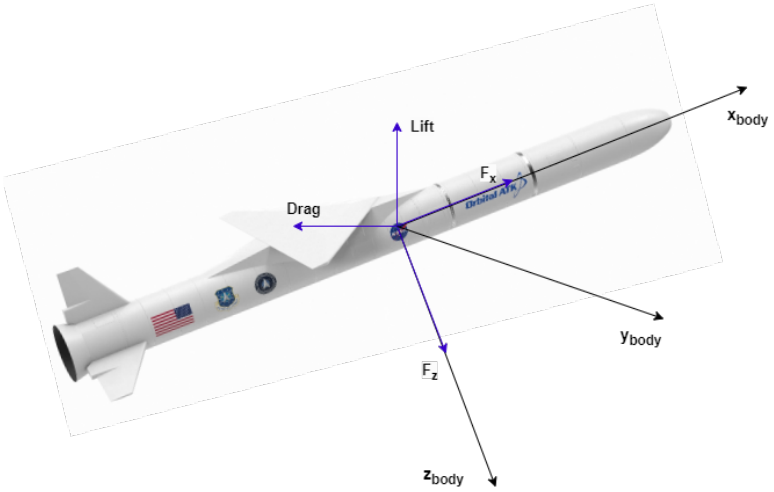



Figure 2.8: Forces Acting on the Body [37]

$$C_{Fx} = C_L \sin\alpha - C_D \cos\alpha \tag{2.5}$$

$$C_{Fz} = -C_L \cos\alpha - C_D \sin\alpha \tag{2.6}$$

## 2.5 Forces and Moments

### 2.5.1 Aerodynamic Forces and Moments

As mentioned in the Section 2.4, force and moment coefficients in the body frame are calculated in the aerodynamic model block. After these coefficients are obtained, forces and moments are dimensionalized and moved to the center of gravity from the center of pressure. Aerodynamic forces can be represented with the following equations;

$$F_x = QS_{ref}C_{Fx} \tag{2.7}$$

$$F_z = QS_{ref}C_{Fz} \tag{2.8}$$

Force on the z-axis which acts on the center of pressure will create a pitch moment around the center of gravity. Hence, the total aerodynamic pitch moment around the y-axis can be represented as follows;

$$M_y = QS_{ref}\bar{c}C_{My} + F_x(x_{cg} - x_{cp}) \tag{2.9}$$

In the above equations, $Q$ is dynamic pressure, which is shown in Section 2.3, $S_{ref}$ is wing reference area, $\bar{c}$ is a mean aerodynamic chord. Calculations of the center of gravity and the center of pressure will be explained in Section 2.7.

### 2.5.2 Thrust Force

The source of thrust is the ORION 50S for the first stage of the Pegasus launch vehicle. Geometrical and performance characteristics and vacuum thrust profile belonging to ORION 50S are given in [27]. The vacuum thrust profile of ORION 50S can be seen in Figure 2.9.

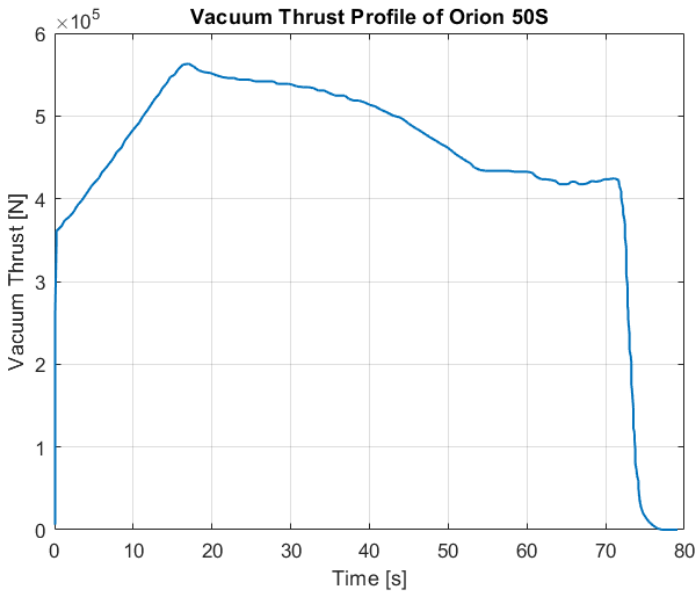



Figure 2.9: Vacuum Thrust Versus Time [27]

The net thrust force is calculated from the vacuum thrust by using ambient pressure ($P$) and sea level pressure ($P_0$).

$$T_{net} = T_{vacuum} + A(P_0 - P) \tag{2.10}$$

where $A$ is the nozzle exit area.

$$A = \frac{\pi d^2}{4} \tag{2.11}$$

where $d$ is the nozzle exit cone diameter and it's value is 1.42 meters, [27].

Pegasus launch vehicle has no thrust vector controller for the first stage, so the net thrust force acts in the negative direction of the x-axis of the body frame.

### 2.5.3 Gravitational Acceleration

Newton's law of universal gravitation is used for the gravity model. According to Newton's law, gravitational acceleration can be represented as the following equation, and the variables used in that equation are given in the Table 2.3.

$$g = \frac{GM_e}{(h + R_e)^2} \tag{2.12}$$

Table 2.3: Variables in Gravitational Acceleration Equation

| | | |
|---|---|---|
| G | Universal Gravitational Constant $[m^3 kg^{-1} s^{-2}]$ | $6.67428 \times 10^{-11}$ |
| $M_e$ | Mass of Earth $[kg]$ | $5.9722 \times 10^{24}$ |
| $R_e$ | Radius of Earth $[km]$ | 6378.1 |

The gravity force acts in the positive direction of the z-axis of the inertial frame. Hence, it should be transformed from the inertial frame to the body frame and it can be basically transformed by using the pitch angle.

$$g_{x_{body}} = -g \sin\theta \tag{2.13}$$

$$g_{z_{body}} = g \cos\theta \tag{2.14}$$

## 2.6 Equations of Motion

To assist in deriving equations of motion, references [26] and [28] are consulted.

### 2.6.1 Translational Dynamics

The translational motion of the center of gravity can be explained by Newton's second law.

$$\sum \vec{F}_B = m\vec{a}_B \tag{2.15}$$

The $\vec{F}_B$ represents the total force, which includes aerodynamic, propulsion, and gravity forces. The acceleration with respect to an inertial frame can be written in the body axis using the Coriolis theorem.

$$\vec{a}_B^I = \begin{bmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{bmatrix} + \begin{bmatrix} p \\ q \\ r \end{bmatrix} \times \begin{bmatrix} u \\ v \\ w \end{bmatrix} \tag{2.16}$$

Translational dynamics equations can be written explicitly as follows;

$$(F_x)_{aero} - F_{thrust} - mg\sin\theta = m(\dot{u} + qw - rv) \tag{2.17}$$

$$(F_y)_{aero} + mg\cos\theta\sin\phi = m(\dot{v} + ru - pw) \tag{2.18}$$

$$(F_z)_{aero} + mgcos\theta\cos\phi = m(\dot{w} + pv - qu) \tag{2.19}$$

$$\dot{u} = rv - qw + \frac{(F_x)_{aero} - F_{thrust}}{m} - g\sin\theta \tag{2.20}$$

$$\dot{v} = pw - ru + \frac{(F_y)_{aero}}{m} + g\cos\theta\sin\phi \tag{2.21}$$

$$\dot{w} = qu - pv + \frac{(F_z)_{aero}}{m} + gcos\theta\cos\phi \tag{2.22}$$

Since the aerodynamic derivative coefficients in the pitch axis will be estimated in this thesis, only the longitudinal ones of these equations have been used, with the lateral parameters considered zero.

$$\dot{u} = -qw + \frac{(F_x)_{aero} - F_{thrust}}{m} - g\sin\theta \tag{2.23}$$

$$\dot{w} = qu + \frac{(F_z)_{aero}}{m} + gcos\theta \tag{2.24}$$

### 2.6.2 Rotational Dynamics

Similarly, in the translational dynamics part, the rotational dynamics of the system are expressed by Coriolis's theorem.

$$\sum \vec{G}_B = \dot{\vec{H}}_B + \vec{\omega}_B \times \vec{H}_B \tag{2.25}$$

$$\vec{H}_B = I_B \vec{\omega}_B \tag{2.26}$$

$$\dot{\vec{H}}_B = \frac{d}{dt}(I_B \vec{\omega}_B) = \dot{I}_B \vec{\omega}_B + I_B \dot{\vec{\omega}}_B = I_B \dot{\vec{\omega}}_B \tag{2.27}$$

$$\vec{\omega}_B = \begin{bmatrix} p \\ q \\ r \end{bmatrix}, \dot{\vec{\omega}}_B = \begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} \tag{2.28}$$

$$\vec{G}_B = \begin{bmatrix} M_x \\ M_y \\ M_z \end{bmatrix} \tag{2.29}$$

where $\vec{G}_B$ represents external moments around the center of gravity, $I_B$ represents the inertia matrix and $\vec{\omega}_B$ is the angular rates in the body-fixed coordinate system.

$$\begin{bmatrix} M_x \\ M_y \\ M_z \end{bmatrix} = I_B \begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} + \begin{bmatrix} p \\ q \\ r \end{bmatrix} \times \left( I_B \begin{bmatrix} p \\ q \\ r \end{bmatrix} \right) \tag{2.30}$$

Cross inertial terms $I_{xy}$ and $I_{yz}$ are assumed as zero since the center of the body frame is the plane of symmetry, $I_{xz}$ is assumed to be minor, and it is neglected. Since the aerodynamic derivative coefficients in the pitch axis will be estimated in this thesis, only the pitch moment equation has been used. It can be written explicitly as follows;

$$M_y = I_{yy}\dot{q} + \dot{I}_{yy}q - I_{xz}(r^2 - p^2) - (I_{zz} - I_{xx})rp \tag{2.31}$$

Furthermore, the lateral parameters are considered to be zero, and the equation gets simpler.

$$M_y = I_{yy}\dot{q} + \dot{I}_{yy}q \tag{2.32}$$

The moment that comes from the above equation is around the center of pressure. All forces and moments are moved to the center of gravity from the center of pressure.

While forces remain the same, an extra moment comes from the movement of the force on the z-axis($F_z$).

$$M_{y_{cg}} = I_{yy}\dot{q} + \dot{I}_{yy}q - F_z(x_{cg} - x_{cp}) \tag{2.33}$$

## 2.7 Mass, Inertia, Center of Gravity Model

The gross weight of Pegasus is 13405 kg, and the burnout mass is 952 kg. The first ignition time is 5 seconds after Pegasus is left from the aircraft, and the burnout time of the engine is approximately 75.3 seconds. While the change in the mass is calculated, fuel change is assumed to be linear. According to this assumption, changing mass during the first stage has a profile like shown in Figure 2.10.

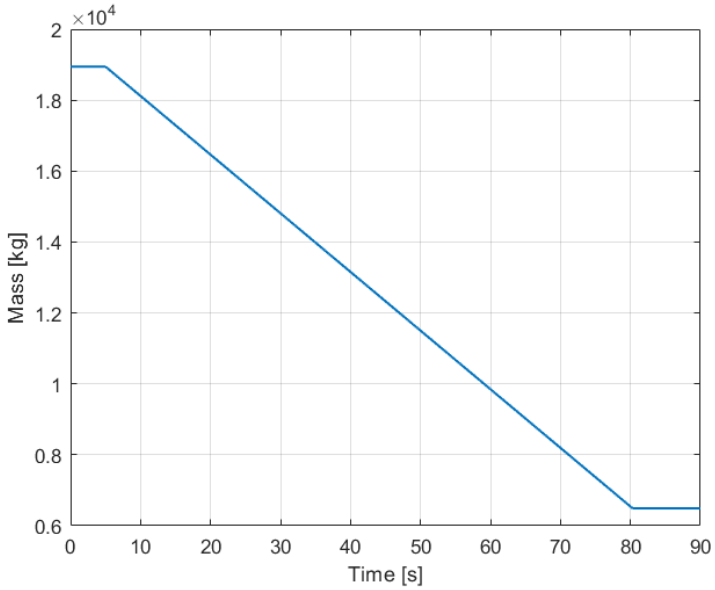



Figure 2.10: Mass Change Versus Time

According to reference, [22], the center of pressure location moves from approximately 31 percent to 29 percent of the total length. In the thesis in [17], the center of gravity locations and weights of each part of an air-launch vehicle very similar to the Pegasus are given. The inertia and the center of gravity of the Pegasus are calculated approximately with these pieces of information and the weight information of

the Pegasus, by proportioning the weight and length of the vehicle given in the article to those of the Pegasus.

$$x_{cg} = \frac{\sum(m_i x_{cg_i})}{\sum(m_i)} \tag{2.34}$$

$$I_{yy} = \sum m_i (x_{cg} - x_{cg_i})^2 \tag{2.35}$$

where $x_{cg}$ is the center of gravity of the vehicle, $x_{cg_i}$ is is the center of gravity of each ($i^{th}$) part and $m_i$ is mass of each part.

Figure 2.11 shows the time-dependent change graph of the moment of inertia along the trajectory that was constructed for a nominal trajectory using the 3-DOF model developed for the Pegasus launch vehicle. Similarly, Figure 2.12 represents changes in the center of gravity and the center of pressure throughout time.

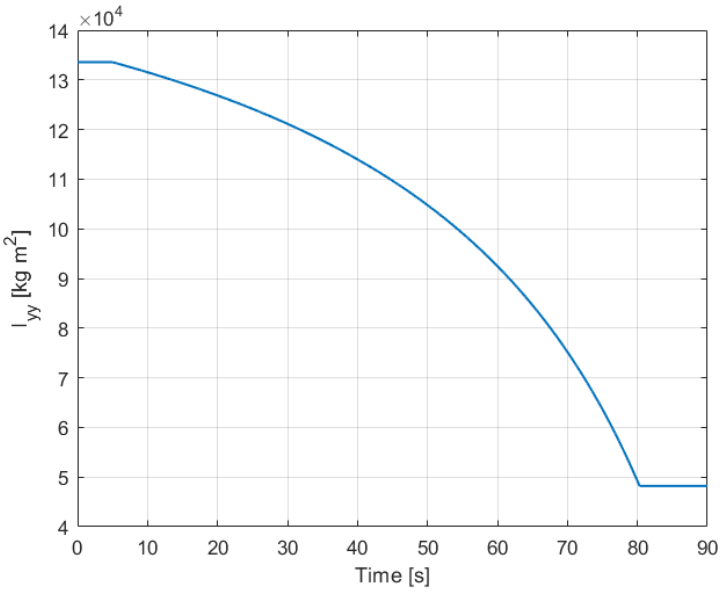



Figure 2.11: Moment of Inertia Change Versus Time

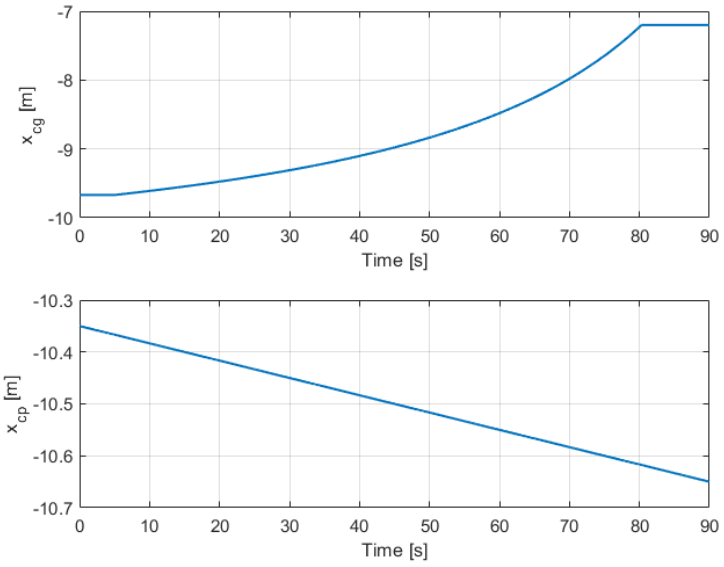



Figure 2.12: Center of Gravity and Center of Pressure Change Versus Time

# CHAPTER 3

# AERODYNAMIC PARAMETER ESTIMATION USING THE LEAST SQUARES METHOD

## 3.1 Introduction

In the literature, the least squares approach is referred to as classical regression analysis. Regression analysis is the most basic approach for estimating parameters. In regression methodology applications, aerodynamic force and moment measurements are necessary. These variables cannot be directly measured but may be calculated using linear and angular accelerations. This approach estimates three aerodynamic forces and three moments operating on the vehicle individually. It may be an advantage that the least squares approach is simple to implement and may perform estimations at once. However, the least squares approach may produce biased, inconsistent, or inefficient estimates due to measurement inaccuracies in the independent variables.

## 3.2 Methodology

If there is a linear system equation $Ax = b$ with the matrix A in size $m \times n$, the vector b in length $m$, and $m > n$, the equation has fewer unknowns than equations. Hence, systems like this mostly do not have a solution. To solve such a problem, one may need to solve the equation $Ax - b = 0$ as possible by finding the vector $x$ with size $n$. In the literature, the solution to this problem is defining a residual vector $r = Ax - b$ and minimizing $r$ in the 2-norm. Furthermore, this gives rise to the Least Squares approach according to [36].

For the purpose of linking independent variables to dependent variables, the modeling

equation can be represented as follows;

$$y = X\theta \tag{3.1}$$

where $y$ represents the dependent variable, $X$ represents the vector independent variables in size of $(N \times n_p)$, and $\theta$ is the unknown model parameters that characterize the effect of each term on the dependent variable y. Size of unknown parameter vector $\theta$ is $(n_p \times 1)$. Also, the regression equation can be represented as follows;

$$z = X\theta + \nu \tag{3.2}$$

where $z$ is the measurements vector of size $(N \times 1)$, and $\nu$ is the measurement errors vector in size $(N \times 1)$.

The regression equation is also called the measurement equation. The least squares model assumes that $\nu$ has a zero mean and is unrelated to a constant variability even though there is no likelihood information for either $\theta$ or $\nu$.

$$E(\nu) = 0, \quad E(\nu\nu^T) = \sigma^2 I \tag{3.3}$$

In a least-squares sense, reducing the sum of squared differences between the model and the measurements yields an optimal estimator of $\theta$. For this purpose, the cost function can be written as below.

$$J(\theta) = \frac{1}{2}(z - X\theta)^T(z - X\theta) \tag{3.4}$$

To minimize the cost function $J(\theta)$, the parameter estimate $\hat{\theta}$ needs to meet certain requirements.

$$\frac{\partial J}{\partial \theta} = -X^T z + X^T X\hat{\theta} = 0 \tag{3.5}$$

or

$$X^T X\hat{\theta} = X^T z \tag{3.6}$$

or

$$X^T(z - X\hat{\theta}) = 0 \tag{3.7}$$

The above equations are called the normal equations. The formula for the least-squares estimator, also known as the ordinary least-squares estimator, is obtained by solving these equations for the unknown parameter vector $\theta$.

$$\hat{\theta} = (X^T X)^{-1} X^T z \tag{3.8}$$

For the above equation to be solvable, in other words, to avoid mathematical singularity, the $X$ matrix of size $(N \times n_p)$ and the matrix $(X^T X)$ must be full rank.

$$rank(X) = N \qquad rank(X^T X) = N \tag{3.9}$$

### 3.3 Input Design Study for The Least Squares

An initial investigation is conducted to examine the influence of the control surface commands on the performance of the least squares algorithm before the estimation of Pegasus' aerodynamic parameters using the least squares method. To begin the input design study, data collected on a linear and basic 3-DOF model with varying wing command inputs is evaluated to better grasp the operation notion of the least squares technique. Only the force derivatives on the body z-axis ($C_{Fz_\alpha}$, $C_{Fz_{\delta_e}}$) and pitch moment derivatives ($C_{My_\alpha}$, $C_{My_{\delta_e}}$) are estimated by the least square method. Hence, the model equations are written as follows;

$$C_{Fz} = C_{Fz_0} + C_{Fz_\alpha}\alpha + C_{Fz_{\delta_e}}\delta_e \tag{3.10}$$

$$C_{My} = C_{My_0} + C_{My_\alpha}\alpha + C_{My_{\delta_e}}\delta_e \tag{3.11}$$

The preceding equations are then rewritten in the manner $\mathrm{y} = \mathrm{Ax}$.

$$C_{Fz}(k) = \begin{bmatrix} 1 & \alpha(k) & \delta_e(k) \end{bmatrix} \begin{bmatrix} C_{Fz_0} \\ C_{Fz_\alpha} \\ C_{Fz_{\delta_e}} \end{bmatrix} \tag{3.12}$$

$$C_{My}(k) = \begin{bmatrix} 1 & \alpha(k) & \delta_e(k) \end{bmatrix} \begin{bmatrix} C_{My_0} \\ C_{My_\alpha} \\ C_{My_{\delta_e}} \end{bmatrix} \tag{3.13}$$

where $k$ represents the step.

The force and moment coefficients $C_{F_z}$ and $C_{M_y}$ in the equation's left-hand matrix are identified using acceleration ($a_z$) and angular velocity change ($q$).
A frequency is selected for the least squares algorithm, and the algorithm packs the data and generates an estimate. For example, the simulation runs at 200 Hz and records data at that frequency, while the acceleration, angular velocity change, angle of attack, and control surface angle recorded from the simulation are fed into the least squares method, divided by the 1 Hz step into packets with time, and an estimate is made for each packet. The algorithm checks if the number of independent columns of the $(A)$ and $(A^T A)$ matrices is 3 with a tolerance of $1 \times 10^{-2}$; if this criterion is not satisfied, the estimation is not updated, and the values obtained in the previous step are used.

### 3.3.1 Estimation of Aerodynamic Derivatives for Linear Model

Initially, a basic 3-DOF model is developed to evaluate the estimation technique using the least squares method. The angle of attack and pitch angular velocity are the states of the model; control surface command is its input, and vertical acceleration is its output. The generated model's space-state equations are as follows. Moreover, some definitions that are used to simplify the following equations are given in Table 3.1.

$$\dot{\alpha} = \frac{Z_\alpha}{U}\alpha + \left(\frac{Z_q}{U} + 1\right) q + \frac{Z_{\delta_e}}{U}\delta_e \tag{3.14}$$

$$\dot{q} = M_\alpha \alpha + M_q q + M_{\delta_e}\delta_e \tag{3.15}$$

$$a_z = Z_\alpha \alpha + Z_q q + Z_{\delta_e}\delta_e \tag{3.16}$$

Since the dynamic derivatives ($Z_q$ and $M_q$) have infinitesimally small values, they are

Table 3.1: Definitions Used to Simplify State-Space Equations

| | |
|---|---|
| $\mathrm{Z_\alpha = \frac{QS_{ref}}{m} C_{z_\alpha}}$ | $\mathrm{M_\alpha = \frac{QS_{ref} l_{ref}}{I_{yy}} C_{m_\alpha}}$ |
| $\mathrm{Z_\delta = \frac{QS_{ref}}{m} C_{z_\delta}}$ | $\mathrm{M_\delta = \frac{QS_{ref} l_{ref}}{I_{yy}} C_{m_\delta}}$ |
| $\mathrm{Z_q = \frac{QS_{ref} l_{ref}}{2mU} C_{z_q}}$ | $\mathrm{M_q = \frac{QS_{ref} l_{ref}^2}{2UI_{yy}} C_{m_q}}$ |

assumed to be zero in this work. The foregoing are the state matrix, control matrix,

and output matrix of state-space equations.

$$A = \begin{bmatrix} \frac{Z_\alpha}{U} & 1 \\ M_\alpha & 0 \end{bmatrix}, B = \begin{bmatrix} \frac{Z_{\delta_e}}{U} \\ M_{\delta_e} \end{bmatrix} \tag{3.17}$$

$$C = \begin{bmatrix} Z_\alpha & 0 \end{bmatrix}, D = \begin{bmatrix} Z_{\delta_e} \end{bmatrix} \tag{3.18}$$

The 3-DOF model prepared in the Simulink environment is run at the desired time with a 5 ms time step, and the actual angle of attack, pitch angular velocity, pitch angular velocity change, and vertical acceleration are recorded. The estimation of the aerodynamic derivatives that are given as input in this model has then been studied using the outputs of this model as the inputs of the least squares method algorithm. This study is carried out initially with a constant Mach number, then with a linearly varying Mach number in the given time period. Besides that, the control surface command is provided to the model in various input forms such as sine, ramp, step, and pulse commands, and the effect of the control surface command on the least squares method results is examined. In Fig. 3.1, the ranks of matrices $A$ and $(A^T A)$, calculated through the algorithm for different control surface commands cases, are shown.

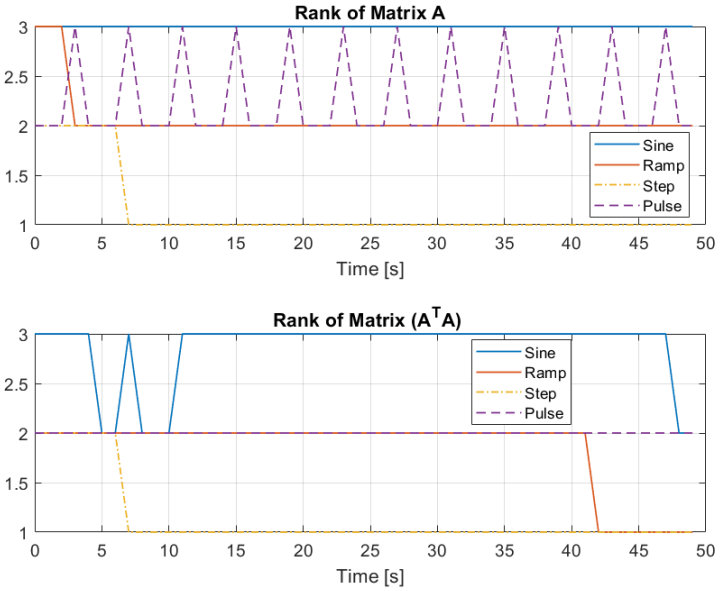



Figure 3.1: Ranks of $A$ and $(A^T A)$ Matrices for All Types of Control Surface Input

#### 3.3.1.1 Control Surface Command using Sine Wave Input and Fixed Mach Number

For the first example, the 3-DOF simulation is run for 50 seconds with a constant altitude and Mach inputs. The control surface command is supplied as a sine wave with a frequency of 2 Hz and an amplitude of -3°. The aerodynamic derivatives of $C_{Fz_\alpha}$, $C_{Fz_{\delta_e}}$, $C_{My_\alpha}$,$C_{My_{\delta_e}}$ are estimated using the least squares method using the angle of attack, pitch angular velocity change and vertical acceleration recorded from the simulation. These values are then compared to those derived via linearization from the aerodynamic database. The aerodynamic derivative coefficients predicted by the least squares approach using 50 seconds of flight simulation data at a given Mach are very close to the expected values according to the database, as shown in Figs. 3.2 and 3.3 as blue and orange lines.

#### 3.3.1.2 Control Surface Command using Sine Wave Input and Variable Mach Number

This time, the 3-DOF model is run under the identical settings and control surface command as in the prior scenario, with the only change being that a variable Mach number is employed instead of a constant Mach. Estimated aerodynamic derivatives are compared with the values obtained from the database, as in the example above. Looking at red and violet lines on Figs. 3.2 and 3.3, we can conclude that the least squares approach successfully estimated the parameters in the changing Mach number case. Minor distortions may be detected at the 5th and 10th seconds. The blue line in the second plot of Fig. 3.1 shows that the number of independent columns of the matrix $(A^T A)$ declines to 2 in the relevant seconds; therefore, the predictive value achieved in the previous time step remains constant until the rank returns to 3.

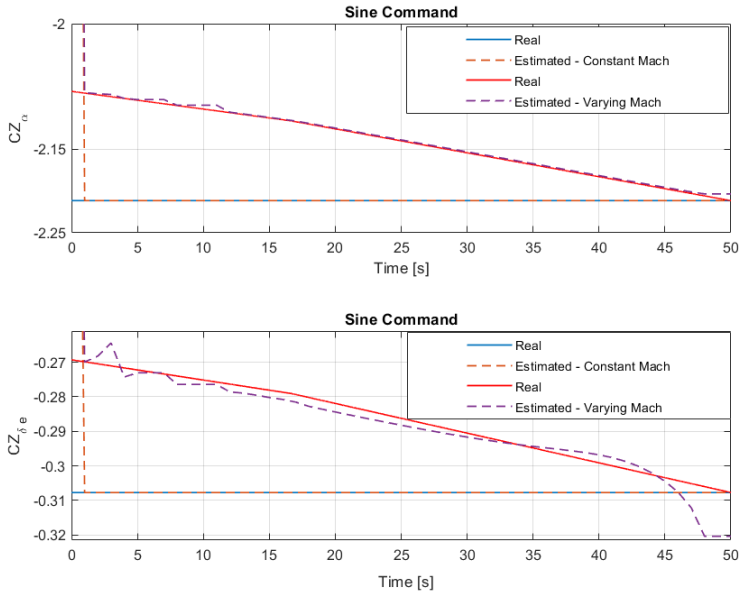



Figure 3.2: Estimation Results for Aerodynamic Force Derivatives for Constant and Varying Mach

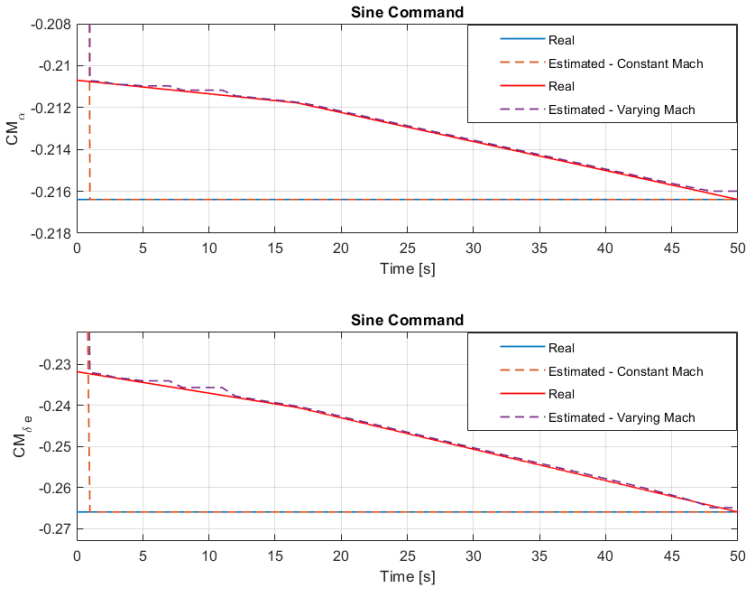



Figure 3.3: Estimation Results for Aerodynamic Moment Derivatives for Constant and Varying Mach

#### 3.3.1.3 Control Surface Command using Ramp Input

This time, the control surface command is fed to the system as a ramp command with a slope of 0.05°. When the system receives the control surface instruction as a ramp input, the columns of the $(A^T A)$ matrix are linked since the angle of attack and wing angle vary linearly and their ratios to each other are essentially constant throughout the expected time intervals. As a result, the estimation results are incorrect. The aerodynamic derivative estimation values are represented with the orange line in Figs. 3.4 and 3.5, and they seem to be zero since the algorithm's beginning value is zero, and the estimation results always remained at the first value because the rank of the $(A^T A)$ matrix is never three as shown with the orange line in Fig. 3.1.

#### 3.3.1.4 Control Surface Command using Step Input

In this example, a 5° constant wing command angle is fed into the system as input. Yellow lines in Figs. 3.4 and 3.5 show that the least squares estimation algorithm does not give proper results when step input is used. The yellow line in Fig. 3.1 illustrates the number of independent columns in the $A$ and $(A^T A)$ matrices for step input; based on these graphs, it is clear that the matrices are not full rank. As in the ramp input example, the $(A^T A)$ matrix is not invertible since the control surface command and angle of attack ratio remain stable. As a result, the estimation results are not updated.

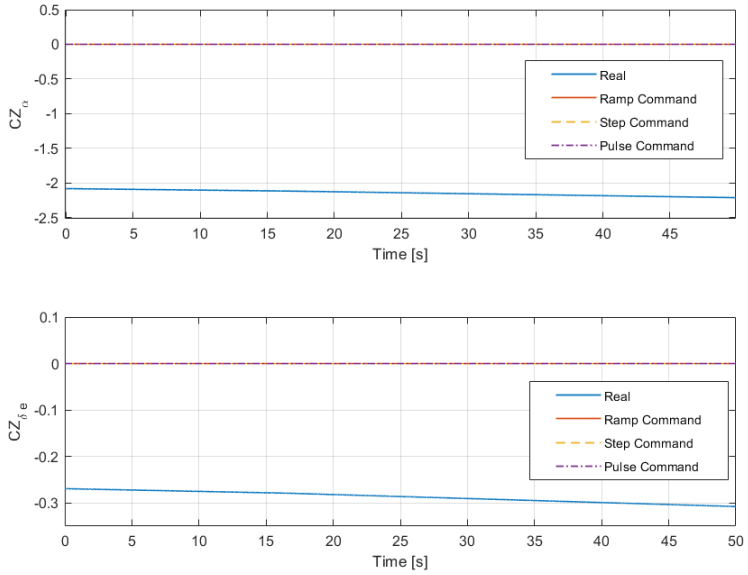



Figure 3.4: Estimation Results for Aerodynamic Force Derivatives for Varying Mach - Ramp, Step, Pulse Input

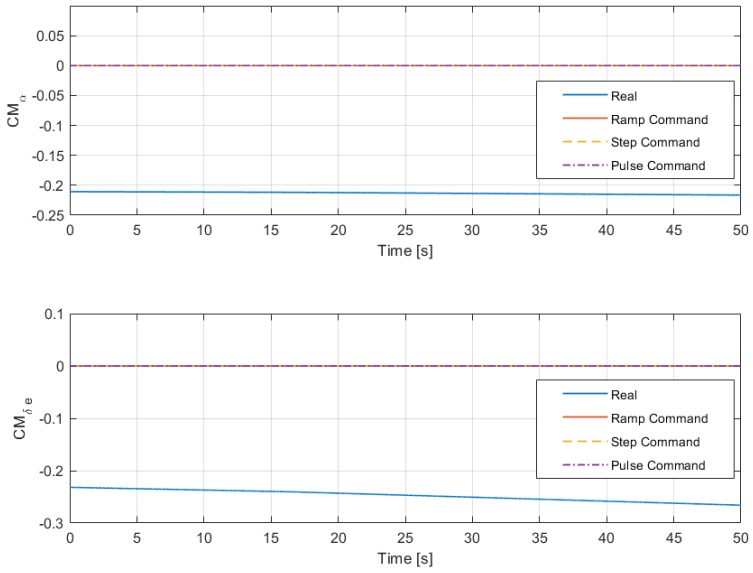



Figure 3.5: Estimation Results for Aerodynamic Moment Derivatives for Varying Mach - Ramp, Step, Pulse Input

#### 3.3.1.5 Control Surface Command using Pulse Input

This time, the control surface command is sent to the system as an 8-second pulse command with an amplitude of 5° and a bias of -1°. The violet dashed line in Fig. 3.1 shows that the rank of the A matrix is three every 4 seconds, but because the rank of the $(A^T A)$ matrix is always 2, the estimation results, which are represented with violet dashed lines in Figs. 3.4 and 3.5, are 0 because the criteria in the algorithm are not satisfied. The condition in the algorithm is then altered to 3 for the rank of matrix $A$ and 2 for the rank of matrix $(A^T A)$, and the least squares approach is tried again with the pulse command.

Figs. 3.6 and 3.7 show that the estimation algorithm gets results that are quite near to those obtained from the aerodynamic database, but because the $(A^T A)$ matrix is not full rank, obtaining its inverse is not a mathematically trustworthy process.

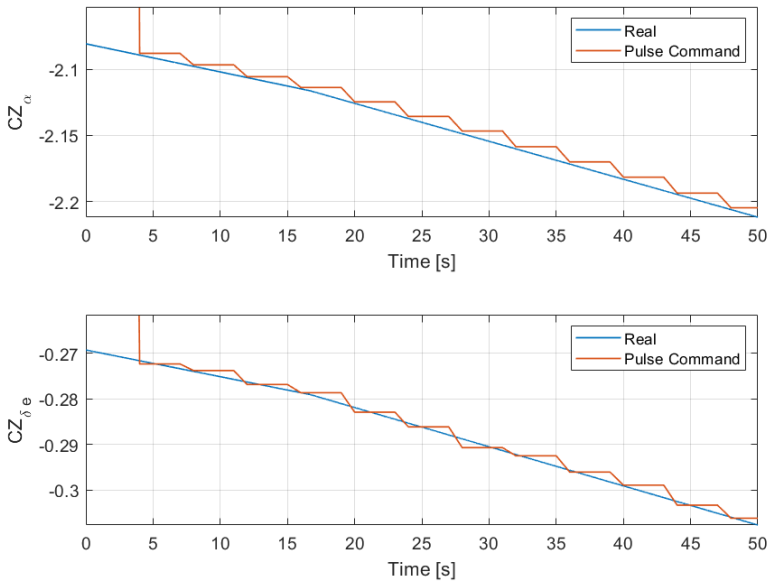



Figure 3.6: Estimation Results for Aerodynamic Force Derivatives for Varying Mach - Pulse Input

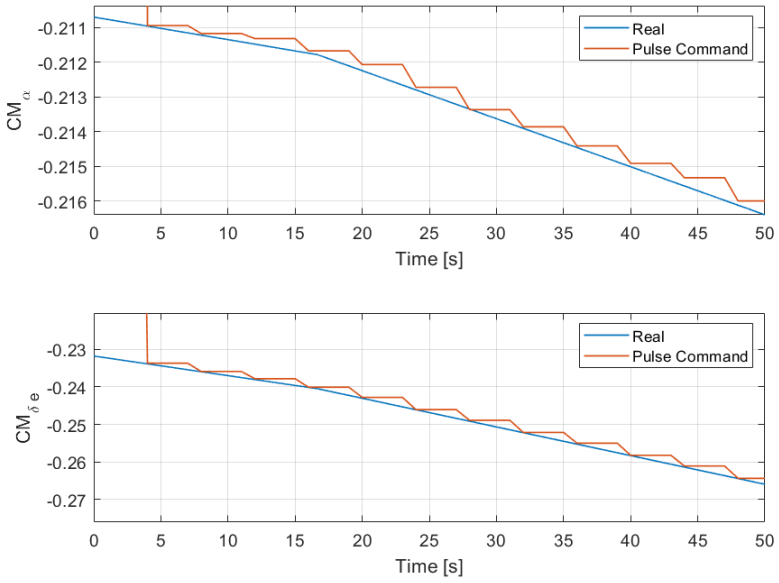



Figure 3.7: Estimation Results for Aerodynamic Moment Derivatives for Varying Mach - Pulse Input

From this work, it can be concluded that the wing command supplied to the system while recording flight data has a significant impact on the performance of the algorithm. When the angle of attack changes following wing commands such as sine or pulse inputs, the linear dependency between them decreases over time. The least-squares approach produces reliable answers in these conditions. If the angle of attack and control surface command angle does not change considerably in the predicted time packages, the columns and rows of the $(A^T A)$ matrix will be linearly coupled to each other. Hence, the least squares approach will not produce mathematically valid conclusions.

## 3.4 Pegasus's Aerodynamic Derivative Estimation by The Least Squares Method

The least-square algorithm inputs are obtained using the Pegasus 3-DOF model described in Chapter 2. The angle of attack, velocities, pitch angle, pitch rate from the equations of motion block, and acceleration from IMU are recorded. Moreover, the

control surface command is fed into the 3-DOF model as an open-loop command. In [22], recorded elevator deflections from the first flight of Pegasus are given. These elevator deflections from flight data are provided to the system in the 3-DOF model by adding pulses.

Getting started with the least squares estimation, the aerodynamic force and moment coefficient equations representing the longitudinal aerodynamic mathematical model can be expressed as follows.

$$C_L = C_{L_\alpha}\alpha + C_{L_{\delta_e}}\delta_e \tag{3.19}$$

$$C_D = C_{D_\alpha}\alpha + C_{D_{\delta_e}}\delta_e \tag{3.20}$$

$$C_{My} = C_{My_\alpha}\alpha + C_{My_{\delta_e}}\delta_e \tag{3.21}$$

The preceding equations are then rewritten in the manner $\mathrm{y} = \mathrm{Ax}$.

$$C_L(k) = \begin{bmatrix}\alpha(k) & \delta_e(k)\end{bmatrix}\begin{bmatrix}C_{L_\alpha} \\ C_{L_{\delta_e}}\end{bmatrix} \tag{3.22}$$

$$C_D(k) = \begin{bmatrix}\alpha(k) & \delta_e(k)\end{bmatrix}\begin{bmatrix}C_{D_\alpha} \\ C_{D_{\delta_e}}\end{bmatrix} \tag{3.23}$$

$$C_{My}(k) = \begin{bmatrix}\alpha(k) & \delta_e(k)\end{bmatrix}\begin{bmatrix}C_{My_\alpha} \\ C_{My_{\delta_e}}\end{bmatrix} \tag{3.24}$$

In the above equations, the second matrix on the right contains the estimated aerodynamic coefficient derivatives, the first matrix contains the angle of attack and the control surface angle, and the left side of the equations contains the measured aerodynamic force and moment coefficients. The aerodynamic coefficient derivatives are calculated for each time step $k$ by using the formula below.

$$x = (A^T A)^{-1} A^T y \tag{3.25}$$

The force and moment coefficients $C_L$, $C_D$ and $C_{M_y}$ in the equation's left-hand matrix cannot be determined directly in simulation or flight tests, but $C_{F_x}$, $C_{F_z}$ and $C_{M_y}$ can be calculated using acceleration ($a_x$), ($a_z$) and angular rate change ($\dot{q}$). Moreover, $C_L$

and $C_D$ can be obtained by translating $C_{F_x}$ and $C_{F_z}$ coefficients from the body axis coordinate system to the wind axis coordinate system. Additionally, when writing the force and moment equations, consideration is given to thrust force and gravitational acceleration.

$$(F_x)_{aero} = m(a_x + qw + g\sin\theta) - F_{thrust} \tag{3.26}$$

$$(F_z)_{aero} = m(a_z - qu - gcos\theta) \tag{3.27}$$

$$(M_y)_{aero} = I_{yy}\dot{q} + \dot{I}_{yy}q - F_z(x_{cg} - x_{cp}) \tag{3.28}$$

$$C_{F_x} = \frac{(F_x)_{aero}}{QS_{ref}} \tag{3.29}$$

$$C_{F_z} = \frac{(F_z)_{aero}}{QS_{ref}} \tag{3.30}$$

$$C_{M_y} = \frac{(M_y)_{aero}}{QS_{ref}l_{ref}} \tag{3.31}$$

$$C_L = C_{F_x}\sin\alpha - C_{F_z}\cos\alpha \tag{3.32}$$

$$C_D = -C_{F_x}\cos\alpha - C_{F_z}\sin\alpha \tag{3.33}$$

According to the results of the study in Section 3.3, the least-squares algorithm works with sine wave or pulse inputs. Hence, sine wave and pulse inputs are added to the control surface command of the Pegasus 3-DOF model, and the data sets are recorded. Then, least-squares estimation is applied to these two data sets to see the effect of these inputs on the algorithm's performance. By looking at equation 3.25, the matrix A includes $\alpha$ and $\delta_e$, so its rank should be 2 to prevent the mathematical singularity. In Figure3.8, it is seen that the $A$ and $(A^TA)$ matrices are full rank in both cases in the least-squares algorithm. For the flight data sets used in the rest of the study, pulses are inserted to control surface commands while creating data in the Pegasus model.

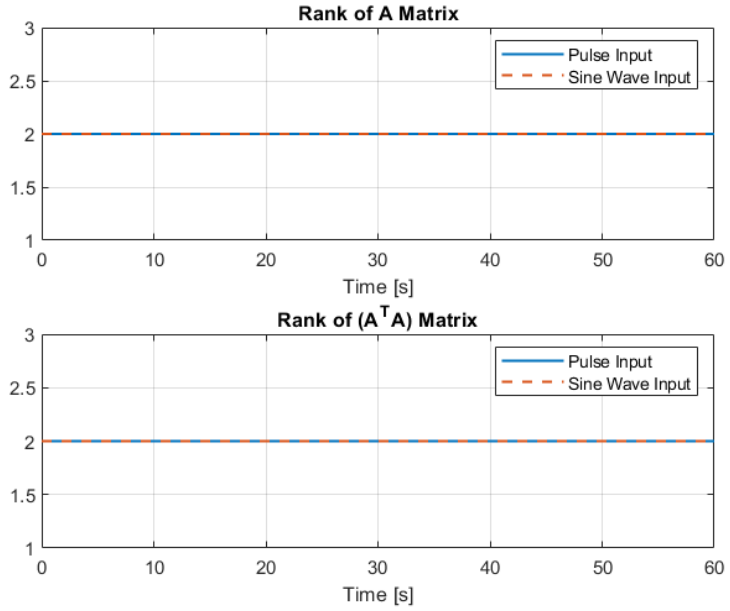



Figure 3.8: Ranks of $A$ and $(A^T A)$ Matrices for Estimation of Data Sets Recorded By Adding Sine Wave and Pulse Inputs to The $\delta_e$

## 3.5 Results

The aerodynamic parameter estimate results produced using the least squares approach utilizing one of the data sets acquired from the 3-DOF model are given in this section. To reduce the linearity between the angle of attack and control surface command, as is described in the Section 3.3, pulse input is introduced to the control surface instructions while the 3-DOF model is running. Data is gathered by running the model at a step time of 1000 Hz, or 0.001 seconds. Then, the prediction algorithm is performed after dividing the recorded data set into 2 Hz windows. The prediction algorithm will perform poorly if its working frequency is high since there is not much information in a window. On the other hand, if the frequency is too low, there will be too much data in a window, which will lead the forecasts to be far from real and the dynamic variations to be missed. The estimating technique in this study is tested at various frequencies, and it is discovered via trial and error that 2 Hz is the ideal frequency.

Figures 3.9, 3.10, and 3.11 show the results of the derivatives of lift, drag, and moment

coefficients with respect to the angle of attack and elevator commands, respectively.

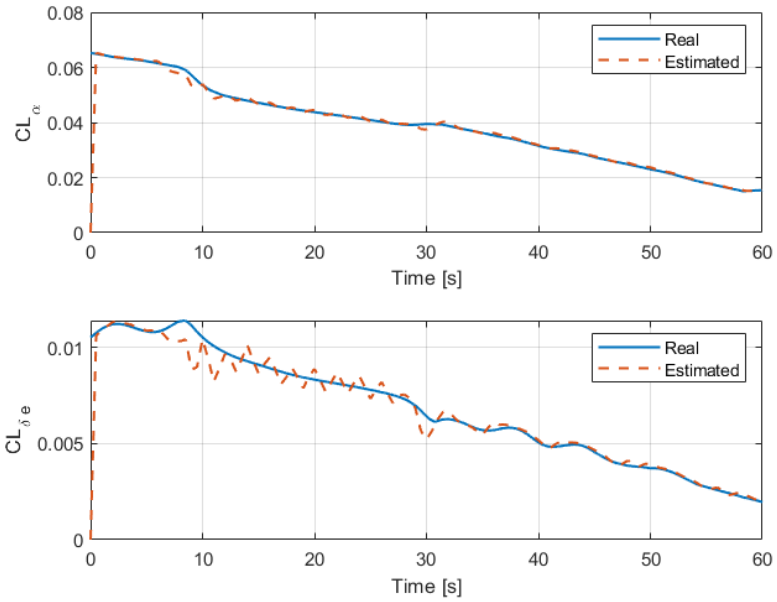



Figure 3.9: Estimation Results of Lift Coefficient Derivatives

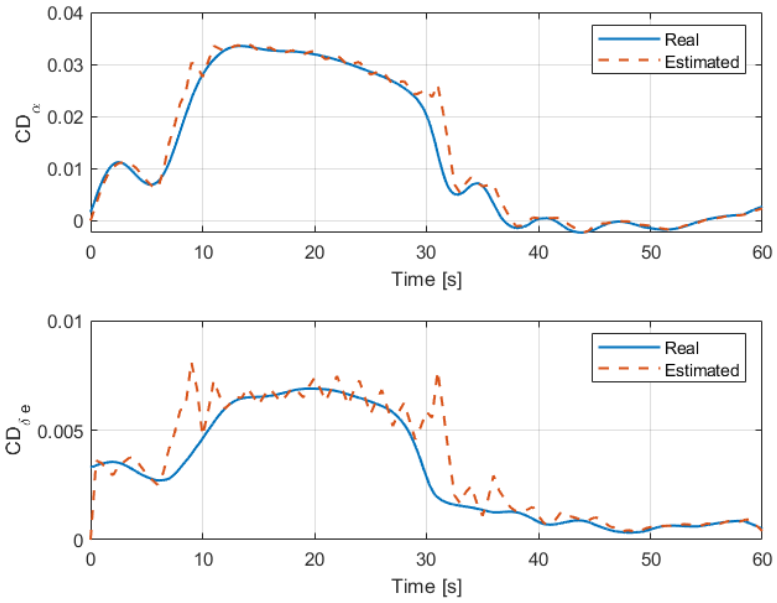



Figure 3.10: Estimation Results of Drag Coefficient Derivatives

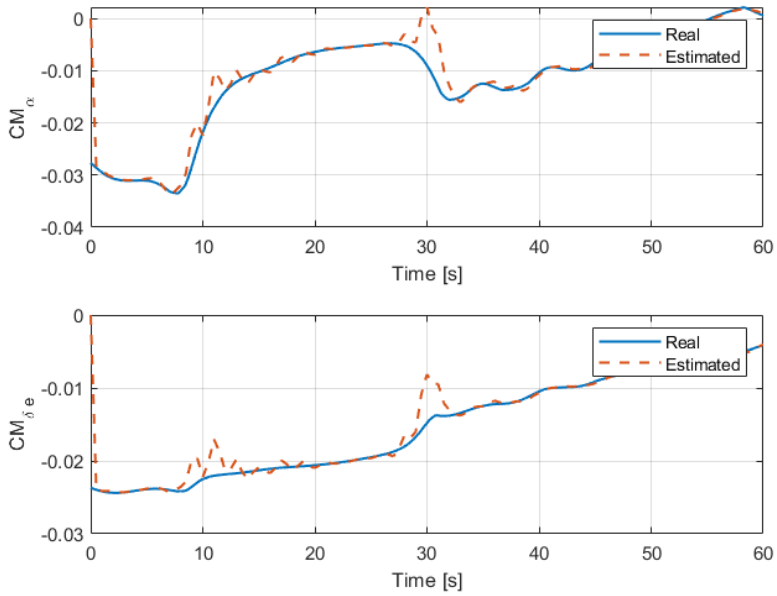



Figure 3.11: Estimation Results of Pitch Moment Coefficient Derivatives

By looking at the above figures, one can see that the estimated aerodynamic derivatives are very close to real ones. Hence, it can be said that the least-square algorithm works well for this case.

When looking at figures 3.9, 3.10, and 3.11, it is noticeable that the predicted values surrounding the 10th and 30th seconds deviate slightly from the actual values relative to the other values. This may be explained by the fact that the angles of attack and control surface change more quickly over the relevant time periods, as seen in Figure 3.12.

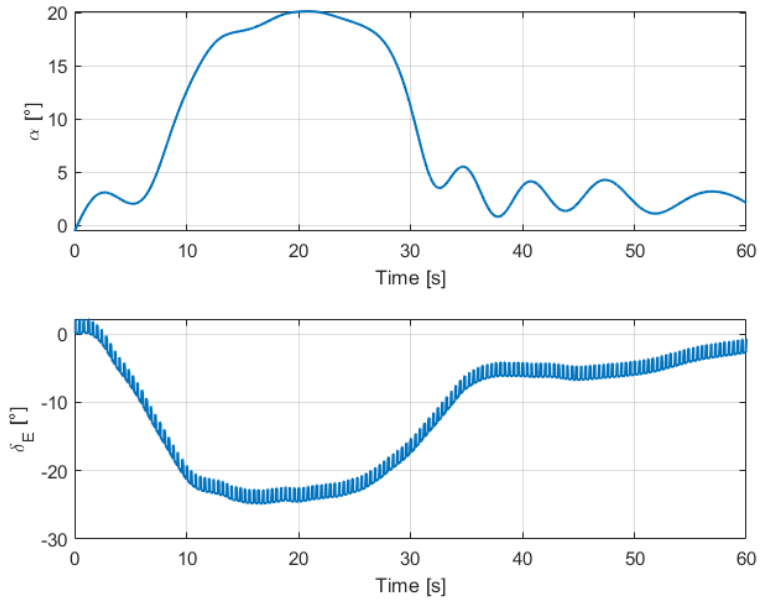



Figure 3.12: Angle of Attack and Control Surface Deflection

Furthermore, using these derivatives together with the angle of attack and elevator deflection, which are provided as inputs to the estimate method, total lift and drag forces and pitch moment coefficients are also computed. One can see the estimated and the real values of total aerodynamic forces and moment coefficients in Figure 3.13, and a zoomed version of this figure is given in Figure 3.14. According to these two figures, it can be seen that predicted total aerodynamic coefficients are quite similar to real ones.

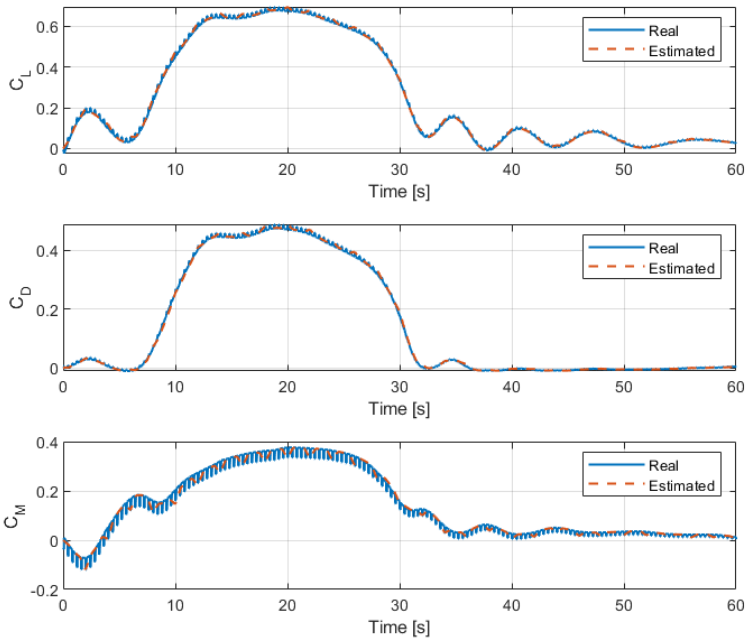



Figure 3.13: Results of Total Aerodynamic Force and Moment Coefficients

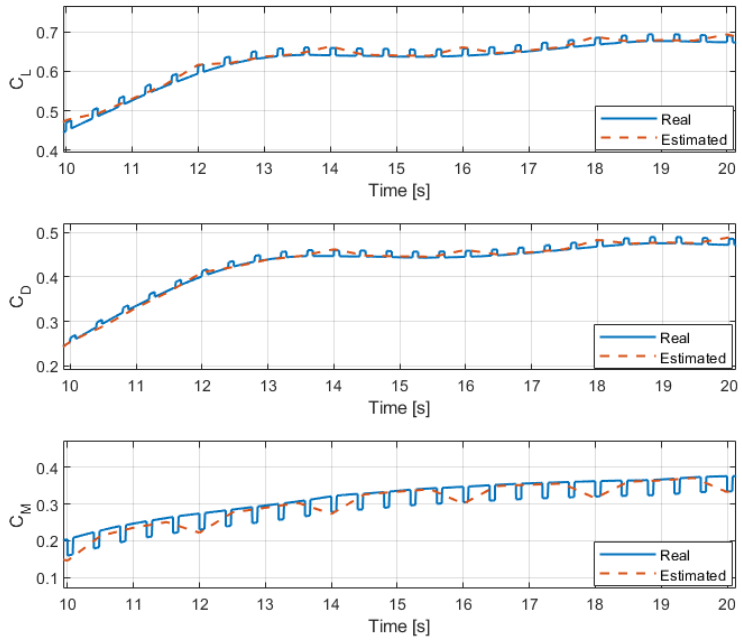



Figure 3.14: Results of Total Aerodynamic Force and Moment Coefficients - Zoomed Figure

To illustrate the effectiveness of the estimate technique, in addition to the result graphs above, the mean-squared error (MSE) value between the real values and estimation results has been calculated for each coefficient. The smaller the mean-squared error value, the more precise the estimation result. The MSE values of the total aerodynamic coefficients and their derivatives with respect to the angle of attack and the control surface angle are shown in Table 3.2.

Table 3.2: Mean-Squared Errors of Least-Squared Estimation Results

| Parameter | MSE | Parameter | MSE | Parameter | MSE |
|---|---|---|---|---|---|
| $C_L$ | $5.92 \times 10^{-6}$ | $C_D$ | $7.76 \times 10^{-6}$ | $C_M$ | $2.69 \times 10^{-5}$ |
| $C_{L_\alpha}$ | $2.25 \times 10^{-4}$ | $C_{D_\alpha}$ | $3.23 \times 10^{-5}$ | $C_{M_\alpha}$ | $3.89 \times 10^{-5}$ |
| $C_{L_{\delta e}}$ | $1.86 \times 10^{-5}$ | $C_{D_{\delta e}}$ | $5.22 \times 10^{-6}$ | $C_{M_{\delta e}}$ | $9.53 \times 10^{-6}$ |

All derivatives in the above figures and table are in the unit of $1/^\circ$.

# CHAPTER 4

# AERODYNAMIC PARAMETER ESTIMATION USING NEURAL NETWORK

## 4.1 Introduction

Neural networks are artificial intelligence models that have been developed to learn complex data relationships. The core components are artificial neurons, which behave similarly to neurons in the human brain. The process of creating a neural model begins with the definition of a network structure consisting of input, hidden, and output layers. The model's learning ability is determined by the number of neurons in each layer and the depth of the layers.

Firstly, the model is initialized with randomly selected weights and bias values and generally processes data using non-linear activation functions. During the training process, optimization techniques such as backpropagation and gradient descent are used to reduce the error between the model output and the actual values. Moreover, the loss function evaluates this error and decides how well the model performs on the training set. Furthermore, to improve the precision of the model, hyper-parameters might to be adjusted. Finally, a separate validation and test data set are employed to determine the model's generalization ability. This procedure is critical in evaluating the model's performance on real-world data and getting ready for implementation.

### 4.1.1 Feed-Forward Neural Network (FFNN)

In the engineering note [16], a feed-forward neural network (FFNN) has been shown as the one with the greatest potential for application in aircraft modeling. The main benefit of FFNN modeling is that it's able to represent highly nonlinear, complicated

relationships precisely, for which model theory with physical comprehension is tough. This ability of an FFNN is the reason that it has been used to estimate aerodynamic parameters from flight data. A few functions from MATLAB's deep learning toolbox [9], including "feedforwardnet", "train" were used to build the neural network model for this thesis study.

An input layer, one or more hidden layers, and an output layer are the layers of neurons that make up FFNN. The number of neurons in the input and output layers is determined by the number of variables that are inputs and outputs of the system, whose parameters will be estimated. Several algorithms have been proposed to determine the number of neurons in the hidden layer, and these algorithms are mentioned in [35]. One crucial hyperparameter affecting the model's complexity and learning ability is the number of neurons in the hidden layer. According to works in references [30] and [29], an FFNN that has one hidden layer with ten neurons is suitable for the longitudinal aerodynamic parameter estimation problem. According to Jategaonkar [14], choosing the optimal number of neurons of the hidden layer depends on certain aspects such as the number of inputs and outputs, how much noise there is in the data, and the level of complexity of the input-output domain. Various rules of thumb have been proposed in the literature for the number of neurons in the hidden layer ($n_h$), such as 1) midway between the number of inputs ($n_u$) and the number of outputs ($n_y$), 2) two-thirds of ($n_u + n_y$), or 3) less than $2n_u$. However, the effectiveness of these rules is questionable since they do not account for the previously mentioned noise and complexity concerns. The complicated subspaces may be under- or over-fitted by having too few or too many neurons, which can both have an impact on generalization, or overall performance and prediction capacity. In [14], it is concluded that the best strategy for determining the ideal number of neurons appears to be one based on numerical examinations of testing out numerous networks with varying numbers of hidden neurons and selecting the one with the lowest estimated error. This led to the experimentation of various hidden layers and neuron numbers in this thesis study. Hence, the decision was made to use a feed-forward neural network model with one hidden layer and ten neurons for this study.

The 3-DOF model, which is explained in chapter 2, was used to record flight data for the neural network algorithm. Measured data are accelerations ($a_x$ and $a_z$), changing of pitch rate ($\dot{q}$), pitch rate ($q$), body velocity ($V_{body}$), the angle of attack ($\alpha$) and elevator deflection ($\delta_e$). The input vector can be written as follows;

$$U = [a_x, a_z, \dot{q}, q, V_{body}, \alpha, \delta_e]^T \tag{4.1}$$

Whereby putting in measurement variables $a_x(k)$, $a_z(k)$, and dynamic pressure $Q(k)$ along with mass and inertia characteristics yields the values of $C_L(k)$, $C_D(k)$, $C_M(k)$ which are outputs of the neural network.

$$C_{F_x}(k) = \frac{m(k)a_x(k) - F_{thrust}(k)}{Q(k)S_{ref}(k)} \tag{4.2}$$

$$C_{F_z}(k) = \frac{m(k)a_z(k)}{Q(k)S_{ref}(k)} \tag{4.3}$$

$$C_M = \frac{I_{yy}\dot{q} + \dot{I}_{yy}q}{QS_{ref}l_{ref}} \tag{4.4}$$

$$C_L = C_{F_x}\sin\alpha - C_{F_z}\cos\alpha \tag{4.5}$$

$$C_D = -C_{F_x}\cos\alpha - C_{F_z}\sin\alpha \tag{4.6}$$

Also, the output vector of the neural model can be written as follows;

$$Z = [C_L, C_D, C_M]^T \tag{4.7}$$

In Figure 4.1, the feed-forward neural network architecture that was created for the longitudinal dynamics of the Pegasus launch vehicle is shown basically.

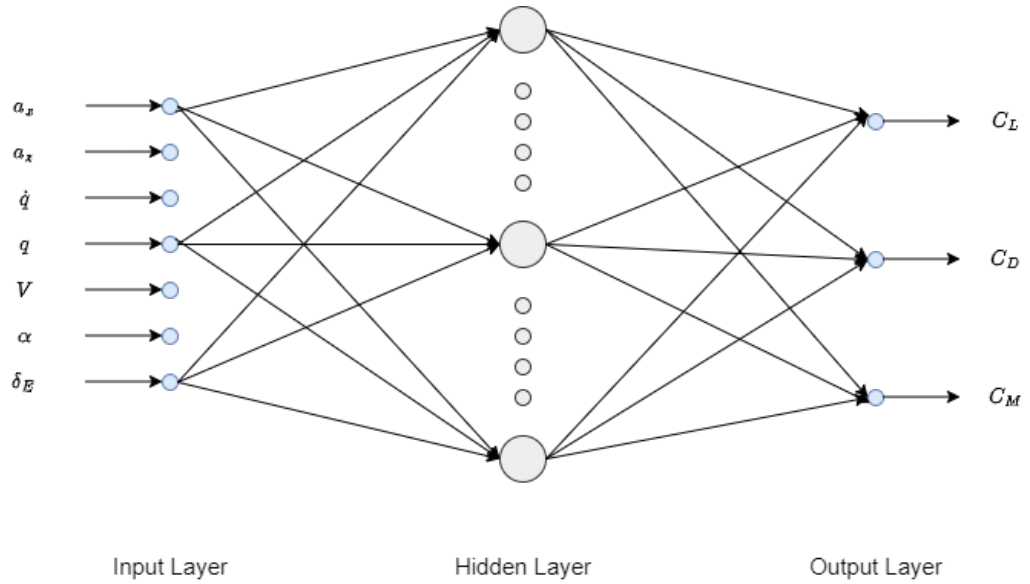



Figure 4.1: Basically Feed-Forward Neural Network Architecture For Longitudinal Motion Dynamic

## 4.2 Training of The Feed-Forward Neural Network

It can be said that training the neural network means regulating the weights between layers of the network. Back-propagation (BP) is the approach most often employed for the training procedure, with the basic idea being to observe the error as a function of network weights or parameters and carry out gradient descent in the parameter range in order to the lowest error between the desired and estimated values. Although back-propagation offers a way to compute gradients, it is not an optimization technique in and of itself. Rather, it begins with the output layer and uses the chain rule to get the slopes at the preceding (hidden) layers.
There are two phases in the training algorithm:

- **1) Forward Pass:** It essentially resembles simulation in that the input signal is propagated through the network while the weights are held constant; this is required to calculate the differences between the measured outputs and the network's estimated responses.
- **2) Backward Pass:** It computes the gradients, changes the weights, and prop-

agates the error backward using several layers.

Moreover, there are two most known types of back-propagation learning algorithms:

- **Batch or Sweep-Through:** Once the entire training data set has been presented, batch BP modifies the network weights. As a result, a training iteration includes a single sweep-through of all training patterns.
- **Sequential or Recursive:** The network weights are changed progressively when the training data set is given point by point, in the case of the recursive BP approach, also known as pattern learning.

According to [34], the recursive BP is more practical and effective when compared to batch BP. Since a single recursive run of the supplied data set is insufficient to reduce error, the process of iteratively computing the data is repeated several times, resulting in a recursive-iterative approach.

### 4.2.1 Forward Propagation

Forward propagation is the process of moving through the model by using various computations from the inputs to calculate the errors in the output. The initial step in forward propagation is to construct weight matrices between the input and hidden layers as well as between the hidden and output layers. The initial defined weight matrices are often made up of random values. The propagation between the two layers is expressed by multiplying the weight matrix for each node by the corresponding layer's input vector and adding bias. In the book of Jategaonkar [14], the propagation procedure between input and hidden layers is expressed with the following equations.

$$y_1 = W_1 u_0 + b_1 \tag{4.8}$$

$$u_1 = f(y_1) \tag{4.9}$$

Where $u_0$ is the vector of node inputs, $W_1$ is the weight matrix between input and hidden layer, $b_1$ is the bias vector between input and hidden layer, $y_1$ is the output of

summing connections, $u_1$ is the hidden layer node output vector. Moreover, $f$ is the node activation function. In the first equation, $y_1$ is linear, but linearity reduces the learning power of the neural model. Hence, the activation function is used to break this linearity issue. There are various activation functions, such as step function, sigmoid function, hyperbolic tangent function, rectified linear unit, etc. In this study, the hyperbolic tangent function was chosen since it is suggested in [14]. The "tansig" command of MATLAB is used as the activation function.
The above propagation procedure is repeated between the hidden layer and the output layer. This time, the node outputs ($u_1$) at the hidden layer are inputs to the output layer.

$$y_2 = W_2 u_1 + b_2 \tag{4.10}$$

$$u_2 = f(y_2) \tag{4.11}$$

### 4.2.2 Back-Propagation

Backpropagation may be seen as an essential step in developing an artificial neural network model. The primary goal of backpropagation, which is apparent by its name, is to decrease the error from the output to the input. As stated in the previous section, an output vector is created by forward propagation utilizing the randomly assigned weight matrices. The cost function is calculated by using the sum of squared errors between the forward pass output vector and the one supplied as the measurement for training the neural model. The weight matrices are then updated using the gradient descent approach to minimize the cost function, thereby improving the network's performance. The mathematical background of the backpropagation is described in [14].

There are several back-propagation functions for training the neural network. Bayesian regularization, conjugate gradient back-propagation, gradient descent back-propagation, Levenberg-Marquardt back-propagation, and scaled conjugate gradient back-propagation can be given as examples of these algorithms. In [5], they brought up the research on the effectiveness of different back-propagation techniques using MATLAB's train-

ing functions. According to this paper, they got better performance results by using Levenberg-Marquardt back-propagation and Polak-Ribiere conjugate gradient back-propagation. Moreover, it is shown that Levenberg-Marquardt has minimum mean-square errors and the best validation performance with minimum iterations, according to [5].

In addition, a study that is about training a feed-forward neural network by using Levenberg-Marquardt is presented in [11]. They mentioned the Marquardt algorithm for non-linear least squares and applied it as a back-propagation method to train a feed-forward neural network. Furthermore, they compared this method with the conjugate gradient algorithm and with a variable learning rate variation of the back-propagation technique. The outcomes of that research show that the Marquardt method performs very well when training networks with several hundredweights. The authors of the paper also noticed that while the conjugate gradient and variable learning rate algorithms could not converge, the Marquardt approach mostly succeeded. Furthermore, the paper [20] presents the Levenberg-Marquardt method for optimizing parameter estimation in nonlinear models, combining Taylor series and gradient approaches to enhance speed and accuracy. According to this brief literature review about the back-propagation algorithms, the Levenberg-Marquardt approach was chosen for the training algorithm in this study. Moreover, the "trainlm" function of MATLAB was used for this purpose.

According to [21], the Jacobian $J$ of performance concerning the weight and bias variables $X$ is computed using backpropagation. Each variable is modified by Levenberg-Marquardt;

$$JJ = J * J \tag{4.12}$$

$$Je = J * E \tag{4.13}$$

$$dX = \frac{-(JJ + I\mu)}{Je} \tag{4.14}$$

where $E$ is the error and $I$ is the identity matrix.

There are some important parameters that affect the performance of the training algorithm. In this section, these parameters will be explained briefly according to [14]. One of these parameters is the learning rate ($\mu$), which represents the step size. If

the learning rate is chosen as very small, convergence will be slow. On the contrary, learning will be faster if the learning rate is chosen as a large value. However, the larger values can also cause sudden shifts in the direction of descent from discrete points to points, especially if the cost function includes strong spikes and drops, resulting in weight oscillations. Another parameter called the momentum factor ($\Omega$) is defined to prevent these oscillations. In this case, the multiplication of the momentum term and the previous weight change is added to the weight-updating equation as an extra parameter. The weight changing is forced to move in a particular direction in weight space by using the momentum term; therefore, it prevents stacking on local minimum points.

Jategaonkar's book [14] includes the recommended learning rate and momentum term values for the aerodynamic parameter estimation problem. It is mentioned that it will be sufficient to choose the range $(0.1 - 0.3)$ for the learning rate and the range $(0.3-0.5)$ for the momentum term. Based on this, after trying several different values, the learning rate was determined to be 0.3, and the momentum constant was determined to be 0.5 for this study.

There is another essential concern in the training of a neural network. The thing is that one needs to normalize the inputs and outputs of the neural network since variables in the inputs and outputs can have different ranges. If normalization is not done, it may cause some of the weights or some of the biases to be too big or small, and the network will be unstable. Hence, normalization provides faster training and may prevent the overfitting problem. In this study, all parameters in the input and output vectors were normalized between 0 and 1, as shown below.

$$P_n = \frac{(P_i - P_{min})}{(P_{max} - P_{min})} \tag{4.15}$$

where $P_n$ represents the normalized value, $P_i$ is the input value, $P_{max}$ is the maximum value of each input vector, and $P_{min}$ is the minimum value of each input vector.

One hyperparameter that specifies how many times the learning algorithm will run through the training dataset is the number of epochs. In MATLAB, the number of epochs is set to 1000 as default.

To improve the generalization of the neural network, three different data sets were recorded from the 3-DOF model (this model was used to generate data since there is

no flight data), and all 3 data were used in training together.

After normalizing them, the data from the 3-DOF model was fed into the "train" function as the input and output vectors mentioned in the Section 4.1.1. It is stated in [1] that the train-validation-test data partition ratio of 2:1:1 has been acknowledged for many years. According to [10], roughly 80% of the data may be utilized for training and 20% for validation and testing. In this study, the data set is divided into three parts: 70% for training, 15% for validation, and 15% for testing. While the model is trained using the training portion of the data set, the validation part is used to evaluate the generalization ability of the model in each training iteration. Furthermore, the model's performance and generalization ability are tested in the section designated for model testing.

## 4.3 Performance of Neural Model

The performance of the neural network is evaluated by using mean squared error (MSE) after training. The MSE values that the neural model has during training, validation, and testing can be seen in Figure 4.2.

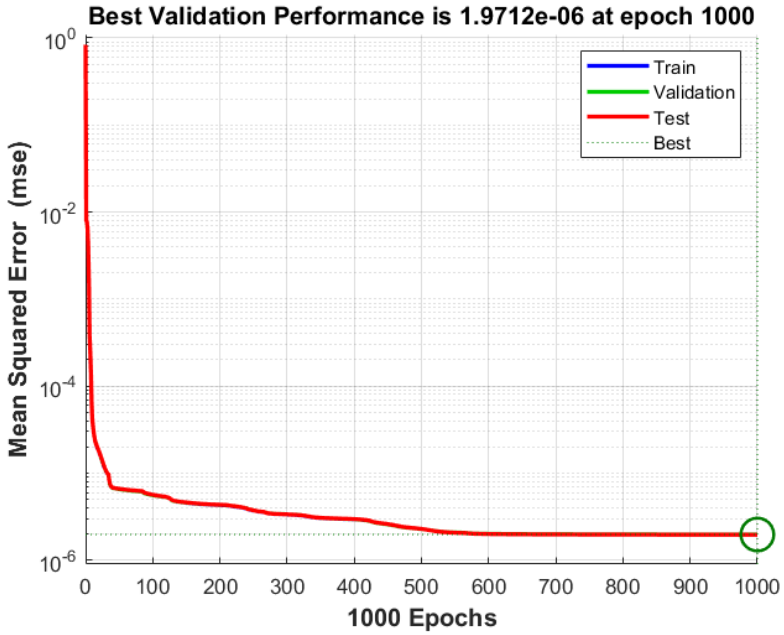



Figure 4.2: Mean-Squared Error of Neural Network Training

One can see that the MSE of training, validation, and testing have a decreasing slope through epochs. Decreasing the slope means increasing the learning ability of the model while reducing error. In general, it can be said that the MSE plot may indicate overfitting or underfitting if the error on the validation data increases or remains constant while the error on the training data continuously decreases. The optimal model should have similar MSE values in the training and validation sets. According to this, one can say that the neural model is optimum by looking at the figure. Additionally, it can be observed that MSE values in the figure stay consistent beyond around the 600th to 700th epoch. This indicates that following MSE stabilization, the neural model's learning capability can not increased more.

In addition to MSE, the neural model was evaluated by giving three data sets as inputs separately, and outputs were compared with the actual values in figures 4.3, 4.4 and 4.5.

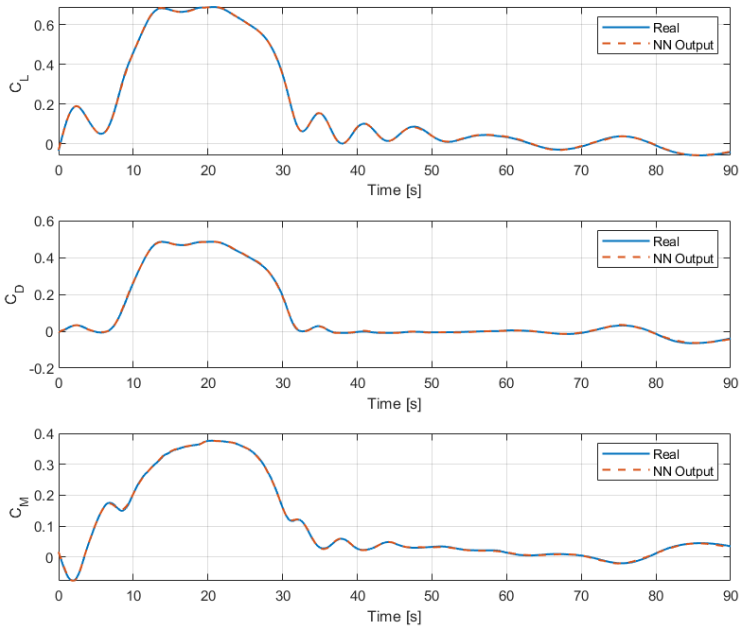



Figure 4.3: Data Set 1 Results

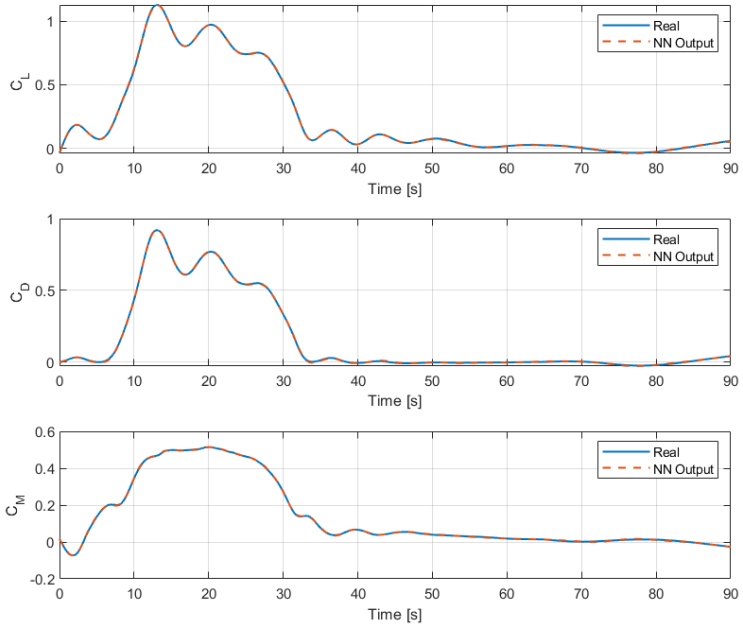



Figure 4.4: Data Set 2 Results

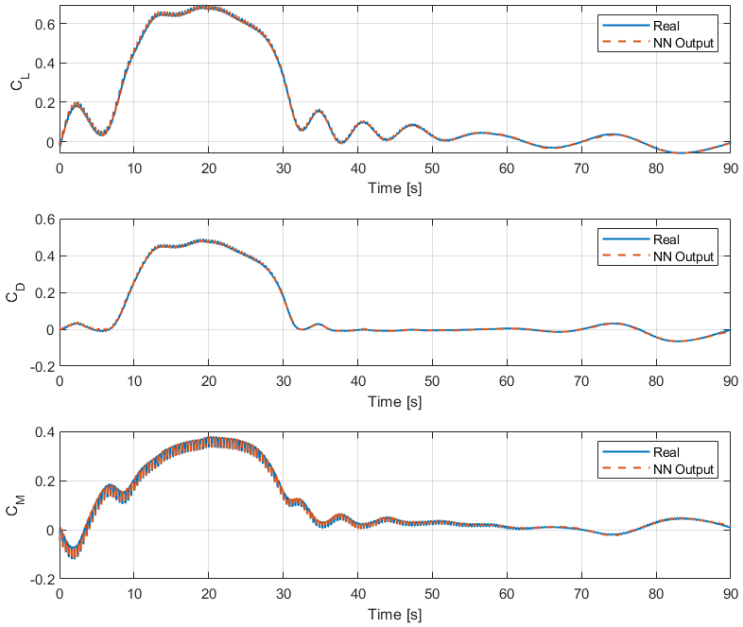



Figure 4.5: Data Set 3 Results

By looking at the above figures, it is seen that the aerodynamic coefficients, which are the outputs of the neural model, and the real ones are on top of each other for three different data sets. Besides the MSE graph, these plots also show that the neural model is working properly.

## 4.4 Aerodynamic Derivative Estimation by Neural Network and Least Squares Method Together

After the neural network training, it was used to estimate the aerodynamic coefficients $C_L$, $C_D$, and $C_M$ for different input sets. However, this study aims to estimate the aerodynamic derivatives ($C_{L\alpha}$, $C_{L_{\delta_e}}$, $C_{D_\alpha}$, $C_{D_{\delta_e}}$, $C_{M_\alpha}$, and $C_{M_{\delta_e}}$). Hence, they need to be calculated by using aerodynamic coefficients, angle of attack, and elevator deflection. In [14], there is an approach suggested for this problem. In this approach, one employs two sets of perturbed $\alpha$ and $\delta_e$ input variables to produce FFNN predictions separately, leaving the other inputs from the training set unmodified. The constant values $\Delta\alpha$ and $\Delta\delta_e$ represent the disturbances. For each of the N data points, the identical perturbation is applied. The derivative is calculated by using the central difference technique to the outputs of the neural model, which has been conducted with perturbed inputs.

$$C_{L_\alpha} = \frac{C_L^+ - C_L^-}{2\Delta\alpha} \tag{4.16}$$

where $C_L^+$ and $C_L^-$ are the neural network outputs corresponding to the inputs $\alpha + \Delta\alpha$ and $\alpha - \Delta\alpha$. The aerodynamic derivatives that are estimated by using this approach are shown in figures 4.6, 4.7, and 4.8.

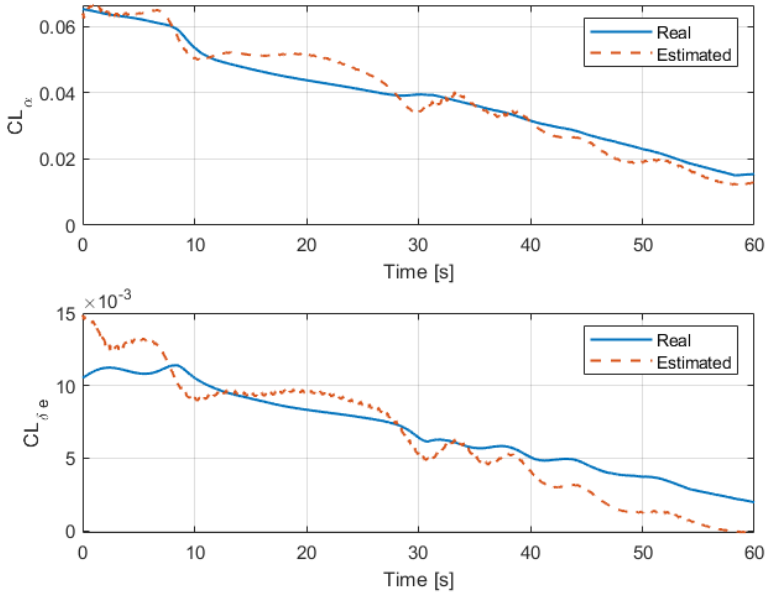



Figure 4.6: Estimation Results of Lift Coefficient Derivatives

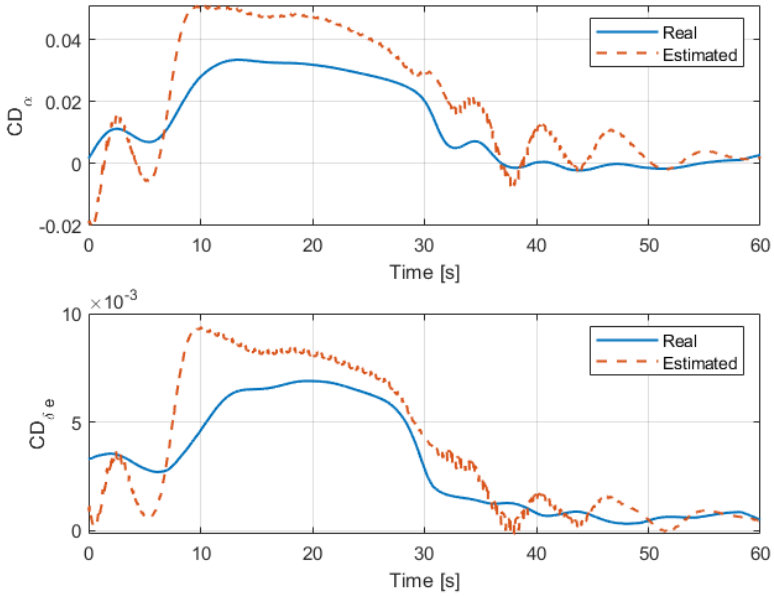



Figure 4.7: Estimation Results of Drag Coefficient Derivatives

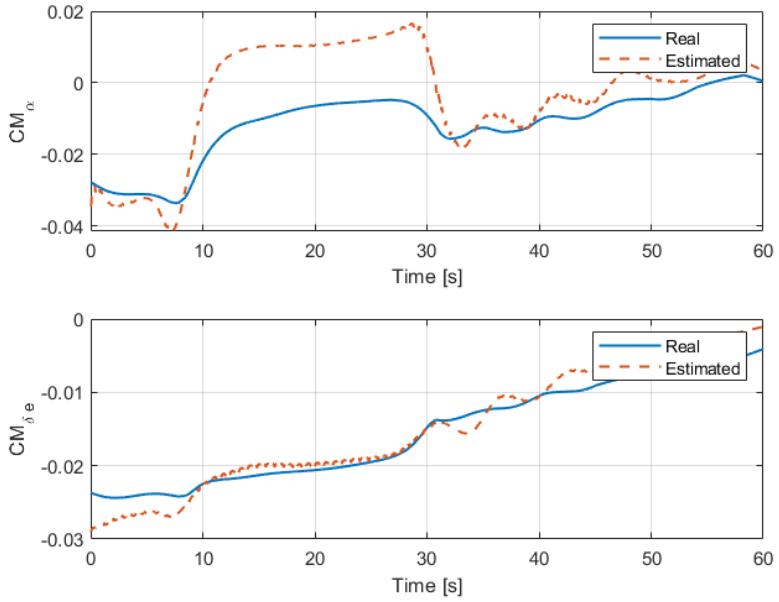



Figure 4.8: Estimation Results of Pitch Moment Coefficient Derivatives

By looking at the result graphs above, it can be seen that the predicted aerodynamic derivatives are not very close to the actual values. Therefore, it can be said that the method mentioned in the book of running the neural model by giving perturbations to $\alpha$ and $\delta E$ and applying the central difference to the outputs does not work very well. The reason for this may be that when calculating the aerodynamic derivative with this method, the change of the total aerodynamic force according to only one parameter ($\alpha$ and $\delta E$, respectively) is obtained. Since this method has a more crude approach, the least-squares estimation method was also used here to obtain the aerodynamic derivative coefficients after obtaining the aerodynamic coefficients with the neural model.

$$\begin{bmatrix} C_L(k) & C_D(k) & C_{My}(k) \\ \vdots & \vdots & \vdots \end{bmatrix} = \begin{bmatrix} \alpha(k) & \delta_e(k) \\ \vdots & \vdots \end{bmatrix} \begin{bmatrix} C_{L_\alpha} & C_{D_\alpha} & C_{My_\alpha} \\ C_{L_{\delta_e}} & C_{D_{\delta_e}} & C_{My_{\delta_e}} \end{bmatrix} \tag{4.17}$$

The left-hand side of the equation 4.17 was estimated from the neural model. As mentioned in the chapter 4, aerodynamic derivatives were obtained by applying the below formulation. This approach differs from the one described in Chapter 4 in

that instead of deriving the aerodynamic coefficients from measurement data using mathematical operations, we used a trained neural model.

$$x = (A^T A)^{-1} A^T y \tag{4.18}$$

## 4.5 Results

In this section, the aerodynamic coefficient results estimated by using the neural network and least squares methods together are included. Initially, many data sets captured at 1000 Hz from the 3-DOF model were used as flight data to train the neural network. Subsequently, the trained neural model was given the data set used to get the results in Section 3.5, generating the total aerodynamic coefficients that are displayed in Figure 4.5. The least-squares approach was used to estimate the aerodynamic derivative coefficients using these total aerodynamic coefficients as well as the angle of attack and elevator angle found in the relevant data set. As stated in Section 3.5, the prediction algorithm ran at a frequency of 2 Hz. Figures 4.9, 4.10, and 4.11 present the estimated aerodynamic derivative coefficients together with their actual values.

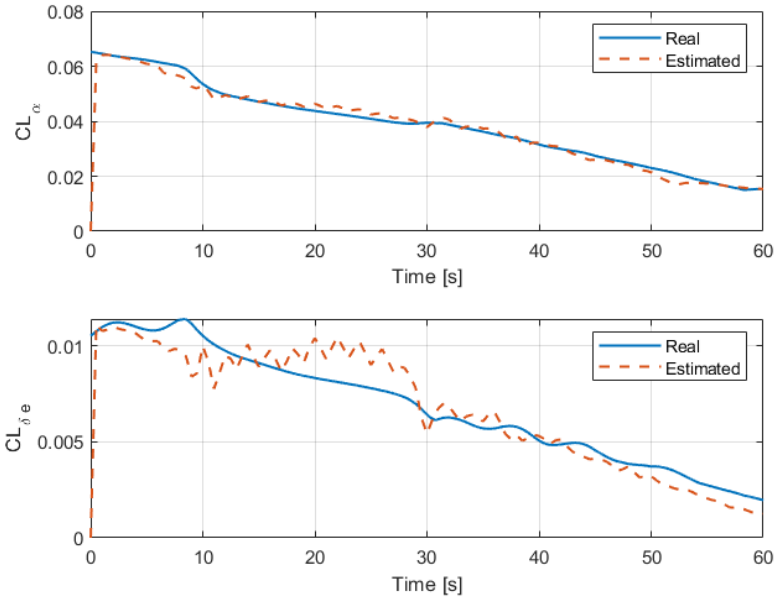



Figure 4.9: Estimation Results of Lift Coefficient Derivatives

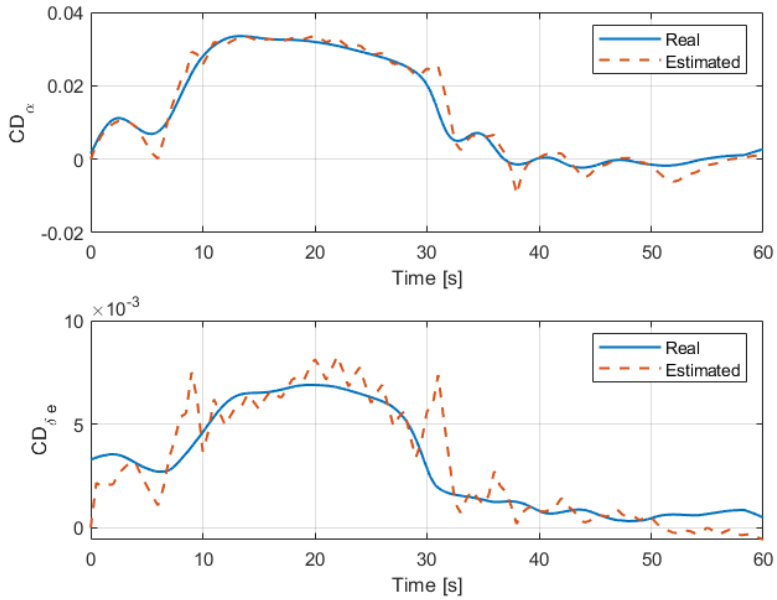



Figure 4.10: Estimation Results of Drag Coefficient Derivatives

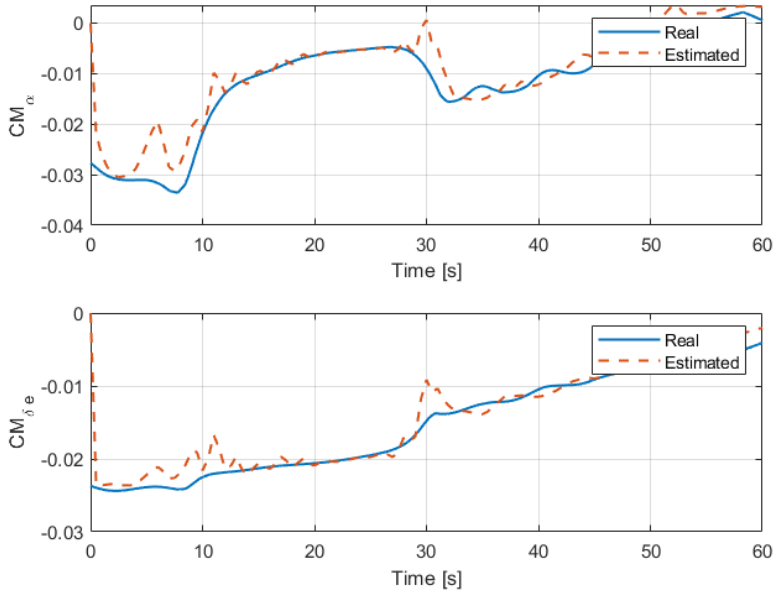



Figure 4.11: Estimation Results of Pitch Moment Coefficient Derivatives

It is evident from the figures mentioned above that the aerodynamic derivative values derived by the neural network approach closely resemble the actual values. Here, it can be seen that the predicted values deviate slightly from the actual ones in regions where the angle of attack and elevator angle change rapidly, which is similar to the Least-square estimation findings in Section 3.5.

Additionally, the mean-squared error (MSE) value between the actual values and the predicted results has been obtained for each coefficient to show the efficacy of the estimation approach. The estimation result is more accurate with the smaller the mean-squared error value. Table 4.1 shows the MSE values of the overall aerodynamic coefficients and their derivatives to the control surface angle and the angle of attack.

Table 4.1: Mean-Squared Errors of Neural Network Approach Results

| Parameter | MSE | Parameter | MSE | Parameter | MSE |
|---|---|---|---|---|---|
| $C_L$ | $9.02 \times 10^{-6}$ | $C_D$ | $9.82 \times 10^{-6}$ | $C_M$ | $2.88 \times 10^{-5}$ |
| $C_{L_\alpha}$ | $3.81 \times 10^{-5}$ | $C_{D_\alpha}$ | $6.90 \times 10^{-6}$ | $C_{M_\alpha}$ | $1.82 \times 10^{-5}$ |
| $C_{L_{\delta e}}$ | $1.89 \times 10^{-6}$ | $C_{D_{\delta e}}$ | $1.31 \times 10^{-6}$ | $C_{M_{\delta e}}$ | $7.11 \times 10^{-6}$ |

All derivatives in the above figures and table are in the unit of $1/^{\circ}$.

The findings in Table 4.1 can be compared to the MSE results of the least-squares approach in Table 3.2. Based on the information shown in these two tables, it can be stated that MSE values in Table 4.1 are almost tenfold smaller than MSE values in Table 3.2. Thus, it can be said that the results obtained through the combination of neural network and least-squares approaches are more accurate than those acquired only from the least-squares method. In chapter 6, a more comprehensive comparison will be presented.

# CHAPTER 5

# AERODYNAMIC PARAMETER ESTIMATION USING OUTPUT-ERROR METHOD

## 5.1 Introduction

The Output Error (OE) method is a technique that is commonly used in system identification and parameter estimation. The difference or error between a mathematical model's output and the actual system output is calculated in this method. Then, the error is attempted to be minimized by optimizing model parameters. The output error method can be described mathematically as an optimization problem; model parameters are modified to reduce the sum of squares of the error function.

The output error method simulates the model's output utilizing time series data and finds the parameter set that minimizes the cost function. Algorithms such as Gauss-Newton or Levenberg-Marquardt, which are successive approximate optimization methods, are generally employed to solve the technique. Moreover, this approach performs well even when the model has a nonlinearity since the nonlinearity of the model output has no influence on the procedure of lowering the error function.

One of the advantages of the output error approach is that it eliminates the need to model the internal dynamics of the system because the model is based directly on input-output data. This can be particularly beneficial when the system's internal knowledge is inadequate. However, the method's performance heavily depends on a well-defined model framework and precise initial parameter estimates. As a result, the output error method generally requires wide initial analysis and careful model selection.

## 5.2 Methodology of The Maximum Likelihood Estimation

In the system identification application, maximum likelihood theory may be used to estimate unknown parameters vector ($\Theta$) by using an assortment of measurements ($z_1, z_2, ..., z_N$) with $N$ samples. The likelihood function, which represents the probability density $p(z|\Theta)$ of the variables that were observed, is written as follows;

$$p(z|\Theta) = \prod_{k=1}^{N} p(z_k|\Theta) \tag{5.1}$$

Fisher established the maximum likelihood (ML) approach as a common prediction process, [14]. It essentially involves choosing a value for $\Theta$ within the acceptable range so that $p(z|\Theta)$ is maximized.

Formulating the expression for $p(z|\Theta)$, or the conditional probability density function, is necessary to use the maximum likelihood function in the dynamical system. First of all, the error between the measurements $(z(t_k))$ and the outputs of the mathematical model $(y(t_k))$ is defined as $\nu(t_k) = z(t_k) - y(t_k)$. The relationship that follows may be expressed assuming that the error $\nu$ at every time step $t_k$ is statistically independent;

$$E\{\nu(t_k)\nu^T(t_k)\} = R\delta_{kl} \tag{5.2}$$

$\delta_{kl}$ is called the Kronecker delta, whose details can be found in [14], and it is equal to 1 if the $k$ equals $l$, and it is equivalent to 0 if $k$ does not equal $l$. The likelihood function is rewritten by considering the measurement error covariance matrix $R$ as the condition.

$$p(z|\Theta, R) = \prod_{k=1}^{N} p[z_k|\Theta, R] \tag{5.3}$$

$$p(z|\Theta, R) = \{(2\pi)^{n_y}|R|\}^{-N/2} exp\left[-\frac{1}{2}\sum_{k=1}^{N}[z(t_k) - y(t_k)]^T \times R^{-1}[z(t_k) - y(t_k)]\right] \tag{5.4}$$

Instead of solving the above probably equation to estimate unknown parameters $\Theta$, minimizing $L(z|\Theta, R)$, the negative logarithm of the probability function, provides a comparable and more feasible solution to the optimization problem [14]:

$$L(z|\Theta, R) = \frac{1}{2}\sum_{k=1}^{N}[z(t_k) - y(t_k)]^T R^{-1}[z(t_k) - y(t_k)] + \frac{N}{2}\ln[\det(R)] + \frac{Nn_y}{2}\ln(2\pi) \tag{5.5}$$

## 5.3 Aerodynamic Parameter Estimation Using The Maximum Likelihood Function

It is considered that the generic nonlinear system description below represents the mathematical model of the dynamic system whose parameters need to be estimated:

$$\dot{x}(t) = f[x(t), u(t), \beta], \qquad x(t_0) = x_0 \tag{5.6}$$

$$y(t) = g[x(t), u(t), \beta] \tag{5.7}$$

$$z(t_k) = y(t_k) + G\nu(t_k) \tag{5.8}$$

Where $\beta$ represents the parameters that will be estimated. For the system identification problem of Pegasus, the mathematical model is defined using the following equations. System states are defined as $x = [u, w, q, \theta]$ and outputs of the mathematical model are $y = [\dot{u}, \dot{w}, q, \dot{q}]$. Firstly, the aerodynamic forces and moment equations at the center of gravity are written using the initial aerodynamic coefficients.

$$F_x = QS_{ref}C_{F_x} \tag{5.9}$$

$$F_z = QS_{ref}C_{F_z} \tag{5.10}$$

$$M_y = QS_{ref}l_{ref}C_{M_y} + F_z(x_{cg} - x_{cp}) \tag{5.11}$$

Then, equations of motion are written as follows;

$$\dot{u} = -qw - g\sin\theta + \frac{F_x + F_t}{m} \tag{5.12}$$

$$\dot{w} = qu + g\cos\theta + \frac{F_z}{m} \tag{5.13}$$

$$\dot{q} = \frac{M_y - \dot{I}_y q}{I_y} \tag{5.14}$$

$$\dot{\theta} = q \tag{5.15}$$

These equations of motion are solved by using Euler's method with $\Delta t = 0.001s$ step size for each time instant.

$$x(t_{k+1}) = x(t_k) + \dot{x}(t_k)\Delta t \tag{5.16}$$

The parameters which will be estimated vector is defined as
$\Theta = [C_{F_x\alpha}, C_{F_x\delta_e}, C_{F_z\alpha}, C_{F_z\delta_e}, C_{M_y\alpha}, C_{M_y\delta_e}]$, and these are initially guessed. By using these, the aerodynamic force and moment coefficients are obtained, and they are used in the equations of motion.

$$C_{F_x} = C_{F_x\alpha}\alpha + C_{F_x\delta_e}\delta_e \tag{5.17}$$

$$C_{F_z} = C_{F_z\alpha}\alpha + C_{F_z\delta_e}\delta_e \tag{5.18}$$

$$C_{M_y} = C_{M_y\alpha}\alpha + C_{M_y\delta_e}\delta_e \tag{5.19}$$

After mathematical equations are solved, the measurement errors ($\nu$) between the measurements and the mathematical model outputs are calculated at each time instant.

$$\nu(t_k) = z(t_k) - y(t_k) \tag{5.20}$$

After this point, the likelihood function can be written in two steps. The first step demonstrates that the maximum likelihood calculation of R is provided for each given value of the parameter vector $\Theta$.

$$R = \frac{1}{N}\sum_{k=1}^{N}[z(t_k) - y(t_k)][z(t_k) - y(t_k)]^T \tag{5.21}$$

The cost function can be written as a substitution of Equation 5.21 into Equation 5.5.

$$J(\Theta) = \frac{1}{2} n_y N + \frac{N}{2} \ln[\det(R)] + \frac{N n_y}{2} \ln(2\pi) \tag{5.22}$$

Remember that given a hypothesized model and a data set under analysis, $n_y$ and $N$ are fixed. Consequently, the first and last term on the right-hand side of equation 5.22 turns out to be a fixed value and may be canceled. As a result, the cost function becomes:

$$J(\Theta) = \det(R) \tag{5.23}$$

Then, the cost function will be minimized by using an optimization algorithm. In this study, the Gauss-Newton Algorithm was chosen as the optimization method.

## 5.4 Gauss - Newton Optimization Technique

The Gauss-Newton method is an iterative method that is frequently used in solving system identification problems. It updates the parameters to be estimated at each iteration to minimize the cost function. The condition for reducing the cost function is that its partial derivative, according to the system parameter, equals zero.

$$\frac{\partial J(\Theta)}{\partial \Theta} = 0 \tag{5.24}$$

When two terms are eliminated, the expansion of the Taylor series of $\frac{\partial J(\Theta)}{\partial \Theta}$ around the parameter vector $\Theta$ at the $i^{th}$ iteration is written as follows:

$$\left(\frac{\partial J(\Theta)}{\partial \Theta}\right)_{i+1} \approx \left(\frac{\partial J(\Theta)}{\partial \Theta}\right)_i + \left(\frac{\partial^2 J(\Theta)}{\partial \Theta^2}\right)_i \Delta\Theta \tag{5.25}$$

$$\Delta\Theta = \Theta_{i+1} - \Theta_i \tag{5.26}$$

The $\Delta\Theta$ is the parameter updating value for the next iteration. The $(\partial^2 J(\Theta)/\partial\Theta^2)_i$ represents the cost function's second gradient respecting $\Theta$, and it is also called the

Hessian. According to the condition in equation 5.24, the right-hand side of the equation 5.25 equals zero, and the $\Delta\Theta$ equation is obtained.

$$\Delta\Theta = -\left[\left(\frac{\partial^2 J(\Theta)}{\partial\Theta^2}\right)_i\right]^{-1}\left(\frac{\partial J(\Theta)}{\partial\Theta}\right)_i \tag{5.27}$$

The algorithm usually requires greater incremental steps farther from the cost function minimum, according to greater values of the gradient, while the steps get progressively smaller as the minimum approaches. The cost function's shape will determine how many steps are needed to find the minimum in a given scenario. Under the initial values, the search process may identify a local extremum if the cost function includes many minima.

To obtain the $\Delta\Theta$, one first needs to calculate the first and second gradients of the cost function. They may be calculated by using the following equations:

$$\frac{\partial J(\Theta)}{\partial\Theta} = -\sum_{k=1}^{N}\left[\frac{\partial y(t_k)}{\partial\Theta}\right]^T R^{-1}[z(t_k) - y(t_k)] \tag{5.28}$$

$$\frac{\partial^2 J(\Theta)}{\partial\Theta^2} = \sum_{k=1}^{N}\left[\frac{\partial y(t_k)}{\partial\Theta}\right]^T R^{-1}\left[\frac{\partial y(t_k)}{\partial\Theta}\right] + \sum_{k=1}^{N}\left[\frac{\partial^2 y(t_k)}{\partial\Theta^2}\right]^T R^{-1}[z(t_k) - y(t_k)] \tag{5.29}$$

It is discovered that the statement $[z(t_k) - y(t_k)]$ in the second term on the right-hand side of equation 5.29 disappears when the process converges, [14]. This term will disappear over time if a sufficient number of data points are collected, depending on the zero mean and independent noise assumptions. The second gradient of the cost function (Hessian) can be rewritten as follows:

$$\frac{\partial^2 J(\Theta)}{\partial\Theta^2} \approx \sum_{k=1}^{N}\left[\frac{\partial y(t_k)}{\partial\Theta}\right]^T R^{-1}\left[\frac{\partial y(t_k)}{\partial\Theta}\right] \tag{5.30}$$

The sensitivity matrix, or response gradients $\frac{\partial y(t_k)}{\partial\Theta}$, calculation covers a few crucial challenges that directly impact the ability to handle various nonlinear models. In the

traditional method, the sensitivity equations are solved and yield the sensitivity matrix $\frac{\partial y(t_k)}{\partial \Theta}$.

$$\frac{\partial \dot{x}}{\partial \Theta} = \frac{\partial f}{\partial x}\frac{\partial x}{\partial \Theta} + \frac{\partial f}{\partial \Theta} \tag{5.31}$$

$$\frac{\partial y}{\partial \Theta} = \frac{\partial g}{\partial x}\frac{\partial x}{\partial \Theta} + \frac{\partial g}{\partial \Theta} \tag{5.32}$$

The use of transition matrices in the solution of the first equation and basic matrix multiplications offers an advanced approach to dealing with linear systems. Nevertheless, any modifications to the model structure require re-deriving the above sensitivity equations if the system model is nonlinear. Hence, adjustments to the parameter estimate software are required. Moreover, the challenge of theoretically characterizing the derivatives may lead to numerical issues when systems have discontinuous nonlinearities. The sensitivity parameters may be approximated by numerical differences, which negates the requirement for their proper derivation. The central difference approach is preferred to obtain sensitivity coefficients.

$$\left[\frac{\partial y(t_k)}{\partial \Theta}\right]_{ij} \approx \frac{y_i^{+p}(t_k) - y_i^{-p}(t_k)}{2\delta\Theta_j} \qquad i = 1, \ldots n_y; \quad j = 1, \ldots n_q \tag{5.33}$$

$$\left[\frac{\partial y(t_k)}{\partial \Theta}\right]_{ij} \approx \frac{g_i[x^{+p}(t_k), u(t_k), \Theta + \delta\Theta_j e^j] - g_i[x^{-p}(t_k), u(t_k), \Theta - \delta\Theta_j e^j]}{2\delta\Theta_j} \tag{5.34}$$

where $n_y$ is the size of the output vector, $n_q$ is the size of the parameters vector that will be estimated, $\delta\Theta_j$ represents the perturbation which is given to the $j^{th}$ element of the unknown parameter vector and $e^j$ is a column vector consisting of one in the $j^{th}$ level and zeros every other row. In the central difference approximation, system outputs are calculated twice by adding perturbation to the system parameter and subtracting it from the system parameters. These calculations are represented by $+p$ and $-p$ symbols, respectively. Then, the subtraction of the minus perturbed output from the plus perturbed output is divided by two times the perturbation.
The perturbation value is taken as $\delta\Theta_j = 10^{-6}\Theta_j$ since it is suggested by [14].

## 5.5 Application of the Algorithm

### 5.5.1 Windowing and Initial Guess

First, the data set was separated into 0.5-second windows, and the output error method was run to estimate parameters for each window. At the beginning of the algorithm, initial guess values were given randomly within the $\mp 10\%$ of the real values. In real-life applications, initial value predictions can be made by looking at the CFD results or the wind tunnel results. After the first-time window estimation, the estimated values are used as the initial value of the next window.

### 5.5.2 Regulation of Step Size and Iteration Terminate

In some cases, the Gauss-Newton algorithm might not work well, or it may not have converged locally. It may be given as an example of such cases in which the finite difference approximation of gradients around the ideal point is due to non-optimal initial guesses or numerical inaccuracies. In these situations, it is essential to regulate the step size carefully after the heading of the adjustment vector has been identified. In [14], three different methods are introduced as solutions to this problem: the heuristic approach, the line search method, and the dominant directions approach. In this study, the heuristic approach was chosen for step size control because of its ease of use.

According to the heuristic application, the parameter update is decreased by half whenever the cost function is larger than the one in the previous iteration. It is similar to reducing the entire step one by one using the factors $(2, 4, 8, \ldots)$. All that needs to be done for these half-stages is the cost-function examination; the gradient does not need to be computed. In the algorithm, this is done by adding a weight factor to the parameter change equation in the manner described below:

$$\Delta\Theta = -2^{-(k-1)} \left[ \left( \frac{\partial^2 J(\Theta)}{\partial \Theta^2} \right)_i \right]^{-1} \left( \frac{\partial J(\Theta)}{\partial \Theta} \right)_i \tag{5.35}$$

The parameter $k$ is set as one at the beginning of each iteration; it is increased by one

if the cost of the current iteration is greater than the one of the previous iteration until the cost function starts to converge.

For the termination of iterations, the relative error of the cost calculation was added to the algorithm.

$$E_{rel} = \frac{J_i - J_{i-1}}{J_i} \tag{5.36}$$

where $E_{rel}$ represents the relative error, and $J_i$ and $J_{i-1}$ are the current and previous costs, respectively.
Error tolerance was introduced as $10^{-5}$ at the beginning of the algorithm. When the relative error value of the iteration is smaller than the error tolerance, the iterations are over since convergence was achieved. Furthermore, the estimation algorithm continues with the next time window.

### 5.5.3 Defining of Unknown Parameters

Let's remember that this study aims to estimate the aerodynamic force and moment derivatives, which are the $[C_{L\alpha}, C_{L\delta_e}, C_{D\alpha}, C_{D\delta_e}, C_{M_y\alpha}, C_{M_y\delta_e}]$. If one wants to estimate $C_{L\alpha}, C_{L\delta_e}, C_{D\alpha}, C_{D\delta_e}$ directly, this needs an extra calculation process in the mathematical model. In the OEM algorithm, the lift, drag, and pitch moment coefficients are calculated by using updated estimation parameters in each iteration, and they are used in the mathematical model to obtain the outputs of the system. The lift and drag coefficients cannot be used directly in the equations of motion; they are converted to the body axis coordinate system, and then accelerations are obtained using the force coefficients in the body frame.

$$C_L = C_{L_\alpha}\alpha + C_{L_{\delta_e}}\delta_e \tag{5.37}$$

$$C_D = C_{D_\alpha}\alpha + C_{D_{\delta_e}}\delta_e \tag{5.38}$$

$$C_{Fx} = C_L \sin\alpha - C_D \cos\alpha \tag{5.39}$$

$$C_{Fz} = -C_L \cos\alpha - C_D \sin\alpha \tag{5.40}$$

These operations cause extra non-linearity in the system due to trigonometric terms. It becomes clear throughout algorithm development that this non-linearity has a negative impact on the performance of the optimization algorithm. Therefore, it is decided that $C_{F_x\alpha}, C_{F_x\delta_e}, C_{F_z\alpha}, C_{F_z\delta_e}$ will be the estimated instead of $C_{L\alpha}, C_{L\delta_e}, C_{D\alpha}, C_{D\delta_e}$. At the end of the algorithm, the estimated parameters are converted into the desired parameters by using the axis transformation.

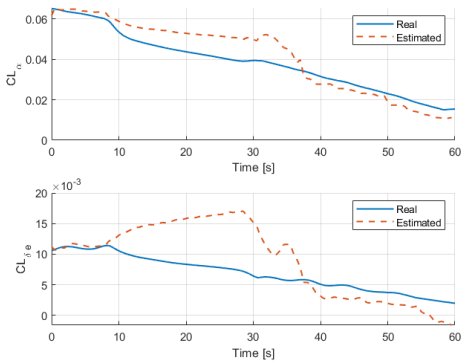
$$C_{L_\alpha} = C_{F_x\alpha} \sin\alpha - C_{F_z\alpha} \cos\alpha \tag{5.41}$$

$$C_{L_{\delta e}} = C_{F_x\delta_e} \sin\alpha - C_{F_z\delta_e} \cos\alpha \tag{5.42}$$

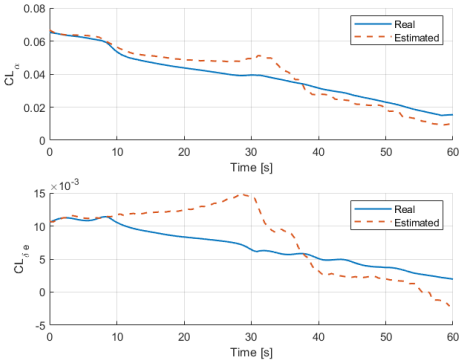
$$C_{D_\alpha} = -C_{F_x\alpha} \cos\alpha - C_{F_z\alpha} \sin\alpha \tag{5.43}$$

$$C_{D_{\delta e}} = -C_{F_x\delta_e} \cos\alpha - C_{F_z\delta_e} \sin\alpha \tag{5.44}$$

The graphs in Figure 5.1 on the left and right respectively represent the results of CL derivatives when $C_L$ and $C_D$ derivatives are directly determined as estimation parameters and the $C_L$ derivatives obtained by determining $C_X$ and $C_Z$ derivatives as estimation parameters and applying the aforementioned transformation equations. Likewise, in Figure 5.2, the graph on the right demonstrates the CD derivatives obtained by determining the $C_X$ and $C_Z$ derivatives as estimation parameters and applying the above transformation equations, while the graph on the left shows the results of $C_D$ derivatives when $C_L$ and $C_D$ derivatives are directly determined as estimation parameters.

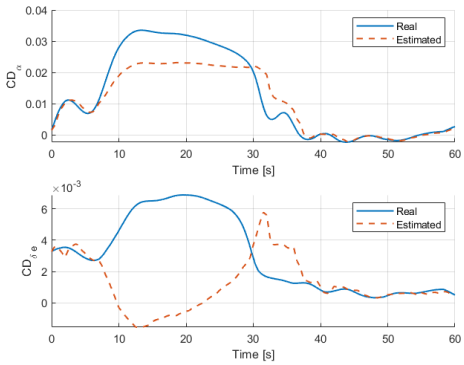



(a) When Estimation Parameters are $C_L$ and $C_D$

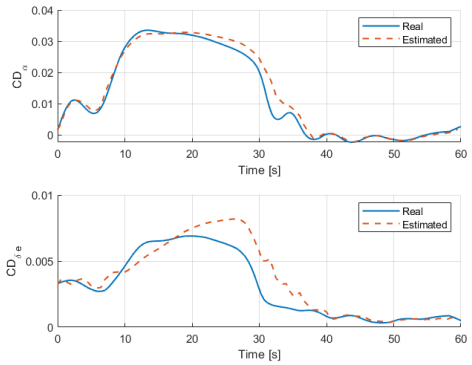



(b) When Estimation Parameters are $C_X$ and $C_Z$

Figure 5.1: Estimation Results of Lift Coefficient Derivatives




(a) When Estimation Parameters are $C_L$ and $C_D$




(b) When Estimation Parameters are $C_X$ and $C_Z$

Figure 5.2: Estimation Results of Drag Coefficient Derivatives

### 5.5.4 Multiple Shooting Approach

In the first results of the algorithm, it is realized that estimated parameters diverge from the real ones after the first ignition, although the total aerodynamic force and moment coefficients that are obtained by using the estimated aerodynamic derivatives are very close to the real ones.

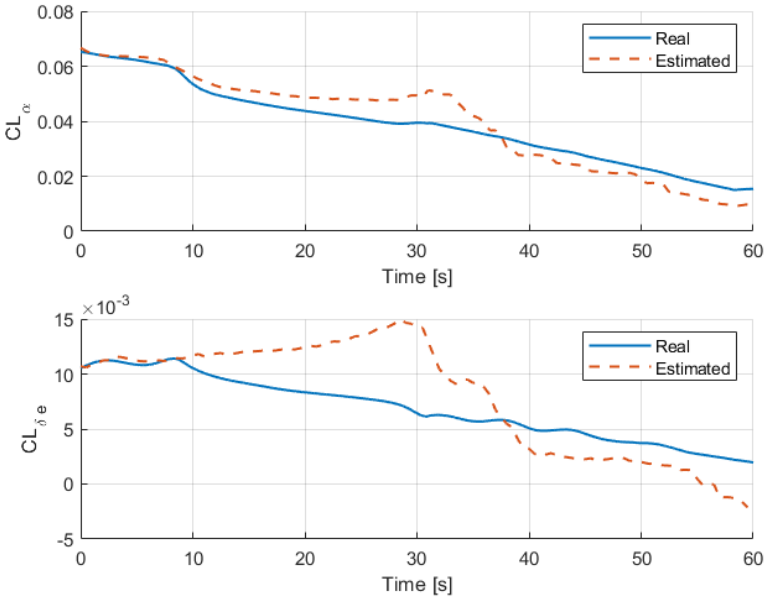



Figure 5.3: Estimation Results of Lift Coefficient Derivatives With Single Shooting

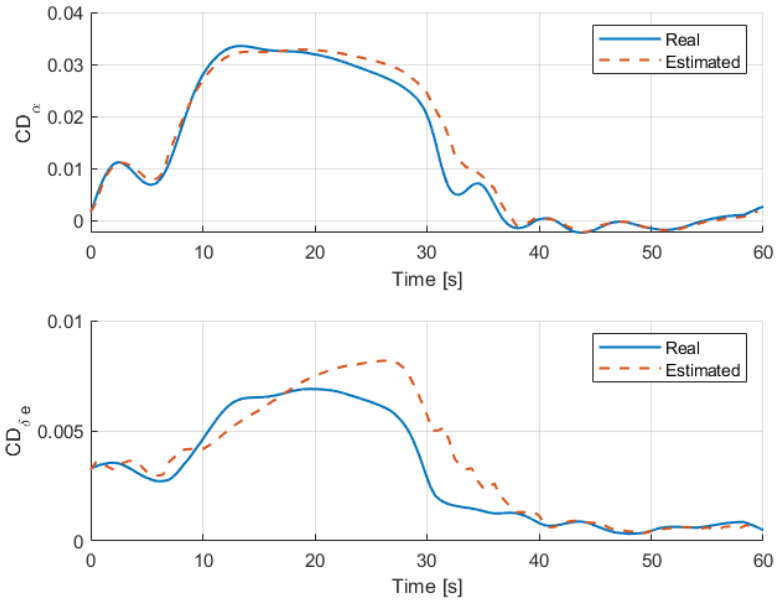



Figure 5.4: Estimation Results of Drag Coefficient Derivatives With Single Shooting

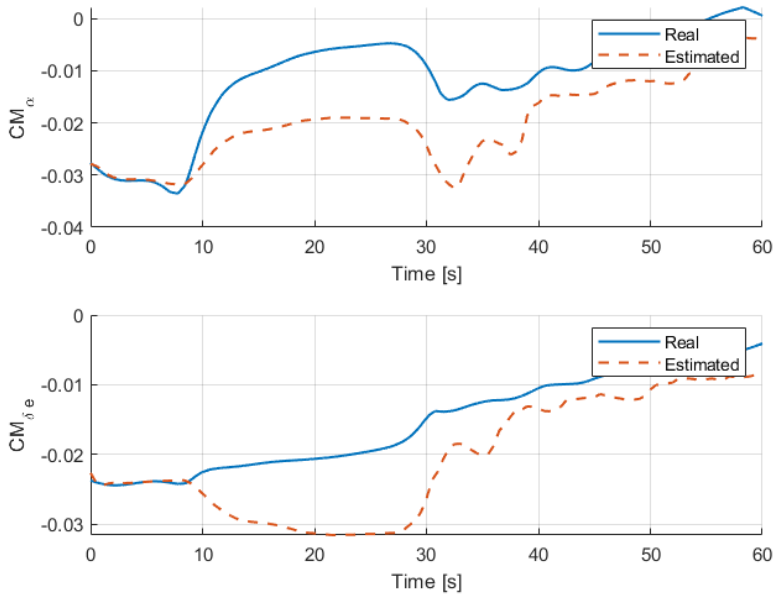



Figure 5.5: Estimation Results of Pitch Moment Coefficient Derivatives With Single Shooting

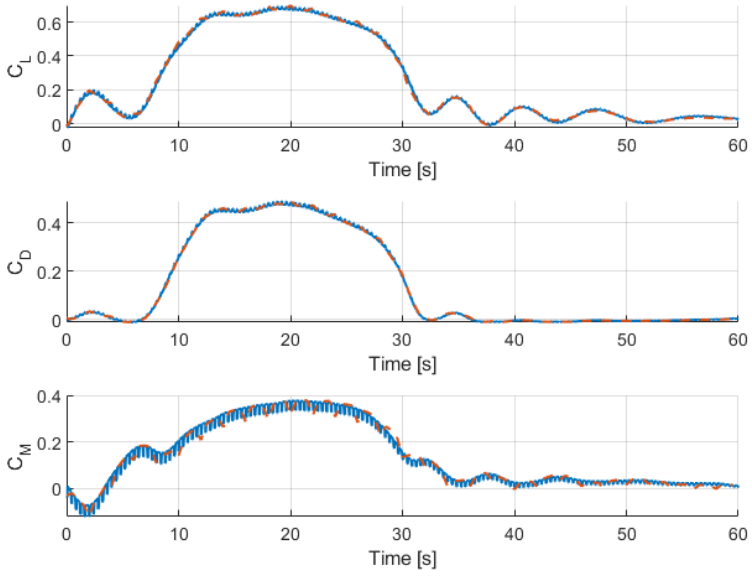



Figure 5.6: Estimation Results of Total Aerodynamic Coefficients With Single Shooting

It is evident in Figures 5.3, 5.4, and 5.5 that the estimated $C_L$, $C_D$, and $C_M$ derivatives diverge from real ones after a certain amount of time, although the fact that the estimated total aerodynamic coefficients and the true values overlap in Figure 5.6.

When one looks at the equations 5.17, 5.18 and 5.19, it can be seen that there is actually more than one aerodynamic derivative coefficient value that can provide accurate total aerodynamic force and moment coefficients. It can be said that the reason why the algorithm cannot estimate the derivatives, although it can estimate the total coefficients correctly, is that there is no unique solution.

The conventional output error technique for parameter estimation in the time domain brings numerical challenges due to the possibility of numerical divergence in simulation and optimization when numerical integration is used for systems that are extremely responsive and naturally unsteady, [13]. The problem that is faced in this study can be given as an example of numerical divergence. There are different modification methods for unstable aircraft identification problems according to used estimation approach. One of the suggested modifications for the output error method is the multiple shooting approach.

The "single shooting" approach is a commonly utilized strategy for resolving estimate issues that includes solving initial value problems iteratively to determine unknown parameters. By optimizing a scalar function, this method fails to utilize measured traces, which are the main information source for the estimating process. Alternatively, the multiple shooting method more effectively accounts for this supplied data. The multiple shooting strategy's fundamental idea is to split the integration interval $[t_0, t_f]$ into $(m-1)$ intervals by an appropriate grid selection :

$$g^m : \quad \{t_0 = \tau_1 < \tau_2 < \ldots < \tau_m = t_f\} \tag{5.45}$$

The associated mathematical system;

$$\dot{x}(t) = f(t, x, \Theta), \qquad x(\tau_j) = \sigma_j, \qquad t \in [\tau_j, \tau_{j+1}] \tag{5.46}$$

$x(\tau_j)$ can be separately solved using standard integration methods with the extra terms

$(\sigma_1, \ldots, \sigma_m)$ as estimations of the states. The following restrictions need to be met in order to guarantee the final solution's continuity across the whole period.

$$h_j(\sigma_j, \sigma_{j+1}, \Theta) = x(\tau_{j+1}|\sigma_j, \Theta) - \sigma_{j+1} = 0; \quad j = 1, \ldots, (m-1) \tag{5.47}$$

The cost function is tried to minimize by using the above constrained. Also, the unknown parameter vector is expanded as $\tilde{\Theta} = [\sigma_1^T, \sigma_2^T, \ldots, \sigma_m^T, \Theta^T]^T$.
By establishing extra grid points, the multiple shooting approach makes sure that the computed solution stays near the measured data. It is possible to eliminate the uncontrollably growing parasitic components when solving unstable differential equations. The choice of grid points has a significant impact on this method's convergence and success. Grid points can be automatically spaced by selecting an error growth tolerance limit, or they can be spaced evenly. In this study, grid points were chosen randomly within the $\mp 10\%$ of the real values, similar to the choosing initial guess values. Also, they are split by intervals of two seconds, meaning every 2000 data points.

## 5.6 Results

The findings of the aerodynamic parameter estimate utilizing the multiple shooting modified output-error approach are presented in this section. The data set from the 3-DOF model recorded at 1000 Hz was divided into 0.5-second intervals, and grid points were placed every 2 seconds to perform the output error estimation technique. First, the derivatives of the forces and moments in the body axis specified in accordance with the angle of attack and the angle of the control surface were determined through the algorithm. Figure 5.7 illustrates the derivative coefficients of the moments and forces in the body axis assembly. By looking at this image, it is possible to conclude that the predicted values are quite near to the actual ones.

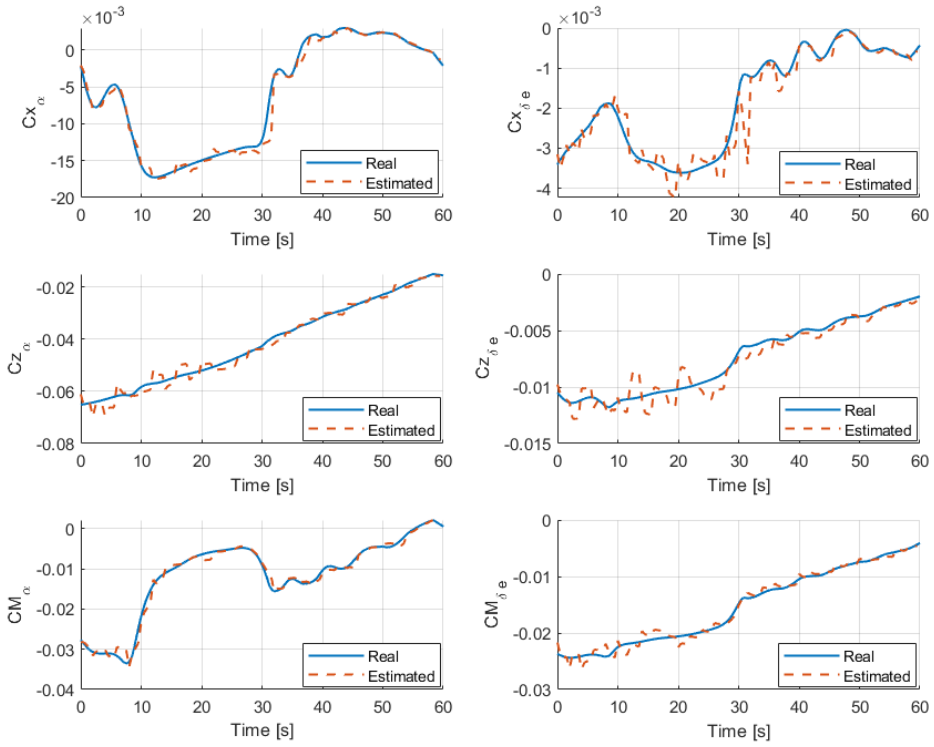



Figure 5.7: Estimation Results in Body Frame

The estimated parameters in the body coordinate frame are transferred to the wind axis coordinate frame at the end of the estimation process. While the force derivative coefficients on the x and z axes transform into the derivative coefficients of the lift and drag forces, the pitch moment derivative coefficients remain the same. The result graphs of lift, drag, and pitch moment derivative coefficients are shown in figures 5.8, 5.9 and 5.10 respectively.

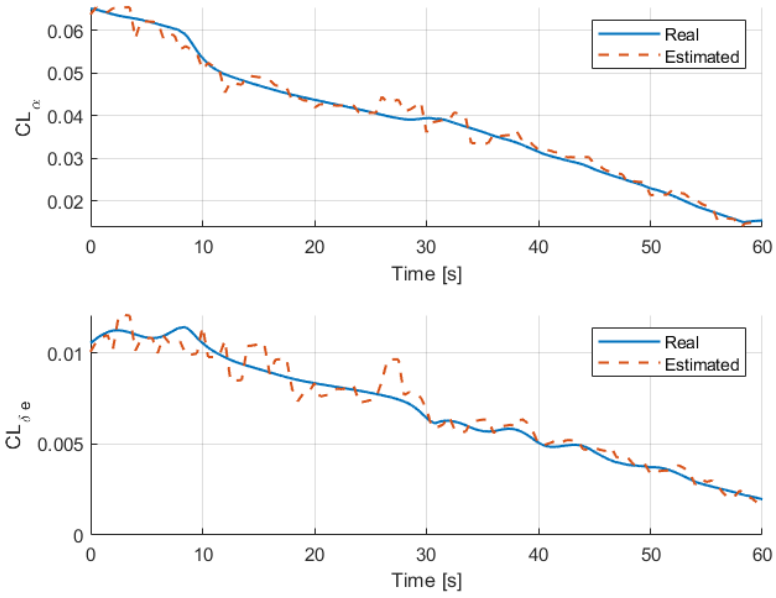



Figure 5.8: Estimation Results of Lift Coefficient Derivatives

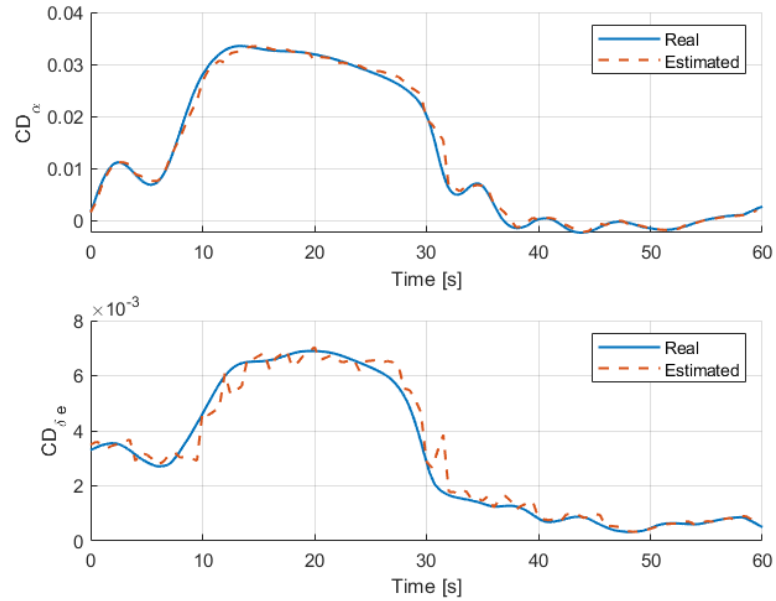



Figure 5.9: Estimation Results of Drag Coefficient Derivatives

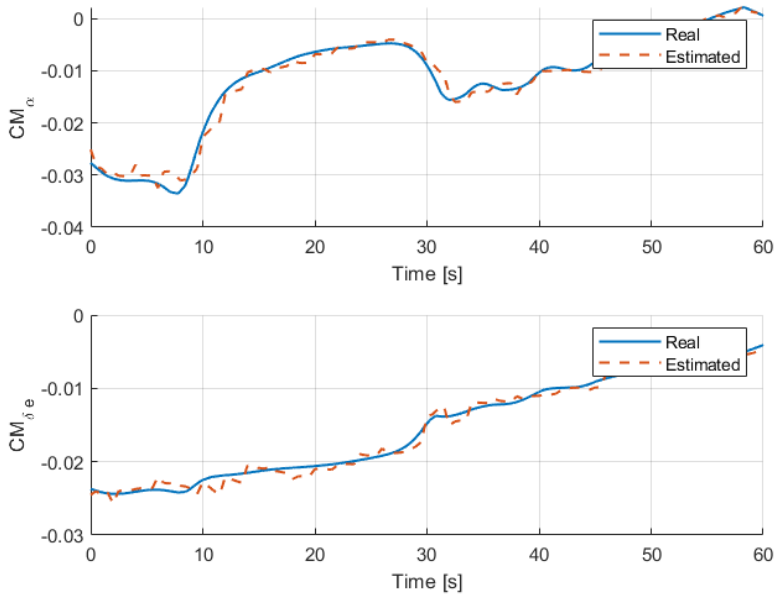



Figure 5.10: Estimation Results of Pitch Moment Coefficient Derivatives

It is apparent from the above figures that the estimated values are quite close to the actual values. It can also be seen that the issue of estimating results deviating from true values in regions where $\alpha$ and $\delta_e$ fluctuate quickly —observed in the result graphs of prior methods— has been resolved using the output-error technique. The output error method's iterative approach to minimizing the error between the estimated and real values improves the outcomes.

The computed aerodynamic derivative coefficients were then used to calculate the overall aerodynamic force and moment coefficients via $\alpha$ and $\delta_e$. Figure 5.11 represents the computed total aerodynamic force and moment coefficients together with their actual values.

To further demonstrate the effectiveness of the output-error estimating method, the mean-squared error (MSE) value between the actual values and the anticipated outcomes has been calculated for each coefficient. The MSE values of the total aerodynamic coefficients and their derivatives to the angle of attack and control surface angle are displayed in Table 5.1.

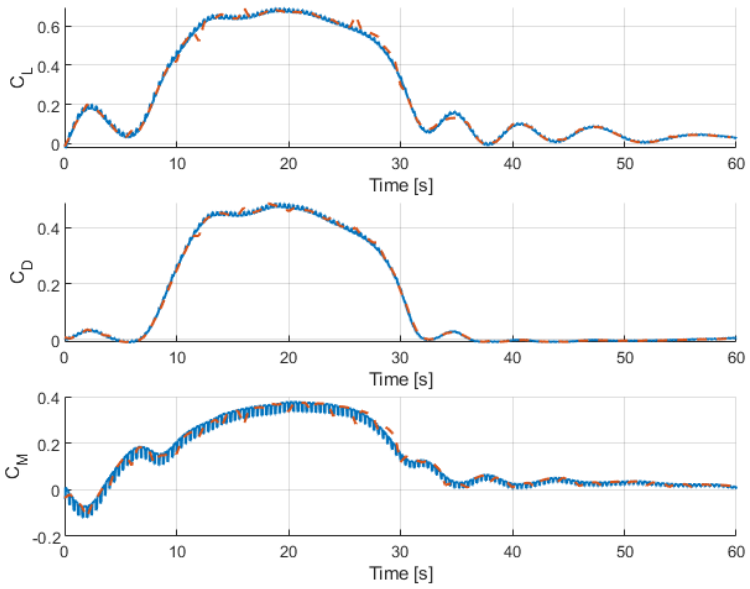



Figure 5.11: Results of Total Aerodynamic Force and Moment Coefficients

Table 5.1: Mean-Squared Errors of Output Error Method Results

| Parameter | MSE | Parameter | MSE | Parameter | MSE |
|---|---|---|---|---|---|
| $C_L$ | $3.44 \times 10^{-4}$ | $C_D$ | $6.56 \times 10^{-5}$ | $C_M$ | $1.31 \times 10^{-4}$ |
| $C_{L_\alpha}$ | $4.13 \times 10^{-6}$ | $C_{D_\alpha}$ | $1.18 \times 10^{-6}$ | $C_{M_\alpha}$ | $1.20 \times 10^{-6}$ |
| $C_{L_{\delta e}}$ | $6.61 \times 10^{-7}$ | $C_{D_{\delta e}}$ | $1.93 \times 10^{-7}$ | $C_{M_{\delta e}}$ | $6.71 \times 10^{-7}$ |

All derivatives in the above figures and table are in the unit of $1/^\circ$.

Looking at Table 5.1, it can be seen that the output-error method has quite small MSE values. The results of this table can be roughly compared with the results of the previous methods, which are the least-square method and the method where neural network and least-square methods are used together. Upon comparing tables 3.2, 4.1, and 5.1, it can be concluded that, in terms of total aerodynamic force and moment coefficients, the output error approach performs slightly worse than the other two techniques. On the other hand, the output error method yields about 20 times better prediction results than the least-square method and about 10 times better results than the combination of neural network and least-square method when examining the aerodynamic force and moment derivative coefficients. Chapter 6 will compare these three approaches in further detail using various data sets.

# CHAPTER 6

# MONTE CARLO RESULTS AND DISCUSSIONS

This chapter covers analyzing and interpreting the estimated findings of aerodynamic derivative coefficients using least square, neural network least square combination, and output error approaches. Initially, 100 runs of the 3-DOF launch vehicle model were conducted using a developed Monte Carlo analysis, and the results were recorded. Following that, prediction algorithms were applied to these collected data sets, and the mean squared error values were determined for each run of the algorithms.

## 6.1 Monte Carlo Analysis

The estimation findings utilizing a sample flight data set are presented in the results sections of the preceding chapters. To better interpret and compare the performance of estimation algorithms, Monte Carlo simulations were carried out under different uncertainties to produce data sets under different flight conditions. System variables, including mass, thrust, center of gravity distance, aerodynamic force, and moment coefficients, are all subject to uncertainty in Monte Carlo simulations. Table 6.1 lists these parameters and the corresponding uncertainty rates. The highest possible values of uncertainty are represented by the values in the table. For the mass, thrust, aerodynamic force, and moment coefficients, a random uncertainty is created in the percentage range denoted as $(+/-)$; the nominal value is multiplied by this number in the simulation. Also, a random uncertainty value is decided in the range of $+/-0.5$ m for the center of gravity, and it is added to the nominal value in the simulation.

Table 6.1: Monte Carlo Parameters

| Parameter | Uncertainty |
|---|---|
| $C_{L_{unc}}$ | 10% |
| $C_{D_{unc}}$ | 10% |
| $C_{M_{unc}}$ | 10% |
| $m_{unc}$ | 5% |
| $T_{unc}$ | 20% |
| $x_{cg_{unc}}$ | 0.5 m |

## 6.2 Aerodynamic Parameter Estimation Results for Monte Carlo Simulations

After conducting the Monte Carlo simulations, the aerodynamic derivative coefficients were estimated using the least squares (LSE), neural network-least squares combination (NN + LSE), and output error (OEM) techniques for each run. The mean-squared errors of each parameter were calculated and recorded for each data set through the estimation algorithms. Figures 6.1, 6.2, and 6.3 show the MSE distributions of the three techniques related to changing aerodynamic lift, drag, and pitch moment according to both angle of attack and elevator deflection.

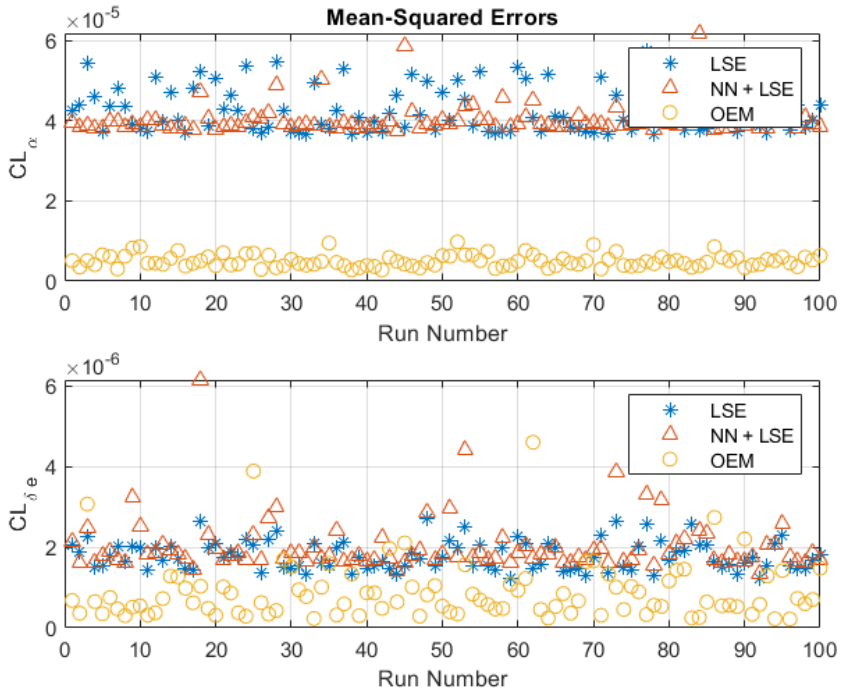



Figure 6.1: Mean-Squared Errors of Lift Coefficient Derivatives

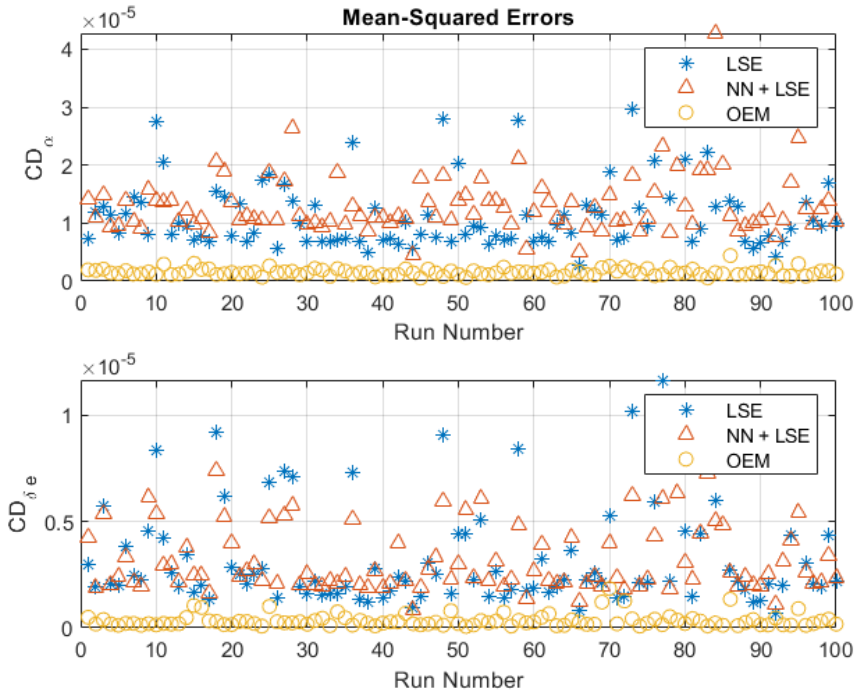



Figure 6.2: Mean-Squared Errors of Drag Coefficient Derivatives

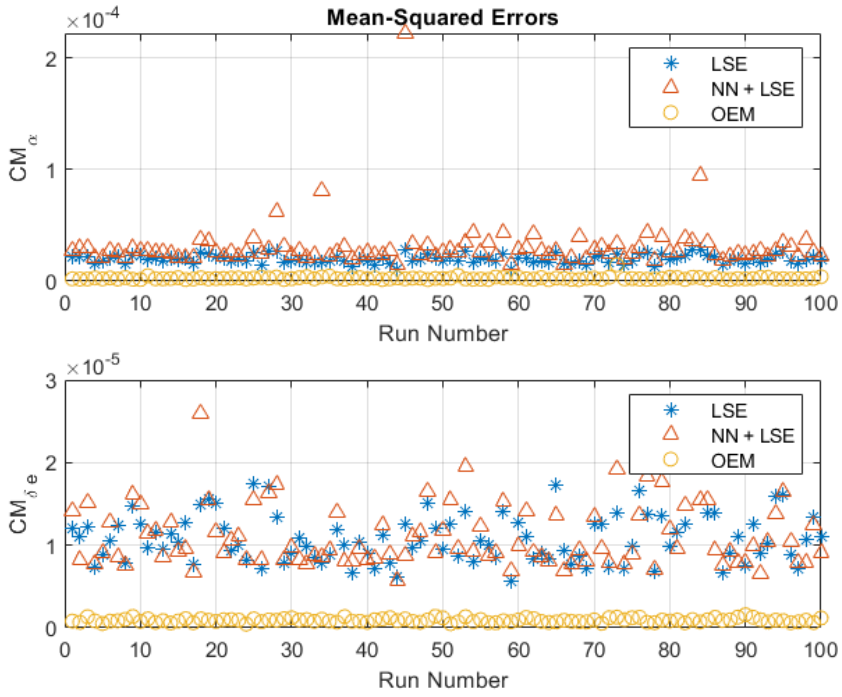



Figure 6.3: Mean-Squared Errors of Pitch Moment Coefficient Derivatives

It is evident from the MSE values in Figure 6.1 that, in comparison to the other two approaches, the output error method's MSE values for $C_{L\alpha}$ and $C_{L\delta_e}$ are often significantly lower. The MSE distributions of the two approaches are much closer to one another in both derivative estimation performances when comparing the outcomes of the least-squares technique and the neural network and least-squares combination approach. Though the findings of the $C_{L\alpha}$ derivatives show that the MSE distribution of LSE is substantially broader, the $C_{L\delta_e}$ derivative results show that the NN + LSE technique has greater MSE values than other methods for a few cases.

Looking at Figure 6.2, it is clearly seen that the MSE values of the OEM are much smaller, and the MSE distribution is much narrower. Furthermore, the MSE values of the $C_D$ derivative estimations of NN + LSE and LSE may be stated to be closer to the OEM's MSE values than they are in Figure 6.1. However, NN + LSE and LSE methods have wide MSE distributions for both $C_{D\alpha}$ and $C_{D\delta_e}$, which means that the results of these two methods have larger differences compared to OEM. Compared to the MSE distributions of the NN + LSE and LSE methods, since the distribution of the NN + LSE method is narrower, it can be said that NN + LSE produces slightly more

consistent results than LSE. In general, looking at this figure, it can be concluded that OEM is slightly closer to reality and has more consistent estimation results than the other two methods.

When Figure 6.3 is viewed, it is seen that the MSE values of OEM are smaller, and their dispersion is narrower than in the previous two pictures. The values obtained using the LSE approach are less than the values obtained from the NN + LSE, and the distribution is narrower when we examine the MSE values of the $C_{M\alpha}$ derivative estimation. Thus, it can be stated that for the $C_{M\alpha}$ derivative, the LSE approach yields more precise and reliable forecasts than the NN + LSE technique. It is clear that the MSE values of the LSE technique are slightly lower than the MSE values of the NN + LSE method, although the distributions of these two approaches appear similar in the $C_{M\delta_e}$ graph. Regarding prediction performance, Figure 6.3 puts in order the three techniques as OEM, LSE, and NN + LSE.

Furthermore, we may evaluate MSE values by looking at the following box charts.

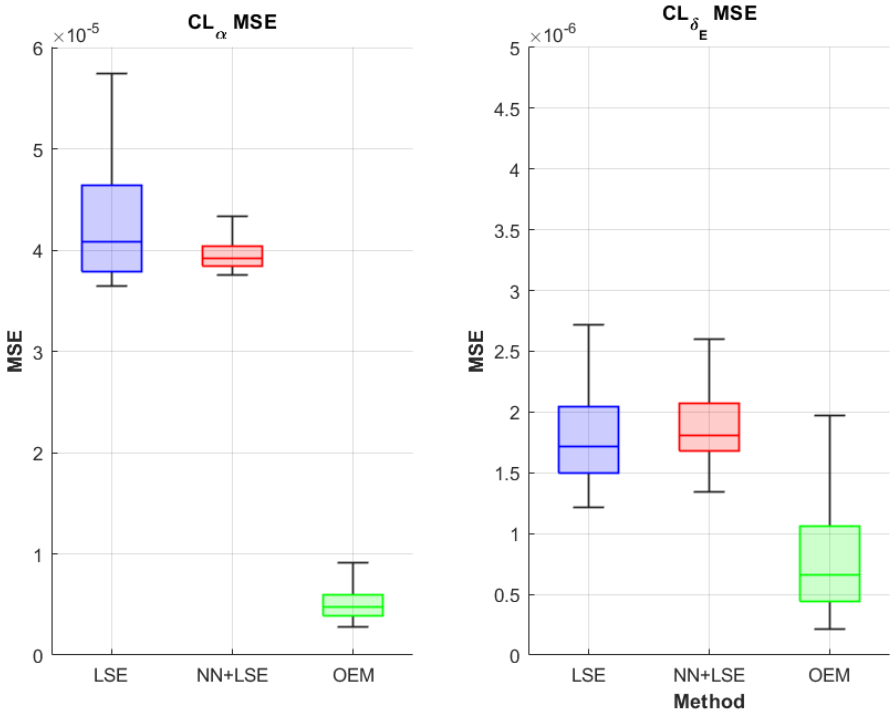



Figure 6.4: Mean-Squared Errors of Lift Coefficient Derivatives

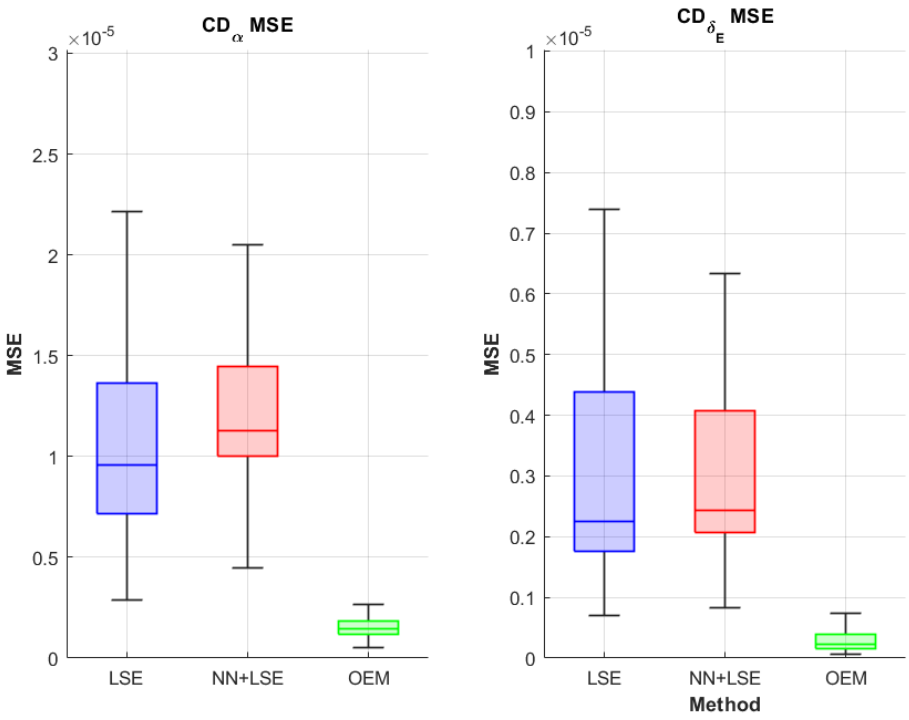



Figure 6.5: Mean-Squared Errors of Drag Coefficient Derivatives

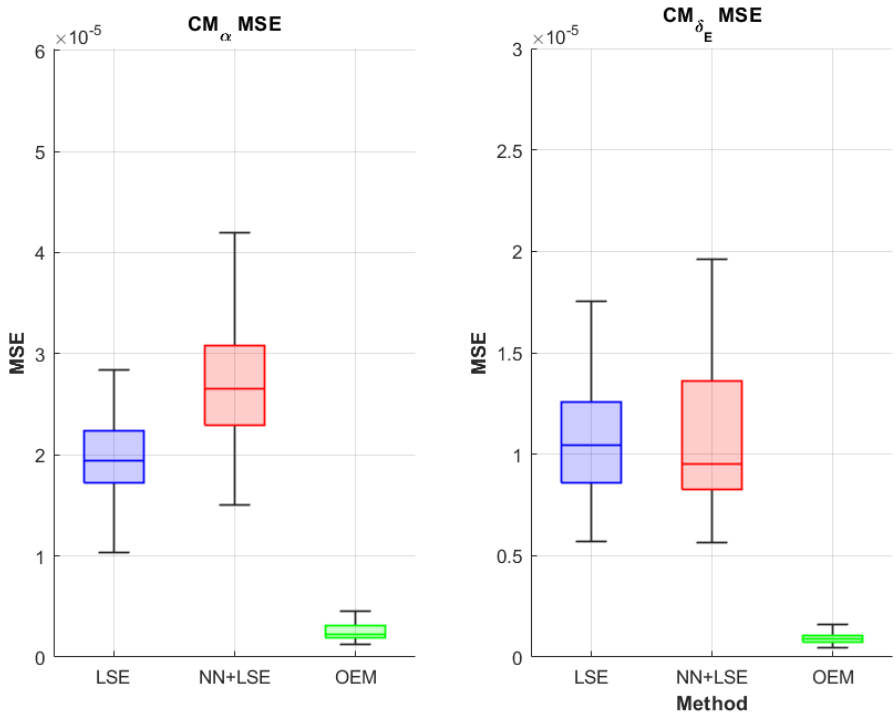



Figure 6.6: Mean-Squared Errors of Pitch Moment Coefficient Derivatives

In Figure 6.4, OEM stands out with the lowest MSE values in $C_L$ derivatives measurements, which shows that it is the most stable and reliable method. Performance between LSE and NN+LSE is similar, but LSE offers a slightly lower median error for $C_{L\delta E}$.

Figure 6.5 demonstrates that the OEM consistently performs well for $C_D$ measures, with the lowest MSE values and smallest ranges. While LSE provides acceptable consistency, particularly for $C_{D\alpha}$, the larger ranges of NN+LSE indicate that this approach exhibits greater variance in these measurements.

Figure 6.6 demonstrates that the OEM has the lowest MSE values and smallest distribution ranges for $C_M$ measures, indicating that it performs better. Although LSE has low median MSE values for $C_M$ measurements, broader ranges show greater variability. It can be observed that in both $C_M$ measurements, NN+LSE shows larger ranges and higher error levels.

In addition to the above results, the averages and standard deviations of the mean-squared errors of the three methods were calculated. Table 6.2 shows the averages of the mean-squared error values for each of the three approaches for various parameter values, and table 6.3 includes their standard deviations.

Table 6.2: Means of Mean-Squared Errors

| Parameter | LSE | NN + LSE | OEM |
|---|---|---|---|
| $C_{L_\alpha}$ | $4.25 \times 10^{-5}$ | $4.01 \times 10^{-5}$ | $5.07 \times 10^{-6}$ |
| $C_{L_{\delta e}}$ | $1.78 \times 10^{-6}$ | $2.03 \times 10^{-6}$ | $8.92 \times 10^{-7}$ |
| $C_{D_\alpha}$ | $1.19 \times 10^{-5}$ | $1.30 \times 10^{-5}$ | $1.55 \times 10^{-6}$ |
| $C_{D_{\delta e}}$ | $3.41 \times 10^{-6}$ | $3.14 \times 10^{-6}$ | $3.40 \times 10^{-7}$ |
| $C_{M_\alpha}$ | $1.98 \times 10^{-5}$ | $3.07 \times 10^{-5}$ | $2.59 \times 10^{-6}$ |
| $C_{M_{\delta e}}$ | $1.09 \times 10^{-5}$ | $1.10 \times 10^{-5}$ | $9.16 \times 10^{-7}$ |

In the tables 6.2 and 6.3, a small mean square value indicates that the estimated result is closer to reality, and a small standard deviation of the mean square values indicates that the outcomes of the estimation method are reliable and consistent.

Table 6.3: Standart Deviations of Mean-Squared Errors

| Parameter | LSE | NN + LSE | OEM |
|---|---|---|---|
| $C_{L_\alpha}$ | $5.52 \times 10^{-6}$ | $3.74 \times 10^{-6}$ | $1.55 \times 10^{-6}$ |
| $C_{L_{\delta e}}$ | $3.56 \times 10^{-7}$ | $6.93 \times 10^{-7}$ | $7.28 \times 10^{-7}$ |
| $C_{D_\alpha}$ | $6.75 \times 10^{-6}$ | $5.06 \times 10^{-6}$ | $6.10 \times 10^{-7}$ |
| $C_{D_{\delta e}}$ | $2.54 \times 10^{-6}$ | $1.49 \times 10^{-6}$ | $3.07 \times 10^{-7}$ |
| $C_{M_\alpha}$ | $3.83 \times 10^{-6}$ | $2.23 \times 10^{-5}$ | $1.45 \times 10^{-6}$ |
| $C_{M_{\delta e}}$ | $3.10 \times 10^{-6}$ | $3.75 \times 10^{-6}$ | $2.36 \times 10^{-7}$ |

Looking at the mean and standard deviation values of the $C_{L_\alpha}$ parameter, it can be said that OEM gives the best and most consistent result. When the other two methods are compared, although the MSE average values of the two methods are very close to each other, it can be said that the NN+LSE approach performs slightly better than the LSE method. Looking at the standard deviation values, it can be concluded that the NN+LSE approach has a more uniform performance in $C_{L_\alpha}$ estimation.

When the mean and standard deviation values for $C_{L_{\delta e}}$ estimation are examined, it is seen that they are approximately 10 times smaller than the values in $C_{L_\alpha}$ for all three methods. According to the mean values of $C_{L_{\delta e}}$, it can be said that OEM is better in terms of prediction success and the NN+LSE method has a slightly worse performance than the other two methods. When looking at the standard deviation values, it can be said that the LSE method has the narrowest distribution, while the OEM and NN+LSE methods have wider MSE distributions. In terms of consistency, the LSE method gave better results in $C_{L_{\delta e}}$ estimation.

The LSE and NN+LSE approaches are comparable in terms of prediction success, while OEM outperforms both by a factor of ten, according to the average MSE values for $C_{D_\alpha}$. In terms of uniformity, the approaches may be ranked as OEM, NN+LSE, and LSE when examining the standard deviations of MSE values.

For all three techniques, $C_{D_{\delta e}}$ predicts more accurately and consistently than $C_{D_\alpha}$, which is similar to the outcome observed in $C_L$ derivatives. The approaches are classified as OEM, NN+LSE, and LSE based on prediction success and consistency of outcomes.

It is possible that it can be inferred from the $C_{M_\alpha}$ and $C_{M_{\delta e}}$ findings that the output-error approach yields more accurate and realistic results than the other two methods, like the $C_L$ and $C_D$ derivatives. It is also possible to argue that the least-squares approach has more efficacy and stability than the combination of neural networks and least squares. The following is a list of the three approaches' success rates and consistency in predicting $C_M$ derivatives: OEM, LSE, and NN+LSE.

It is essential to evaluate estimating techniques in terms of computation time in addition to their accuracy and consistency of estimation findings. In this thesis research, a computer with an AMD Ryzen 3600x processor, 16 GB capacity 3200 MHz DDR4 RAM, and a Samsung 980 Pro 1 TB hard disk was used.

The least squares estimation technique can generate estimation results for a data set in around 3.5 seconds. Approximately 6 minutes is required for the LSE algorithm to execute on 100 data sets.

Estimating a data set using the neural network and least squares combination technique takes about 6.75 seconds. Using three data sets and the techniques outlined in Chapter 4.2, it takes about an hour to train the neural model that will be utilized in the prediction algorithm; however, this step is only carried out once. Making predictions for 100 data sets using NN + LSE takes around 12 minutes, excluding training time.

The output error procedure, which includes the Gauss-Newton optimization method, takes around 20 minutes to forecast a data set. This method takes longer to calculate than others since it uses an iterative approach. It takes approximately 36.5 hours to make an estimation for 100 data sets with the output error method, but this duration can be reduced by parallelizing the process. For example, in this study, this duration was reduced to approximately 17 hours by using more than one Matlab simultaneously.

When comparing the running times of these three approaches, it is evident that OEM has the largest computational cost. If the training time of the neural network is excluded, although the running times of the NN+LSE and LSE methods are close to each other, it can be said that the fastest method is LSE.

As a result, if a general comparison is made, using the output error method for aero-

dynamic parameter estimation gives much more accurate and consistent results than the other two methods. However, the disadvantage of OEM compared to others is that its computation cost is high. OEM may be preferred in cases where computational cost is not very important and where exact solutions are desired for systems with sensitive system dynamics.

It can be said that the neural network – least squares combination estimation approach and the least square method are not very unsuccessful and inconsistent in terms of the performance of the estimation results. It can be said that the performances and computation costs of both methods are close to each other. Although both methods use the least-squares method as an algorithm to estimate aerodynamic derivatives, total aerodynamic force moments are obtained from measurement data using mathematical equations in the LSE algorithm, while these values are obtained by feeding the measurement data to the neural model in the NN + LSE approach. For this reason, the NN + LSE method can be preferred for systems with more complex system dynamics that are more difficult to formulate mathematical models for. At this point, the computational cost factor should also be taken into consideration.

On the other hand, if one has simpler system dynamics in the estimation problem and wants to get a faster solution, the LSE algorithm can be chosen.

# CHAPTER 7

# CONCLUSIONS AND FUTURE STUDIES

## 7.1 Conclusions

This thesis study focuses on the application of three different system identification methods for the aerodynamic parameter estimation of a launch vehicle and the comparison of these methods. After doing an adequate review of the literature on the topic, it was determined that the Pegasus LV would be the vehicle used to carry out the approach.

A mathematical model was created using the Pegasus vehicle's characteristic features and aerodynamic database. A three-degree-of-freedom (3-DOF) model was created in the Simulink environment since it would be sufficient to apply the aerodynamic parameter estimation only on the longitudinal axis. This model has been used to generate true values and collect measurement data for the estimation algorithms.

Following the discussion of the theory of least squares technique, one of the aerodynamic parameter estimation methods, input design research was conducted to determine the right model control input to enhance the functioning of this approach. After deciding on the proper control input to the system, the least squares approach was used to predict the aerodynamic coefficients of Pegasus LV. When the algorithm results were examined, it was discovered that, although the predicted values deviated slightly from the real values in regions where the angle of attack and control surface angle changed rapidly, the predicted values were very close to the real values for the rest of the trajectory.

The combination of the neural network technique and the least squares method was utilized as the second method for aerodynamic parameter estimation. The feed-forward neural network technique was used to create a neural model structure that

would extract the overall aerodynamic force and moment coefficients. This model was then trained using various data sets. Aerodynamic derivative coefficients were identified using the least squares approach with these neural model outputs after the accuracy of the trained model was confirmed. The outcomes of this method were then investigated using the sample data set. Except for regions where the angle of attack and control surface angle vary drastically, it has been shown that the NN+LSE approach offers acceptable predictions, very comparable with the results obtained by the least squares method.

Aerodynamic derivative parameters were discovered using the output-error approach, which is an iterative method, as the third approach. The maximum likelihood function was written by using the error between measurements and the outputs of the mathematical model. Then, this function was optimized using the Gauss-Newton optimization technique. The force derivatives in the body axis set were estimated and afterward transformed to the wind axis set to reduce the non-linearity in the mathematical model. Furthermore, the output error technique was first implemented using the single-shooting method; however, this typical OEM methodology resulted in a numerical divergence problem due to the unsteady nature of the system. As a result, the OEM algorithm had to be modified to include the multiple-shooting methodology. It may be stated that this approach produces better prediction outcomes than the other two identifying methods.

Finally, 100 separate data sets were recorded using Monte Carlo simulations, and aerodynamic parameter identification was performed using three alternative approaches for each data set. The mean squared error value was computed in each run, and the three approaches were compared using the averages and standard deviations of these values. One can conclude that OEM may be the better option when accurate estimations are needed for systems with sensitive system dynamics, and computation expense is not a major consideration. Furthermore, it may be argued that the NN + LSE approach might be preferred for systems with complex dynamics that are difficult to explain mathematically. On the other hand, when dealing with simpler system dynamics in the estimation problem and seeking a quicker solution, the LSE algorithm could be a preferable choice.

## 7.2 Future Studies

When the results of the LSE and NN+LSE algorithms are compared, they are found to be quite similar. The distinction between these two algorithms is that one uses mathematical equations and the other a neural model. Neural models are expected to be more beneficial in systems with complex dynamics whose mathematical expressions are written more difficultly. For this reason, by integrating disturbances such as gusts, including bending dynamics, and increasing system uncertainties, the 3-DOF model system designed to gather data may be transformed into more complicated in further research. Furthermore, the performance of the NN+LSE algorithm can be examined for this more complex model. Also, data more similar to actual flight data may be gathered in this manner.

It may be possible to improve the algorithm in further research to speed up the OEM algorithm. A heuristic application was used in this study to update the parameter by examining changes in cost. To optimize the parameter update procedures in further research, the dominant directions or line search approaches in reference [14] might be attempted. In addition, time windows were determined at fixed intervals in this study. Future studies can develop an algorithm that employs the sliding window approach and determines time intervals automatically. Additionally, a strategy that combines the neural network and output error method described in references [30] or [29] can be tested for aerodynamic parameter estimation.

# REFERENCES


[1] Charu C Aggarwal. *Neural Networks and Deep Learning*. Springer, 2018.

[2] A. Aksu. "Aerodynamic Parameter Estimation of a Missile". In: (2013). MS Thesis, METU.

[3] Marie Albisser et al. "Aerodynamic coefficient identification of a space vehicle from multiple free-flight tests". In: *Journal of Spacecraft and Rockets* 54 (2 2017), pp. 426–435. ISSN: 15336794. DOI: 10.2514/1.A33587.

[4] Vincenzo D ' Antuono et al. *ESTIMATION OF AERODYNAMIC ANGLES AND WIND COMPONENTS FOR A LAUNCH VEHICLE*. URL: https://www.researchgate.net/publication/363671804.

[5] Md. Ashek-Al-Aziz and Abdullah-Hil Muntakim. "Performances of Various Back-propagation Learning Algorithms of Neural Network Using Matlab". In: *International Journal of Sciences: Basic and Applied Research (IJSBAR)* 57.2 (2021), pp. 112–136. ISSN: 2307-4531. URL: http://gssrr.org/index.php?journal=JournalOfBasicAndApplied.

[6] G. Aygın. "Aerodynamic Parameter Estimation and Validation with Flight Data in Closed-Loop Control of a Missile". In: (2012). MS Thesis, METU.

[7] T. Bayoğlu. "Kalman Filtering for Aerodynamic Parameter Estimation in a Supersonic Missile with High Rate Dynamics". In: (2016). MS Thesis, METU.

[8] Robert E Curry, Robert R Meyer, and Gerald D Budd. *Pegasus Hypersonic Flight Research*. 1992. URL: https://www.jstor.org/stable/44733138.

[9] Howard Demuth and Mark Beale. *Computation Visualization Programming Neural Network Toolbox For Use with MATLAB ® User's Guide*. 1992. URL: http://www.mathworks.com.

[10] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*.

[11] Martin T. Hagan and Mohammad B. Menhaj. “Training Feedforward Networks with the Marquardt Algorithm”. In: *IEEE Transactions on Neural Networks* 5.6 (1994), pp. 989–993. DOI: 10.1109/72.329697.

[12] R. V. Jategaonkar and E. Plaetschke. “Maximum Likelihood Estimation of Parameters in Nonlinear Flight Mechanics Systems”. In: *IFAC Proceedings Volumes* 18 (5 July 1985), pp. 663–668. ISSN: 1474-6670. DOI: 10.1016/S1474-6670(17)60636-0.

[13] R. V. Jategaonkar and F. Thielecke. “Evaluation of parameter estimation methods for unstable aircraft”. In: *Journal of Aircraft* 31 (3 1994), pp. 510–519. ISSN: 00218669. DOI: 10.2514/3.46523.

[14] Ravindra V. Jategaonkar. *Flight Vehicle System Identification*. American Institute of Aeronautics and Astronautics, Aug. 2006. DOI: 10.2514/4.866852.

[15] Jian Jia, Weifeng Chen, and Zixuan Wang. “Aerodynamic Parameter Estimation for Launch Vehicles”. In: 2023, pp. 201–213. DOI: 10.1007/978-981-99-0613-0_7.

[16] R. Kumar and A. K. Ghosh. “Nonlinear Longitudinal Aerodynamic Modeling Using Neural Gauss-Newton Method”. In: *Journal of Aircraft* 48.5 (2011). DOI: 10.2514/1.C031253.

[17] M. Lavagna and F. Venditti. “An air-launched light satellite launcher design”. In: (2010). MS Thesis, POLITECNICO DI MILANO.

[18] Seungmin Lee, Shinseong Kang, and Kyunghoon Lee. “Development of a Gaussian process modeling application for the prediction of missile aerodynamic coefficients”. In: *Journal of Mechanical Science and Technology* 35 (3 Mar. 2021), pp. 987–997. ISSN: 19763824. DOI: 10.1007/s12206-021-0212-1.

[19] M. Majeed and Indra Narayan Kar. “Aerodynamic parameter estimation using adaptive unscented Kalman filter”. In: *Aircraft Engineering and Aerospace Technology* 85 (4 2013), pp. 267–279. ISSN: 00022667. DOI: 10.1108/AEAT-Mar-2011-0038.

[20] Donald W Marquardt. *An Algorithm for Least-Squares Estimation of Nonlinear Parameters*. 1963.

[21] *MATLAB Deep Learning Toolbox: trainlm Function*. https://www.mathworks.com/help/deeplearning/ref/trainlm.html?searchHighlight=trainlm&s_tid=srchtitle_support_results_1_trainlm.

[22] M. R. Mendenhall et al. "Aerodynamic Analysis of Pegasus - Computations vs Reality". In: (1993).

[23] Eugene Morelli, Stephen Derry, and Mark Smith. "Aerodynamic Parameter Estimation for the X-43A (Hyper-X) from Flight Data". In: vol. 1. Aug. 2005. ISBN: 978-1-62410-055-0. DOI: 10.2514/6.2005-5921.

[24] M. Mosier et al. "Pegasus' First Mission - Flight Results". In: (1990).

[25] G. K. Noffz et al. "Aerothermal Test Results From the First Flight of the Pegasus Air-Launched Space Booster". In: (1991).

[26] "Notes of Flight Mechanics - ASE 372". In: (2019). Lecture Notes.

[27] *Orion Motor Series*. Tech. rep. Northrop Grumman, 2016.

[28] Bülent Özkan. *DYNAMIC MODELING, GUIDANCE, AND CONTROL OF HOMING MISSILES*. Phd Thesis, METU. 2005.

[29] N. K. Peyada and A. K. Ghosh. "Aircraft parameter estimation using a new filtering technique based upon a neural network and Gauss-Newton method". In: *The Aeronautical Journal* 113.1142 (Apr. 2009).

[30] N. K. Peyada and A. K. Ghosh. "Aircraft Parameter Estimation using Neural Network-based Algorithm". In: *AIAA Atmospheric Flight Mechanics Conference*. American Institute of Aeronautics and Astronautics. Chicago, Illinois, 2009, pp. 1–8.

[31] N. K. Peyada and H. O. Verma. "Estimation of longitudinal aerodynamic parameters using recurrent neural network". In: *The Aeronautical Journal* (Apr. 2022), pp. 255–267. DOI: 10.1017/aer.2022.39.

[32] M. N. Rhode, W. C. Engelund, and M. R. Mendenhall. "Experimental Aerodynamic Characteristics of the Pegasus Air-Launched Booster and Comparisons with Predicted and Flight Results". In: (1995).

[33] D. Rovner. “GNC for Pegasus® Air-Launched Space Booster: Design and First Flight Results”. In: (1991).

[34] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. “Learning Internal Representations by Error Propagation”. In: *Parallel Distributed Processing: Explorations in the Microstructure Cognition, Volume 1: Foundations*. Cambridge, MA: The MIT Press, 1986, pp. 318–362.

[35] K. G. Sheela and N. D. Subramaniam. “Review on methods to fix number of hidden neurons in neural networks”. In: *Mathematical Problems in Engineering* (2013). DOI: 10.1155/2013/425740.

[36] E. Süli and D. Mayers. *An Introduction to Numerical Analysis*. Cambridge University, 2003.

[37] TurboSquid. *Pegasus XL Rocket 3D Model*. https://www.turbosquid.com/3d-models/pegasus-xl-rocket-3d-model-1740285. Online; accessed [Date of Access]. Accessed 2024.

[38] “U.S. Standard Atmosphere, 1976”. In: (1976).