ISTANBUL TECHNICAL UNIVERSITY ★ GRADUATE SCHOOL

A COMPARATIVE STUDY ON CONVENTIONAL AND ENERGY-BASED AUTOPILOT SYSTEMS ON A JET-POWERED AIRCRAFT

M.Sc. THESIS

Rumeysa KÖKOĞLU

Department of Control and Automation Engineering

Control and Automation Engineering Programme

JUNE 2025

ISTANBUL TECHNICAL UNIVERSITY ★ GRADUATE SCHOOL

A COMPARATIVE STUDY ON CONVENTIONAL AND ENERGY-BASED AUTOPILOT SYSTEMS ON A JET-POWERED AIRCRAFT

M.Sc. THESIS

Rumeysa KÖKOĞLU
(504211117)

Department of Control and Automation Engineering

Control and Automation Engineering Programme

Thesis Advisor: Prof. Dr. Müjde GÜZELKAYA

JUNE 2025

ISTANBUL TEKNİK ÜNİVERSİTESİ ★ LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ

GELENEKSEL VE ENERJİ TABANLI OTOPİLOT SİSTEMLERİNİN JET MOTORLU BİR HAVA ARACINDA KARŞILAŞTIRMASI

YÜKSEK LİSANS TEZİ

Rumeysa KÖKOĞLU
(504211117)

Kontrol ve Otomasyon Mühendisliği Anabilim Dalı

Kontrol ve Otomasyon Mühendisliği Programı

Tez Danışmanı: Prof. Dr. Müjde GÜZELKAYA

HAZİRAN 2025

Rumeysa KÖKOĞLU, a M.Sc. student of ITU Graduate School student ID 504211117, successfully defended the thesis entitled "A COMPARATIVE STUDY ON CONVENTIONAL AND ENERGY-BASED AUTOPILOT SYSTEMS ON A JET-POWERED AIRCRAFT", which she prepared after fulfilling the requirements specified in the associated legislations, before the jury whose signatures are below.

Thesis Advisor : **Prof. Dr. Müjde GÜZELKAYA** ....
Istanbul Technical University

Jury Members : **Assoc. Prof. Dr. Ercan ATAM** ....
Boğaziçi University

Asst. Prof. Dr. Erhan YUMUK ....
Istanbul Technical University

Date of Submission : 30 May 2025
Date of Defense : 17 June 2025

to my loved ones,

FOREWORD

First and foremost, I would like to sincerely thank my thesis advisor, Prof. Dr. Müjde Güzelkaya, for her invaluable guidance, and encouragement. Her expertise and constructive feedback were instrumental in shaping this study, and I am truly grateful for her unwavering support.

I am also deeply thankful to TUSAŞ (Turkish Aerospace Industries), where I had the privilege of conducting my research. The company's resources, innovative environment, and the expertise of its team greatly contributed to the success of this work. My special thanks go to Chief Engineer Reşit Demirkıran for his mentorship, technical insights, and continuous support during my research. His guidance helped me bridge the gap between theory and practice, making this study more comprehensive.

I would like to extend my heartfelt gratitude to my family for their endless love, patience, and encouragement. Their belief in me has been my greatest motivation, and I owe them more than words can express.

Lastly, and most importantly, I would like to thank Ahmet, for his unconditional love and invaluable support in all areas of my life. His presence has been my greatest strength, and I will be forever grateful for his understanding and encouragement.

I am sincerely grateful to everyone who contributed to this work, directly or indirectly.

June 2025 Rumeysa KÖKOĞLU

TABLE OF CONTENTS

Page

ABBREVIATIONS

AFCF	: Automatic Flight Control Function
CAS	: Calibrated Air Speed
DCM	: Direction Cosine Matrix
DEHA	: Deneysel Hava Aracı
EAS	: Equivalent Air Speed
EPR	: Engine Pressure Ratio
FMS	: Flight Management System
GPS	: Global Positioing System
IAS	: Integrated Air Speed
IMU	: Inertial Measurement Unit
ISA	: International Standard Atmosphere
LNAV	: Lateral Navigation
LQR	: Linear Quadratic Regulator
MIMO	: Multi-Input Multi-Output
MPC	: Model Predictive Control
PI	: Proporitonal-Integral
PID	: Proportional-Integral-Derivative
SCAS	: Stability and Control Augmentation System
SISO	: Single-Input Single-Output
SLAM	: Simultaneous Localization and Mapping
SLC	: Successive Loop Closure
TAS	: True Air Speed
TECS	: Total Energy Control System
THCS	: Total Heading Control System
TSRV	: Transport System Research Vehicle
UAV	: Unmanned Aerial Vehicle
VNAV	: Vertical Navigation

SYMBOLS

a	: Speed of sound
g	: Acceleration due to gravity
h	: Altitude
L, M, N	: Moments
m	: Mass
p, q, r	: Angular rates
R	: Gas constant
T	: Temperature
u, v, w	: Linear velocities
X, Y, Z	: Forces
$\boldsymbol{\delta_{T_c}}$	: Throttle command
$\boldsymbol{E_T}$	: Total energy
$\boldsymbol{\dot{E}_T}$	: Total energy rate
$\boldsymbol{\dot{E}_s}$	: Specific energy rate
$\boldsymbol{\gamma}$	: Flight path angle
$\boldsymbol{\dot{h}}$	: Vertical speed
$\boldsymbol{\dot{L}_s}$	: Specific energy distribution rate
$\boldsymbol{\dot{\phi}, \dot{\theta}, \dot{\psi}}$	: Euler rates
$\boldsymbol{\rho}$	: Air density
$\boldsymbol{\tau}$	: Time constant
$\boldsymbol{\theta_c}$	: Pitch attitude command
$\boldsymbol{\dot{V}}$	: derivative of airspeed

LIST OF TABLES

Page

LIST OF FIGURES

Page

A COMPARATIVE STUDY ON CONVENTIONAL AND ENERGY-BASED AUTOPILOT SYSTEMS ON A JET-POWERED AIRCRAFT

SUMMARY

The development of flight control systems has been vital to the historical progress of aviation. Initially developed for directional stabilization only, autopilot systems have evolved into multi axis and more integrated structures due to increasing mission complexity and high speed conditions. The purpose of this study is to comprehensively analyze the performance of two basic architectures among the longitudinal axis flight control systems used in modern aircraft, namely the conventional autopilot structures and the energy based Total Energy Control System (TECS) architecture. Effective and balanced control of longitudinal states such as airspeed and altitude is critical, especially in highly maneuverable aircraft. Accordingly, both control approaches are evaluated in mathematical modelling, controller design and numerical simulation environments.

Conventional autopilot architecture, which forms the basis of the study, represents the structures that have been successfully implemented on different platforms for many years in the historical development of flight control systems. This architecture is basically based on the control of flight parameters through independent loops. Airspeed is managed by throttle and altitude by elevator commands. Usually, PI or PID type controller structures are used for each control variable and control loops are designed separately. This structure simplifies the controller design process and allows the system behavior to be analyzed separately. Moreover, the modularity of conventional architecture facilitates maintenance and improvement processes in engineering applications.

In this study, conventional structure is modeled through three basic control modules: pitch attitude, altitude and speed autopilot. While the pitch attitude autopilot stabilizes the pitch angle in the inner loop, the altitude autopilot is positioned outside this structure and first generates vertical speed commands and then generates a pitch angle reference. The speed autopilot generates throttle commands using a PI controller structure and follows a specific true airspeed reference. Each control function is subjected to open loop frequency analysis by loop breaking method and gain tuning is performed in MATLAB/Simulink environment according to the stability margins of the system.

Among the advantages of conventional control structures, reliable performance is particularly important in flight stages with low mission complexity. In missions such as maintaining constant altitude and airspeed, the system can successfully track reference values and limit the load on the control surfaces. Furthermore, the high level of industrial maturity in terms of system comprehensibility and engineering experience increases the preference of this architecture. However, the main limitation of conventional systems is that the physical relationship of the states in the longitudinal axis is not sufficiently considered in the control architecture. Especially in scenarios such as loss of airspeed during climb or climb during acceleration, independent control

loops generate commands that affect each other, which can lead to conflicting control actions. This can have negative effects on flight safety and comfort.

TECS architecture, which constitutes the other half of the study, offers a more integrated and energy based control approach developed to overcome the limitations mentioned above. TECS manages the airspeed and altitude states to be controlled during flight through kinetic and potential energy components. Throttle commands are generated to control the total energy rate ($\dot{E}$), while elevator commands are used to determine how this energy is distributed between the kinetic and potential energy. Thus, the control signals given by the system are not only associated with reference values, but also with physical energy conversion mechanisms. In contrast to conventional architecture, this structure is based on MIMO control philosophy and inherently takes couple effects into account.

TECS architecture was implemented through an eight steps design process in this thesis. Initially, throttle and elevator loops were modeled independently, and then the couple effects between these loops were gradually incorporated into the system. At each step, the controller gains were evaluated through open loop frequency analysis, and PI control loops were configured with MATLAB/Simulink. After the integrated system was obtained, two proportional control loops were designed to track the reference altitude and airspeed values during flight, and these loops generate the throttle and pitch angle commands.

The performance of TECS has produced remarkable results, especially in complex missions where airspeed and altitude are changed simultaneously. While throttle and elevator commands often conflict in conventional autopilots, TECS synchronizes these actions through energy rate and distribution, reducing the load on the control surfaces and achieving reference values in less time. In addition, the significant reduction in throttle deviations increases system reliability and optimizes fuel consumption in jet engine systems. However, the more complex structure of TECS increases design time and makes system implementation more sophisticated. Its high dependency on sensor data can adversely affect the performance of the system in the presence of environmental disturbances or hardware induces deviations.

Both architectures were modeled on the mathematical model of DEHA and tested with different mission scenarios. The mission scenarios included constant altitude and airspeed missions, climbing during airspeed changes, sudden bank angle changes and combinations with energy intensive transitions. Simulation results show that while both architectures perform well in simple missions, TECS outperforms in complex scenarios with significant energy transitions. The simplicity of conventional architecture makes it particularly suitable for standard missions, while TECS stands out in situations requiring higher accuracy, precision and flight efficiency.

This study has shown that the choice of architecture in the design of flight control systems should be based not only on technical criteria, but also on the flight profile, platform characteristics and mission requirements. While conventional architectures offer fast implementation and low complexity due to their industry established nature energy based architectures such as TECS promise higher levels of control accuracy and efficiency. Therefore, it is recommended that these two architectures should not be considered as alternatives, but rather as complements for different types of missions.

In the future, the development of hybrid control systems that combine the advantages of both structures can provide more flexible and adaptive solutions for modern flight missions. Furthermore, supporting TECS structure with adaptive algorithms can increase the robustness of the system against environmental uncertainties. Integration of conventional systems with advanced flight management systems can further improve control accuracy.

In conclusion, this thesis has provided a comparative perspective in the field of flight control engineering by providing a comprehensive evaluation of both conventional autopilot systems and TECS architecture. The application areas, limitations and potentials of both systems have been clearly demonstrated, and a scientific framework has been presented to guide decision makers in their choice of flight control architecture.

GELENEKSEL VE ENERJİ TABANLI OTOPİLOT SİSTEMLERİNİN JET MOTORLU BİR HAVA ARACINDA KARŞILAŞTIRMASI

ÖZET

Uçuş kontrol sistemlerinin gelişimi, havacılığın tarihsel ilerleyişinde hayati bir rol oynamıştır. İlk dönemlerde yalnızca yön stabilizasyonu amacıyla geliştirilen otopilot sistemleri, zamanla artan görev karmaşıklığı ve yüksek hız koşulları nedeniyle çok eksenli ve daha entegre yapılara dönüşmüştür. Bu çalışma, modern hava araçlarında kullanılan boylamsal eksenli uçuş kontrol sistemleri arasında yer alan iki temel mimarinin, yani geleneksel otopilot yapıları ile enerji temelli TECS (Total Energy Control System) mimarisinin performansını kapsamlı biçimde analiz etmek amacıyla gerçekleştirilmiştir. Hız ve irtifa gibi boylamsal büyüklüklerin etkili ve dengeli bir biçimde kontrol edilmesi, özellikle yüksek manevra kabiliyetine sahip hava araçlarında kritik öneme sahiptir. Bu doğrultuda, her iki kontrol yaklaşımı, matematiksel modelleme, kontrolcü tasarımı ve sayısal simülasyon ortamlarında değerlendirilmiş; elde edilen bulgularla birlikte sistemlerin güçlü ve zayıf yönleri ortaya konmuştur.

Çalışmanın temelini oluşturan geleneksel otopilot mimarisi, uçuş kontrol sistemlerinin tarihsel gelişimi içerisinde uzun yıllar boyunca farklı platformlarda başarıyla uygulanmış yapıları temsil etmektedir. Bu mimari temel olarak, uçuş parametrelerinin bağımsız döngüler üzerinden kontrol edilmesini esas alır. Hız, throttle kontrolü ile; irtifa ise elevator komutları aracılığıyla yönetilir. Her bir kontrol değişkeni için genellikle PI veya PID tipi kontrolcü yapıları kullanılmakta ve kontrol döngüleri ayrı ayrı tasarlanmaktadır. Bu yapı, kontrolcü tasarım sürecini sadeleştirirken, sistem davranışlarının ayrı ayrı analiz edilebilmesine olanak tanır. Ayrıca, geleneksel mimarinin sahip olduğu modülerlik özelliği, mühendislik uygulamalarında bakım ve iyileştirme süreçlerini kolaylaştırmaktadır.

Bu çalışmada, geleneksel yapı; yunuslama tutum, irtifa ve hız otopilotları olarak üç temel kontrol modülü üzerinden modellenmiştir. Yunuslama tutum otopilotu, iç döngüde yunuslama açısının stabilizasyonunu sağlarken, irtifa otopilotu bu yapının dışında konumlanarak önce dikey hız komutu üretmekte ve ardından yunuslama açısı referansı oluşturmaktadır. Hız otopilotu ise PI denetleyici yapısı kullanarak itki komutları üretir ve belirli bir gerçek hava hızı referansını takip eder. Her bir kontrol modülü, döngü kırma (loop breaking) yöntemiyle açık çevrim frekans analizlerine tabi tutulmuş, kazanç ayarlamaları ise sistemin kararlılık marjlarına göre MATLAB/Simulink ortamında gerçekleştirilmiştir.

Geleneksel kontrol yapılarının avantajları arasında özellikle düşük görev karmaşıklığına sahip uçuş fazlarında sağladığı güvenilir performans ön plana çıkmaktadır. Sabit irtifa ve hız korunumu gibi görevlerde sistem, hedef değerleri başarılı şekilde izleyebilmekte, kontrol yüzeylerine binen yükü sınırlı tutabilmektedir. Ayrıca, sistemin anlaşılabilirliği ve mühendislik tecrübesi açısından yüksek düzeyde endüstriyel olgunluğa ulaşmış olması, bu mimarinin tercih edilme oranını artırmaktadır. Bununla birlikte, geleneksel sistemlerin temel sınırlılığı, uçuşun

boylamsal ekseninde yer alan büyüklüklerin fiziksel ilişkisinin kontrol mimarisi içinde yeterince dikkate alınmamasıdır. Özellikle irtifa kazanımı sırasında hız kaybı veya hızlanma sırasında irtifa artışı gibi senaryolarda, bağımsız kontrol döngülerinin birbirini etkileyen sinyaller üretmesi, kontrol eylemlerinin çatışmasına neden olabilmektedir. Bu durum, uçuş güvenliği ve konfor açısından olumsuz etkiler yaratabilmektedir.

Çalışmanın diğer yarısını oluşturan TECS mimarisi, yukarıda belirtilen sınırlılıkları ortadan kaldırmaya yönelik olarak geliştirilmiş, daha entegre ve enerji temelli bir kontrol yaklaşımı sunmaktadır. TECS, uçuş sırasında kontrol edilmek istenen hız ve irtifa büyüklüklerini, kinetik ve potansiyel enerji bileşenleri üzerinden yönetmektedir. İtki komutları, toplam enerji oranını ($\dot{E}$) kontrol etmek amacıyla üretilirken; irtifa dümeni komutları, bu enerjinin kinetik ve potansiyel formlar arasında nasıl dağıtılacağını belirlemek amacıyla kullanılmaktadır. Böylece, sistemin verdiği kontrol sinyalleri yalnızca hedef değerlerle değil, aynı zamanda fiziksel enerji dönüşüm mekanizmaları ile de ilişkilendirilmiş olur. Bu yapı, geleneksel mimarinin tersine, MIMO (Multiple Input Multiple Output) kontrol felsefesine dayanır ve çapraz etkileşimleri doğal olarak hesaba katar.

TECS mimarisi, tez kapsamında sekiz aşamalı bir tasarım süreciyle gerçekleştirilmiştir. Başlangıçta itki ve irtifa dümeni kontrol döngüleri birbirinden bağımsız olarak modellenmiş, ardından bu döngüler arası etkileşimler aşamalı şekilde sisteme dahil edilmiştir. Kontrolcü kazançları, her adımda açık çevrim frekans analizleri (Bode çizimleri) ve zaman yanıtları üzerinden değerlendirilmiş; MATLAB Control System Designer aracı ile PI kontrol döngüleri yapılandırılmıştır. Entegre sistem elde edildikten sonra, uçuş sırasında hedeflenen irtifa ve hız değerlerinin takibi için ayrıca iki oransal kontrol döngüsü tasarlanmış, bu döngüler itki ve yunuslama açısı komutlarını üretmiştir.

TECS'in performansı, özellikle hız ve irtifanın aynı anda değiştirildiği karmaşık görevlerde dikkat çekici sonuçlar üretmiştir. Geleneksel otopilotlarda itki ve irtifa dümeni komutları sıklıkla çatışırken, TECS bu eylemleri enerji oranı ve dağılımı üzerinden senkronize etmekte, kontrol yüzeylerine binen yükü azaltmakta ve hedef değerleri daha kısa sürede yakalamaktadır. Ayrıca itki salınımlarının belirgin biçimde azalması, jet motorlu sistemlerde sistem güvenilirliğini artırmakta, yakıt tüketimini optimize etmektedir. Bununla birlikte, TECS mimarisinin daha karmaşık yapısı, tasarım süresini uzatmakta ve sistem uygulamasını daha sofistike hale getirmektedir.

Her iki mimari, DEHA platformu üzerinde modellenmiş, farklı görev senaryoları ile test edilmiştir. Görev senaryoları arasında sabit irtifa-hız görevleri, tırmanış-esnasında hız değişimi, ani yatış açısıyla yön değiştirme ve enerji yoğun geçişler içeren kombinasyonlar yer almıştır. Simülasyon sonuçları göstermektedir ki; basit görevlerde her iki mimari de başarılı sonuçlar üretirken, enerji geçişlerinin belirgin olduğu karmaşık senaryolarda TECS daha üstün performans sergilemektedir. Geleneksel mimari, sadeliği sayesinde özellikle standart görevlerde tercih edilebilecek uygun bir yapı sunarken, TECS daha yüksek doğruluk, hassasiyet ve uçuş verimliliği gerektiren durumlarda öne çıkmaktadır.

Bu çalışma, uçuş kontrol sistemlerinin tasarımında mimari seçimin yalnızca teknik kriterlere değil, aynı zamanda uçuş profiline, platformun özelliklerine ve görev gereksinimlerine göre yapılması gerektiğini ortaya koymuştur. Geleneksel mimariler, endüstride oturmuş yapıları sayesinde hızlı entegrasyon ve düşük karmaşıklık sunarken; TECS gibi enerji temelli yapılar daha ileri seviye kontrol doğruluğu ve verimlilik vaat etmektedir. Dolayısıyla, bu iki yapının birbirinin alternatifi değil, farklı görev türleri için tamamlayıcısı olarak değerlendirilmesi önerilmektedir.

Gelecekte, her iki yapının avantajlarını bir araya getiren hibrit kontrol sistemlerinin geliştirilmesi, modern uçuş görevleri için daha esnek ve uyarlanabilir çözümler sağlayabilir. Ayrıca, TECS yapısının adaptif algoritmalarla desteklenmesi, çevresel belirsizliklere karşı sistemin dayanıklılığını artırabilir. Geleneksel sistemlerin ise gelişmiş uçuş yönetim sistemleri ile entegrasyonu, kontrol doğruluğunu daha da iyileştirebilir.

Sonuç olarak bu tez hem geleneksel otopilot sistemlerinin hem de TECS mimarisinin kapsamlı bir şekilde değerlendirilmesini sağlayarak, uçuş kontrol mühendisliği alanında karşılaştırmalı bir bakış açısı sunmuştur. Her iki sistemin uygulama alanları, sınırları ve potansiyelleri açıkça ortaya konmuş; karar vericilere uçuş kontrol mimarisi seçiminde yol gösterici olacak bilimsel bir çerçeve sunulmuştur.

1. INTRODUCTION

Throughout human history, the idea of flight has intrigued many civilizations. Leonardo da Vinci’s drawings of flying machines in the 15th century are one of the earliest examples of this dream. However, in practice, powered flight was realized on December 17, 1903, with the Wright Brothers’ “Flyer I” (Anderson, 2002). This flight was a turning point not only in terms of taking off but also in terms of the effectiveness of guidance and control mechanisms. From the beginning of the 20th century, aviation technology developed rapidly and gained great momentum during World Wars I and II, during which many engineering studies were conducted on the stability and controllability of airplanes (Bilstein, 2001). With the advent of jet engines, high speed flights became possible, and aircraft control became more complex. This necessitated the development of automatic control systems.

The first flight attempts were aimed at understanding basic aerodynamic principles and mechanical balance. In this process, the development of control mechanisms has gained vital importance, and autopilot structures, defined as automatic control systems, have emerged over time. The effective use of control surfaces has made it possible to increase flight safety, reduce pilot load and automate flight tasks.

From the first attempts, aircraft control methods have undergone a major evolution. One of the most striking developments in this evolution is the emergence and development of autopilot systems integrated into aircraft. Aircraft autopilot systems began to be developed in the early 20th century to improve flight safety, reduce pilot workload, and provide stability in long-range flights. One of the first works in this field was the gyroscope-based autopilot system developed by Sperry Corporation in 1912. Although this system only provides orientation stabilization, it laid the foundation for modern flight control systems (Stevens et al., 2016).

The development of autopilot systems in aircraft has evolved over many years to meet the ever-increasing requirements of aviation in terms of safety, efficiency, and operational complexity. While the first generation of autopilot systems only provided automatic stabilization in limited axes, today's systems have become more complex with multiple flight modes and energy management algorithms.

The first examples of automatic control systems date back to the 1910s. Early autopilot applications consisted of simple PID-based systems to maintain a constant altitude and heading during flight. Over time, as aircraft aerodynamics became more complex and mission scenarios expanded, autopilot systems became multi-channel, multi-target (speed-hold, altitude-hold, path-tracking) (Stevens & Lewis, 2003).

Flight control systems are one of the most critical components of modern aircraft and have evolved over time in parallel with increasing mission complexity on both manned and unmanned platforms. The historical development of these systems has evolved from primarily manual control systems to today's sophisticated automatic flight control systems (AFCS).

The concept of autopilot was first developed in 1912 with a gyroscopic based system developed by Lawrence Sperry. This system reduced the pilot's workload by keeping the airplane flying on a fixed route (Roland, 1985). Initially providing directional protection only in the horizontal axis, these systems evolved over time to control different aspects of the flight. During World War II, autopilot systems became more widespread and were used especially for targeted bombing and long range flights. Autopilot systems operating with analog circuits in the 1950s were replaced by more precise and programmable systems with the development of digital electronics in the 1970s. Since the 1980s, with the increase in flight control computers, autopilot systems have been developed to provide multi axis control (Dorf & Bishop, 2011).

Autopilot systems are generally divided into three basic categories: speed autopilot, altitude autopilot, and attitude autopilot. Speed autopilot maintains a given airspeed or ground speed by controlling the aircraft's propulsion system. These systems are used to improve fuel efficiency, especially in commercial aircraft. Altitude autopilot keeps the aircraft flying at a given altitude by controlling its movements on the vertical axis. This control is usually provided by barometric altimeter data or radar altimetry. Attitude autopilot maintains the aircraft's heading and pitch by controlling pitch, roll, and yaw movements. This type of autopilot works in conjunction with the IMU (Inertial Measurement Unit) and GPS to provide stability, especially in small UAVs (McLean, 1990).

Navigation based autopilot systems include algorithms that control both the longitudinal and lateral states of the route. Longitudinal navigation includes airspeed, altitude, and pitch control, while lateral navigation corrects left/right deviations of the aircraft and includes heading and yaw control. In these systems, PID (Proportional-Integral-Derivative) based controllers or more advanced LQR (Linear Quadratic Regulator) and MPC (Model Predictive Control) algorithms are usually used. Especially in aircraft from major manufacturers such as Boeing and Airbus, these navigation commands are transmitted to the autopilot system via the flight management system (FMS), and the autopilot controls the entire flight plan (Skogestad & Postlethwaite, 2007).

Autopilot systems today are capable of performing a wide range of tasks, not only directional and altitude control but also fuel management, navigation, collision avoidance, and emergency maneuvers. In large passenger aircraft (e.g. Boeing 787 or Airbus A350), more than 90% of the flight is operated by autopilot (FAA, 2020). Autopilot systems developed for unmanned aerial vehicles and drones generally use open source software (e.g., ArduPilot, PX4) and are integrated with low cost sensors. In these systems, sensor data such as IMU, GPS, and barometer are used to ensure flight stability (Mahony et al., 2012). In advanced UAVs, autopilots equipped with systems such as SLAM (Simultaneous Localization and Mapping), stereo cameras, and lidar have become capable of making more autonomous decisions (Bry et al., 2015).

Conventional longitudinal autopilot systems are based on SISO (Single Input, Single Output) control architecture. In this architecture, the altitude and speed control loops are designed independently. Typically, the elevator provides altitude control with a PI or PID control structure, while the throttle is used for speed control with a similar control structure (McLean, 1990; Stevens et al., 2016). These systems have been widely used on different platforms, such as commercial passenger aircraft, business jets, and unmanned aerial vehicles (UAVs). For example, in Airbus and Boeing airplanes, altitude and airspeed control in autopilot modes are implemented separately based on PID (Cook, 2007).

Conventional autopilot systems on longitudinal axis usually manage flight parameters such as airspeed, and altitude through independent control loops. In these systems, the throttle usually controls airspeed and the elevator controls altitude. However, this approach does not adequately reflect the physical dynamics of aircraft because airspeed and altitude are directly linked. When altitude is increased, airspeed decreases, and when airspeed is increased, altitude decreases. This cross interaction can cause conventional control loops to conflict with each other and cause the system to react erratically (Chudy & Rzucidlo, 2012). This limitation degrades system performance, especially when sudden maneuvers, turbulence, and rapid energy changes are required.

While early autopilot systems were simple structures that ensured stable flight only in certain axes, today's control architectures have become highly integrated structures that manage different flight modes, perform mission-oriented optimizations, and make energy-based decisions. TECS, developed in this context, is a flight control architecture that aims to perform aircraft control in the longitudinal axis directly through physical energy concepts instead of conventional speed and altitude separation methods (Lambregts, 1983). It was first implemented by Antonius A. Lambregts at NASA on the Boeing 737 flight test platform in 1981, and detailed results were published in 1983 (Lambregts, 1983). TECS is based on the MIMO (Multiple Input, Multiple Output) approach instead of the traditional SISO (Single Input, Single Output) and can directly control flight energy (potential and kinetic energy) through thrust and elevators.

The basic assumption of TECS is to indirectly manage airspeed and altitude by controlling the total energy rate and energy distribution of the aircraft. TECS control architecture generally has two basic loops: a throttle loop that controls the total energy rate and an elevator loop that controls the energy distribution rate. These loops are operated with proportional-integral (PI) controllers. For example, the throttle command provides the acceleration required to achieve a target energy rate, while the elevator command generates flight path angle or pitch commands to achieve the target energy distribution (Giusti et al., 2020).

The mathematical background is based on the derivative of energy with respect to time and Lagrangian definitions. TECS control algorithm usually has two PI control loops: one for the rate of the total energy (throttle) and one for the distribution rate of the total energy (elevator). This control structure improves control performance by minimizing cross-effects between altitude and airspeed control (Argyle & Beard, 2016; Degaspare & Kienitz, 2020).

The first platform on which TECS was implemented, NASA’s Transport System Research Vehicle (TSRV), was a modified Boeing 737 aircraft. Flight tests on this platform showed that compared to conventional autopilots, TECS reached target altitudes and speeds in less time, showed less throttle oscillation, and significantly improved fuel efficiency (Lambregts, 1983).

TECS architecture was later applied to small aircraft, especially unmanned aerial vehicles (UAVs). Chudy and Rzucidlo (2012) implemented a TECS-THCS based system on a general aviation class aircraft and found that speed and altitude regulation was more successful than conventional PID structures. According to the test results, throttle oscillations were reduced by 35%, the time to reach altitude was shortened, and the system became more stable. TECS has been implemented on platforms ranging from UAV systems to commercial aircraft. TECS modules in open source software platforms such as ArduPilot and PX4 provide energy efficiency in various missions. Weiser et al. (2024) tested the performance of TECS in high-altitude aircraft, while Spark et al. (2021) compared TECS with conventional SISO systems.

In recent years, many innovations have been added to the classic TECS structure. Extended TECS have been developed with nonlinear control techniques. Chakraborty et al. (2021) designed a Lyapunov based PI structure, the system shows less deviation, and the throttle and elevator commands are smoother and more balanced. TECS architectures have been more integrated with the engine control system and optimized together with engine dynamics, especially in systems operating with gas turbine engines. Guisti and Kienitz (2020) proposed a more integrated TECS architecture in which variables such as engine pressure ratio (EPR) are directly used as control parameters. In this way, not only flight control but also a structure that prevents exceeding engine limits is obtained. In another study, the IAS (indicated airspeed) priority control approach is discussed. While the elevator maintains IAS, the throttle operates according to the total energy, and the stall risk is minimized (Bauer, 2023).

2. DEHA MODEL AND SIMULATION ENVIRONMENT

Within the scope of this thesis, DEHA (Deneysel Hava Aracı) jet aircraft was used as a test and simulation model. DEHA is an aircraft within TUSAŞ (Türk Havacılık ve Uzay Sanayii) to support academic studies on high technology topics such as aircraft simulation models, flight control algorithms, etc. The mass and geometric data of the unmanned aerial vehicle DEHA shown in Figure 2.1 are given in Table 2.1 (Altay, H., 2021).

Table 2.1 : Parameters of DEHA.

Parameter	Symbol	Value	Unit
Total mass	m_{total}	14.48	kg
Mass without fuel	m_{empty}	11.64	kg
Wingspan	b	2.2	m
Length	l	2.1	m
Wing surface	S	91.6	dm^2
Position of cg on x-axis	X_{cg}	1.087	m
Position of cg on y-axis	Y_{cg}	0	m
Position of cg on z-axis	Z_{cg}	-0.007	m
Moment of Inertia	I_{xx}	0.121	$kg\ m^2$
Moment of Inertia	I_{yy}	0.353	$kg\ m^2$
Momet of Inertia	I_{zz}	0.501	$kg\ m^2$
Product of Inertia	I_{xz}	0	$kg\ m^2$

The nonlinear model of DEHA is modeled in MATLAB/Simulink environment and is a highly accurate model with more than 200 sorties to date. Among these sorties, there are system identification flight tests required to verify the aerodynamic model of DEHA, performance validation of the developed various control algorithms, and autonomous flight tests.

Figure 2.1 : DEHA small jet-powered aircraft

It can be seen in Figure 2.2 that DEHA UAV model consists of six blocks that represent the actuator model, aerodynamic model, engine model, environment model, mass & inertia, and equation of motion model.

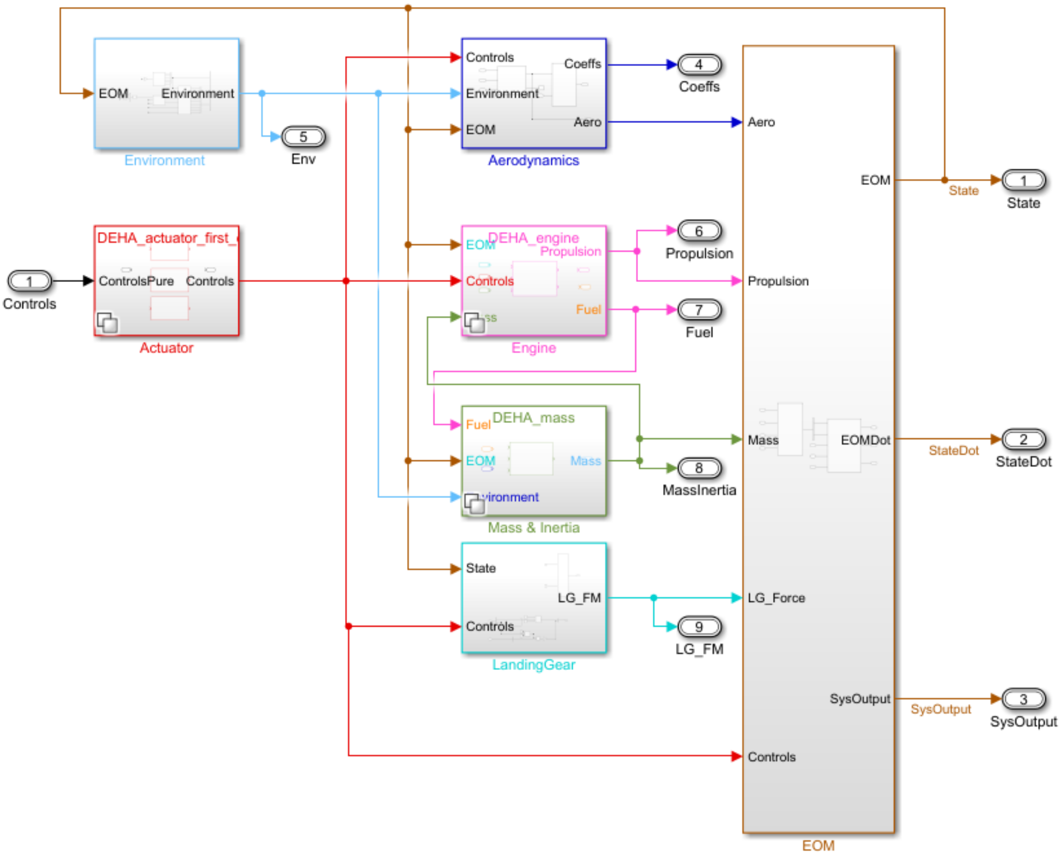

Figure 2.2 : Open-Loop Model of DEHA.

The aerodynamic model integrates data from various components, such as the environmental model, control surface actuators, and motion equations. It processes all of this input to compute the aerodynamic forces and moments acting on the vehicle.

These computed forces and moments are then passed back into the equations of motion to determine the system's overall dynamics and behavior. To evaluate the static aerodynamic coefficients defined in the wind axis frame, lookup tables rely on key variables. On the other hand, the dynamic aerodynamic coefficients are derived using a linear approximation that depends on the angular velocity components (p, q, r).

The DEHA UAV’s propulsion system was modeled using a Power Curve approach. Here, the throttle input is interpreted as a percentage of the available engine thrust. The actual thrust output at a given throttle level is obtained by scaling the available thrust—based on the aircraft's current altitude and velocity—by this percentage.

The atmospheric model uses the vehicle’s altitude to compute outputs such as Mach number, dynamic pressure, temperature, and air pressure. Within the broader Environment model, there are three core modules: the ISA model, the wind model, and the airspeed model. The ISA (International Standard Atmosphere) module specifically handles calculations for parameters like temperature (T), speed of sound (a), pressure (P), and air density (ρ), all based on standard atmospheric assumptions. These values are computed using a set of physical equations involving the lapse rate (L), the adiabatic index (γ), and the universal gas constant (R).

$$T = T_0 - L \tag{2.1}$$

$$a = \sqrt{\gamma R(T_0 - L)} \tag{2.2}$$

$$P = P_0 e^{\frac{g(h_{trop}-h)}{RT}} \tag{2.3}$$

$$\rho = \rho_0 e^{\frac{g(h_{trop}-h)}{RT}} \tag{2.4}$$

Wind disturbances such as gusts and turbulence are modeled within the Wind module using the Dryden turbulence model along with Direction Cosine Matrix (DCM) transformations. The outputs from this module include the wind velocity components (u, v, w) and the angular rates (p, q, r). These outputs, together with data from the ISA (International Standard Atmosphere) module, are used as inputs to the Airspeed module. Within this system, key flight parameters are computed—including True Airspeed (TAS), Calibrated Airspeed (CAS), Equivalent Airspeed (EAS), wind rate components, gust rates, Mach number, and dynamic pressure.

The equations of motion module serves as a computational core that combines inputs from the aerodynamic, propulsion, and environmental subsystems to determine the aircraft's position, velocity, angular motion, and aerodynamic angles. This model consists of six interrelated components: external forces, resultant forces, aerodynamic moments, Euler angles, navigation data, and the transformation from body axes to wind axes. Together, these elements work in tandem to simulate the complete motion and dynamic response of the aircraft. The governing equations of motion are given as follows.

Force equations:

$$X - mgS_{\theta} = m(\dot{u} + qw - rv) \tag{2.5}$$

$$Y + mgC_{\theta}S_{\phi} = m(\dot{v} + ru - pw) \tag{2.6}$$

$$Z + mgC_{\theta}C_{\phi} = m(\dot{w} + pv - qu) \tag{2.7}$$

Moment equations:

$$L = I_x\dot{p} - I_{xz}\dot{r} + qr\left(I_z - I_y\right) - I_{xz}pq \tag{2.8}$$

$$M = I_y\dot{q} + rp(I_x - I_z) + I_{xz}(p^2 - r^2) \tag{2.9}$$

$$N = -I_{xz}\dot{p} + I_z\dot{r} + pq\left(I_y - I_x\right) + I_{xz}qr \tag{2.10}$$

Body Angular Velocities:

$$p = \dot{\phi} - \dot{\psi}\sin(\theta) \tag{2.11}$$

$$q = \dot{\theta}\cos(\phi) + \dot{\phi}\cos(\theta)\sin(\phi) \tag{2.12}$$

$$r = \dot{\psi}\cos(\theta)\cos(\phi) - \dot{\theta}\sin(\phi) \tag{2.13}$$

Euler Rates:

$$\dot{\theta} = qcos(\phi) - rsin(\phi) \tag{2.14}$$

$$\dot{\phi} = p + qsin(\phi)\tan(\theta) + rcos(\phi)\tan(\theta) \tag{2.15}$$

$$\dot{\psi} = (qsin(\phi) + rcos(\phi))\sec(\theta) \tag{2.16}$$

Aircraft Velocity:

$$\begin{bmatrix} \frac{dx}{dt} \\ \frac{dy}{dt} \\ \frac{dz}{dt} \end{bmatrix} = [X]\begin{bmatrix} u \\ v \\ w \end{bmatrix} \tag{2.17}$$

$$X = \begin{bmatrix} \cos(\theta)\cos(\psi) & \sin(\phi)\sin(\theta)\cos(\psi) - \cos(\phi)sin(\psi) & \cos(\phi)\sin(\theta)\cos(\psi) + \sin(\phi)\sin(\psi) \\ \cos(\theta)\sin(\psi) & \sin(\phi)\sin(\theta)\sin(\psi) + \cos(\phi)\cos(\psi) & \cos(\phi)\sin(\theta)\sin(\psi) - \sin(\phi)\cos(\psi) \\ -\sin(\theta) & \sin(\phi)\cos(\theta) & \cos(\phi)\cos(\theta) \end{bmatrix} \tag{2.18}$$

3. CONTROLLER DESIGN

3.1 Conventional Autopilots

The main responsibility of automatic flight control functions (AFCF) is to reduce pilot workloads. When AFCF is activated, the pilot will be out of the loop, and the autopilots generate all the reference inputs. DEHA has the following automatic flight control functions: Attitude Hold and Select (Pitch & Roll Attitude Hold and Select), Altitude Hold and Select, Speed Hold and Select, Route Hold (Lateral Navigation (LNAV) & Vertical Navigation (VNAV)).

Autopilots may have one or both hold and select modes. In hold mode, the relevant autopilot maintains the existing values of the aircraft at the time of engagement. In select mode, the relevant autopilot captures and maintains the reference inputs. These functions are closed over the SCAS (Stability and Control Augmentation Systems), and when they are activated, the autopilots will generate reference inputs of SCAS. This thesis focuses only on the longitudinal autopilots, which are pitch attitude autopilot, altitude autopilot, and speed autopilot. The main scheme of the longitudinal autopilots is shown in Figure 3.1. Basically, in the outer loop longitudinal autopilots consist of altitude and speed autopilots. Altitude autopilot produces pitch angle command for the pitch attitude autopilot. Then, it produces a pitch rate command for the inner loop rate controller. Speed autopilot produces throttle command input.

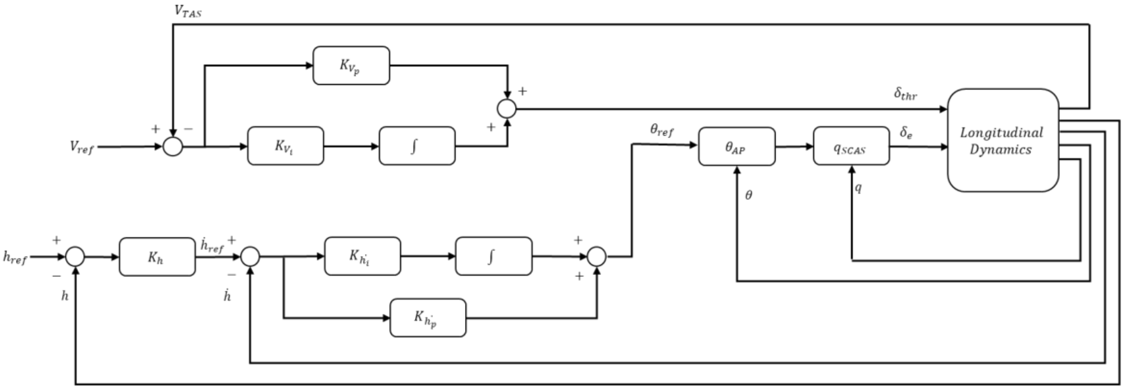

Figure 3.1 : Main Scheme of the Longitudinal Autopilots.

To design the autopilot system, a Successive Loop Closure (SLC) method is utilized since this approach simplifies the design of autopilots and allows to utilize the technique commonly used for autopilot design (McLean, D., 1990; Cook, M. V., 2013; MIL-F-9490D). SLC is a hierarchical control design strategy where inner control loops are closed and stabilized before designing the outer loops. This approach ensures robustness and stability by progressively building up the control structure layer by layer. The first step starts with the innermost loop, which has faster dynamics such as rate and angular velocity. Then, this loop is designed and stabilized independently, assuming that the outer loop inputs are constant or slowly changing. Once stable, the closed inner loop is treated as a new plant, and the next outer loop (e.g., attitude) is designed. The process is repeated for each additional loop, moving outwards in the control hierarchy.

An open loop system is shown in Figure 3.2 to illustrate this method. The open loop dynamics of the system are given as $P(s) = P_1(s)P_2(s)P_3(s)$. Each transfer function has an output as y_1, y_2, and y_3. In this case, it is interested in controlling the output y_3. Then, closed feedback loops are designed around y_1, y_2, and y_3 in succession instead of closing a single feedback loop with y_3, as shown in Figure 3.3. The main idea is to design the compensators $C_1(s), C_2(s)$, and $C_3(s)$ in succession.

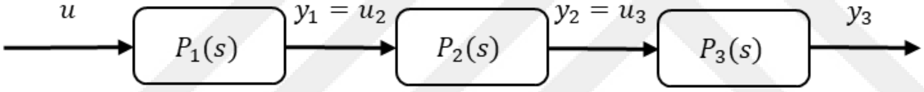

Figure 3.2 : Cascade of the three open-loop transfer functions.

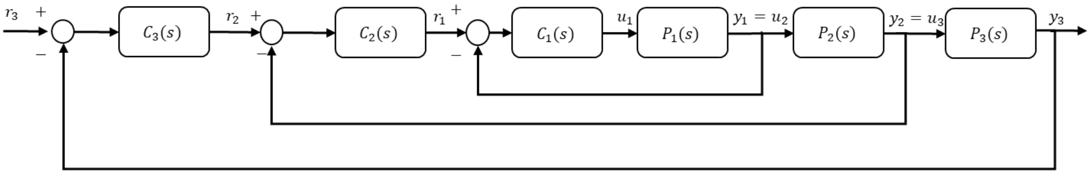

Figure 3.3 : Three-stage successive loop closure design.

Looking at the inner loop illustrated in Figure 3.3, the objective is to create a closed-loop system from r_1 to y_1 with a designated crossover frequency ω_{cg1}. It's assumed that for frequencies significantly lower than ω_{cg1}, the closed-loop transfer function $y_1(s)/r_1(s)$ behaves approximately like a unity gain system. This concept is visually represented in Figure 3.4. By treating the inner loop as having a gain of 1, the design of the subsequent (outer) loop becomes more straightforward, since it now only involves the plant $P_2(s)$ and the compensator $C_2(s)$. A key consideration in this sequential loop-closing approach is ensuring that each successive loop has a crossover frequency lower than that of the previous one. This same design approach is then applied iteratively to all subsequent loops.

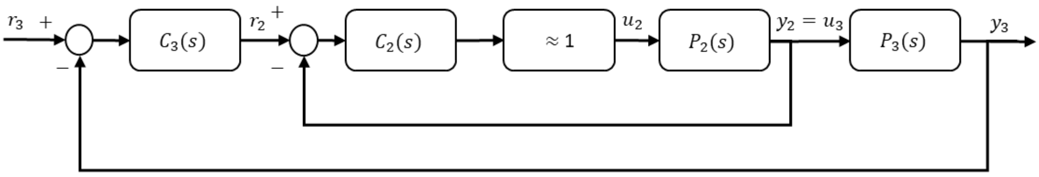

Figure 3.4 : SLC design with inner loop medeled as a unity gain.

The design process of the autopilot control law algorithm begins with an examination of the inner loop characteristics and their dynamic behavior. This is followed by an evaluation of key stability metrics, such as gain margin and phase margin, which provide insight into the system's robustness. Once the stability criteria are understood, the parameters of the autopilot control law can be systematically designated to meet performance objectives. Throughout this process, it is essential to consider the overall requirements of the autopilot system to ensure that the final design aligns with both operational needs and performance constraints. In this context, the design criteria for autopilots are typically specified as a gain margin of at least 6 dB and a phase margin of no less than 45 degrees.

In order to obtain controller gains that meet the specified stability requirements, loop breaking method is used. Loop breaking means artificially opening the feedback loop in a control system. In other words, the system normally operates in closed loop, but when the loop is broken, this closed loop flow is interrupted, and an open loop response is obtained at the interrupted point. The main purpose of loop breaking is to open the closed loop and examine the open loop behavior of the system at that point. With the help of loop breaking, the stability of the system can be analyzed using Nyquist and Bode plots, stability margins such as gain and phase margins can be measured, and the robustness analysis can be performed to measure sensitivity to uncertainties (Demirkiran, R. & Isci, H., 2023; Demirkiran R. & Isci, H. & Turkoglu, E., 2024). The loop is usually broken at the controller output or plant input because the purpose is to examine the controller and the system in series. The transfer function of the closed loop system is as follows.

$$T(s) = \frac{C(s)P(s)}{1+C(s)P(s)} \tag{3.1}$$

After the loop breaking, the analyzed system becomes the following.

$$L(s) = C(s)P(s) \tag{3.2}$$

When it is desired to design a controller using the loop breaking method, the closed loop system is opened; in other words, it is broken from the error signal. Then, a controller is designed by considering the stability analysis with the frequency responses of the obtained open loop system with the help of MATLAB Control System Designer. In this thesis, the loop breaking method is used in the design phases of both conventional and TECS based altitude and speed autopilots.

In consideration of the SLC and loop-breaking methods, supplementary design constraints have been integrated into the autopilot requirements. According to these constraints, the crossover frequency of the outer loop should lie between one-fifth and one-half of the inner loop’s crossover frequency. Moreover, the bandwidth of the outer loop is expected to be approximately one-third of that of the inner loop.

$$\frac{\omega_{cp,inner\ loop}}{5} \leq \omega_{cp,outer\ loop} \leq \frac{\omega_{cp,inner\ loop}}{2} \tag{3.3}$$

$$\omega_{BW,outer\ loop} \approx \frac{\omega_{BW,inner\ loop}}{3} \tag{3.4}$$

3.1.1 Pitch attitude autopilots

The pitch attitude autopilot is the inner loop of the longitudinal autopilots and simply controls the pitch angle of the aircraft. This autopilot is designed to ensure that the commanded pitch angle is tracked. When this autopilot is engaged, reference input of the longitudinal SCAS is generated by the pitch attitude autopilot, and the aircraft captures and maintains the reference pitch angle.

The pitch attitude autopilot operates as an outer loop of the longitudinal stability and command augmentation system and generates only pitch rate commands. The basic structure of pitch attitude autopilot is given in Figure 3.5.

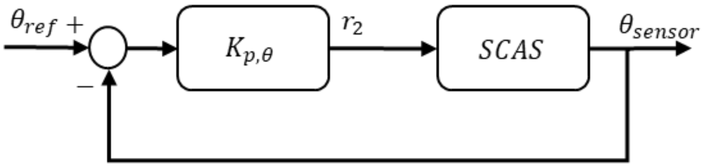

Figure 3.5 : Pitch attitude autopilot scheme.

In Figure 3.5, θ_{ref} refers to the reference pitch angle, and θ_{sensor} refers to the current Euler pitch angle. q_{ref} is the reference pitch rate, which is produced by the pitch attitude autopilot and sent to the longitudinal stability and command augmentation system.

In the design of pitch attitude autopilot, proportional (P) control law is employed with fixed proportional gain since the pitch rate responses in the autopilot design envelope have the same behavior and characteristics. The formulation of the pitch attitude autopilot is given in Equation 3.3.

$$q_{ref} = K_{p,\theta}(\theta_{ref} - \theta_{sensor}) \tag{3.3}$$

Since a proportional control law is used for the design of pitch attitude autopilot, the design of the proportional gain is the objective of the successive loop closure procedure, and it is required to obtain the proportional gain. Under ideal conditions, the proportional gain shows adequate tracking performance as long as the inner loop has an integrator structure.

With the loop breaking method, the closed loop is broken from the θ_{error} signal. Then, with the help of the control system design tool in MATLAB, the controller gains are found by considering the frequency and time responses. The open loop and closed loop frequency responses are shown in Figure 3.6 for 45m/s trim condition. $K_{P,\theta}$ gain is found as follows.

$$K_{P,\theta} = 1.32 \tag{3.4}$$

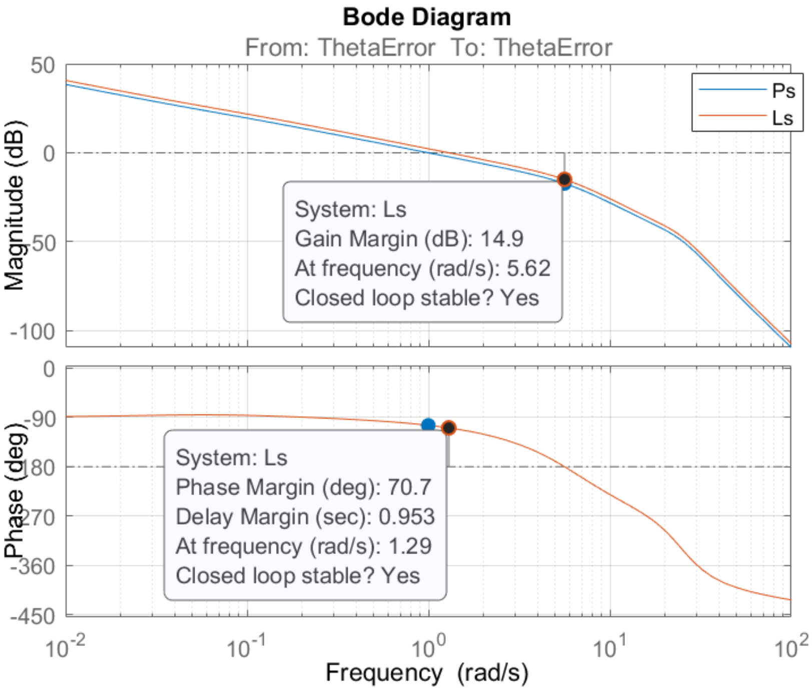

Figure 3.6 : Frequency Response of Pitch Attitude Loop.

As can be seen in Figure 3.6, the closed loop system has a gain margin of 14.9dB, a phase margin of 70.7° and meets the specified design requirements.

The same design process is repeated for the entire envelope in which the autopilots will operate, and the following Table 3.1 is produced.

Table 3.1 : Pitch Attitude Autopilot Proportional Gain Table.

Air Speed (m/s)	$K_{P,\theta}$
30	1.21
35	1.25
40	1.27
45	1.30
50	1.32
55	1.40
60	1.41

Figure 3.7 shows the linear model responses of the pitch attitude autopilot at intervals of 5m/s from 30 to 60m/s.

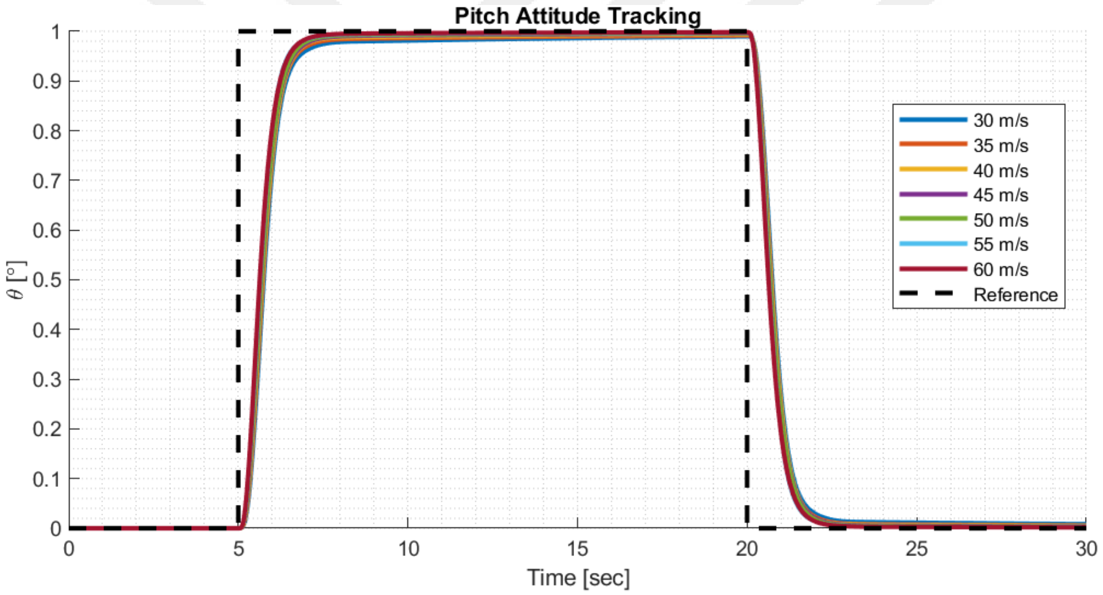

Figure 3.7 : Linear Model Singlet Response – Pitch Attitude Loop.

3.1.2 Altitude autopilot

The altitude autopilot is designed to decrease pilot workload by tracking the commanded altitude. When the altitude autopilot is engaged, the autopilot captures and maintains the selected altitude.

Altitude autopilot operates as an outer loop of the pitch attitude autopilot and generates the reference pith angle for pitch attitude autopilot. Altitude autopilot is designed to have two sub-functions: altitude and vertical speed autopilots. Altitude autopilot is designed as an outer loop around the vertical speed loop. Altitude autopilot generates vertical speed demand. Then, a vertical speed autopilot is used to ensure the desired vertical speed produced by the altitude autopilot and generates the pitch angle commands. The basic structure of altitude autopilot is given in Figure 3.8.

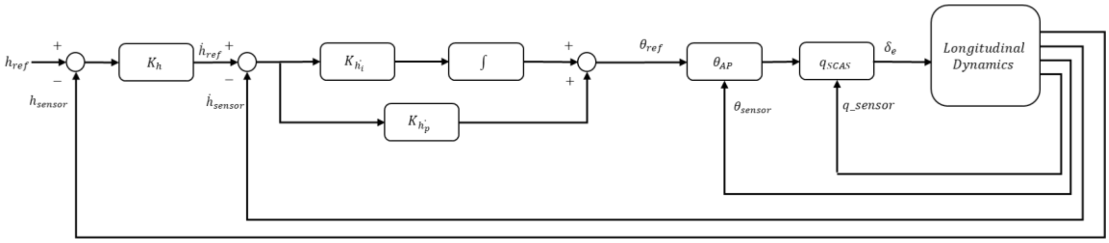

Figure 3.8 : Controller Structure of Altitude Autopilot.

In Figure 3.8, h_{ref} refers to the reference altitude, $\dot{h}_{ref}$ refers to the reference vertical speed, which is produced by the altitude autopilot, and θ_{ref} refers to the reference pitch angle which is produced by the vertical speed autopilot in altitude autopilot. h_{sensor}, $\dot{h}_{sensor}$, and θ_{sensor} refer to the current altitude, vertical speed, and pitch angle of the aircraft, respectively. q_{ref} is the reference pitch rate which is produced by the pitch attitude autopilot to send the stability and command augmentation system.

In the design of altitude autopilot, proportional (P) control law is used in altitude autopilot and proportional integral (PI) control law is used in vertical speed autopilot. For altitude autopilot a fixed proportional gain is used while vertical speed gains are scheduled with airspeed. The control law for altitude autopilot is given in Eq. 3.5.

$$\dot{h}_{ref} = K_{p,h}(h_{ref} - h_{sensor}) \tag{3.5}$$

The control law for the vertical speed autopilot is given in Eq. 3.6.

$$\theta_{ref} = K_{I,\dot{h}} \int (\dot{h}_{ref} - \dot{h}_{sensor})\, dt + K_{P,\dot{h}}(\dot{h}_{ref} - \dot{h}_{sensor}) \tag{3.6}$$

To find the gains of altitude autopilot, successive loop closure approach and design process are conducted. The procedure is applied in two steps for altitude autopilot. Firstly, controller gains of the vertical speed loop are obtained. After obtaining controller gains of the vertical speed loop, a proportional gain of the altitude loop is determined. The frequency response design method is used to tune the controller gains, as in the pitch attitude autopilot design processes.

With the loop breaking method, the closed loop is broken from the $\dot{h}$ error signal. Then, with the help of the control system design tool in MATLAB, the controller gains are found by considering the frequency and time responses. The open loop and closed loop frequency responses are shown in Figure 3.9 for 45m/s trim condition. $\mathrm{K}_{\mathrm{P},\dot{h}}$, and $K_{I,\dot{h}}$ gains are found as follows.

$$\mathrm{K}_{\mathrm{P},\dot{h}} = 0.0625, \mathrm{K}_{\mathrm{I},\dot{h}} = 0.5 \tag{3.7}$$

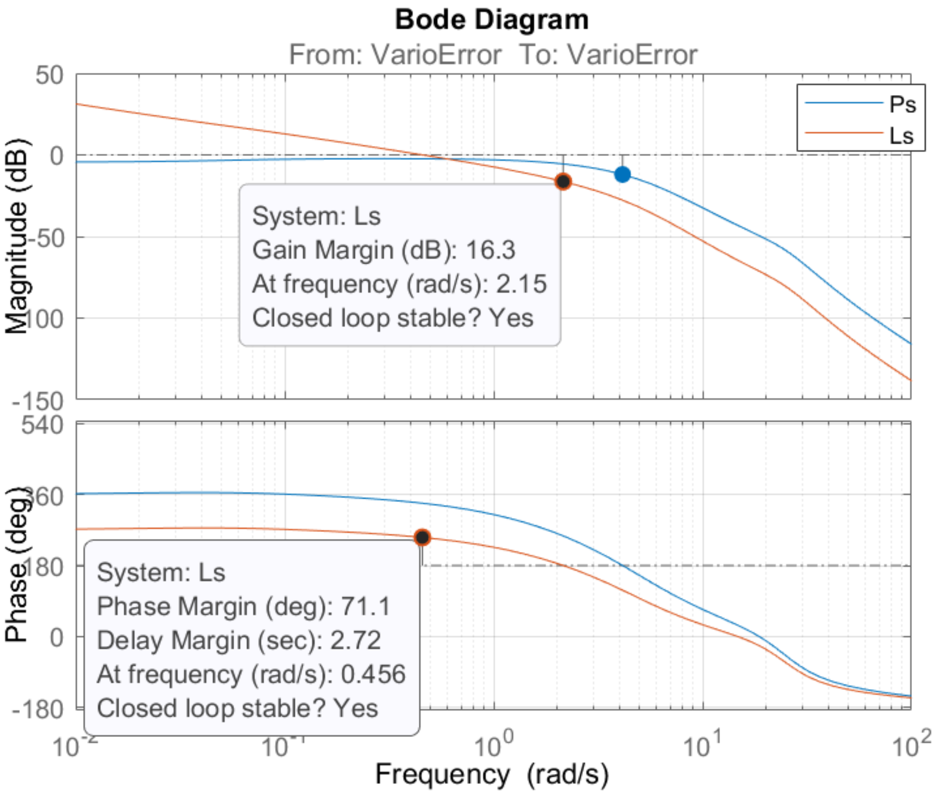

Figure 3.9 : Frequency Response – Vertical Speed Loop.

As can be seen in Figure 3.9, the system has a gain margin of 16.3dB, a phase margin of 71.1° and meets the specified design requirements. Figure 3.10 shows the linear model responses of the vertical speed loop at intervals of 5m/s from 30 to 60m/s.

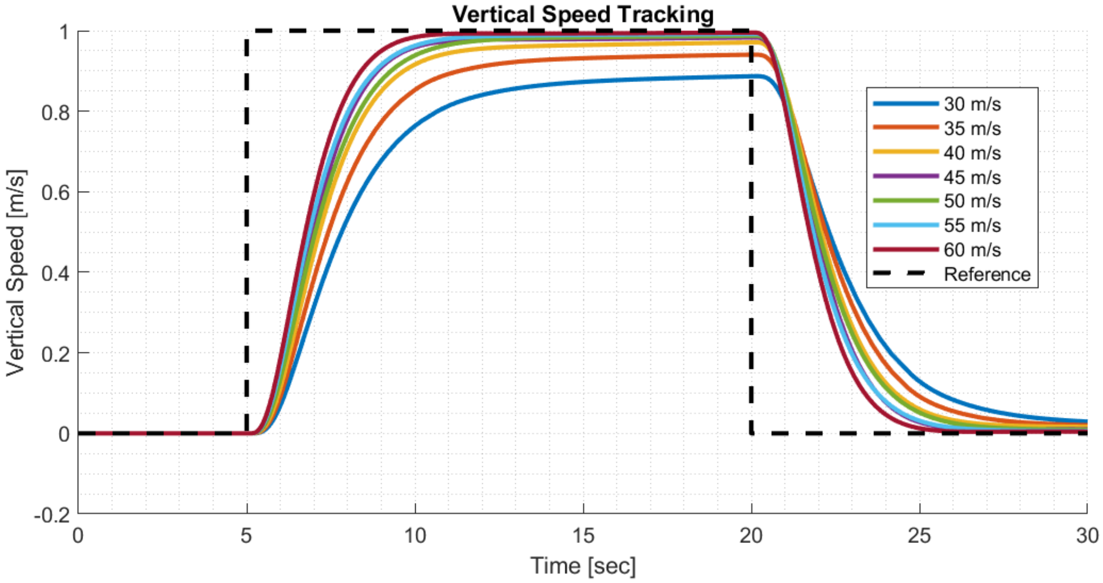

Figure 3.10 : Linear Model Singlet Response – Vertical Speed Loop.

After obtaining the gains of the vertical speed loop, as a second step, a proportional gain of the altitude loop is determined using the same process. With the help of the loop breaking method, the closed loop system is broken from the h error signal. Then, with the help of the control system design tool in MATLAB, the controller gains are found by considering the frequency and time responses. The open loop and closed loop frequency responses of the system are shown in Figure 3.11 for 45m/s trim condition. At the end, $\mathrm{K}_{\mathrm{P},h}$ gain is found as follows.

$$\mathrm{K}_{\mathrm{P},h} = 0.16 \tag{3.8}$$

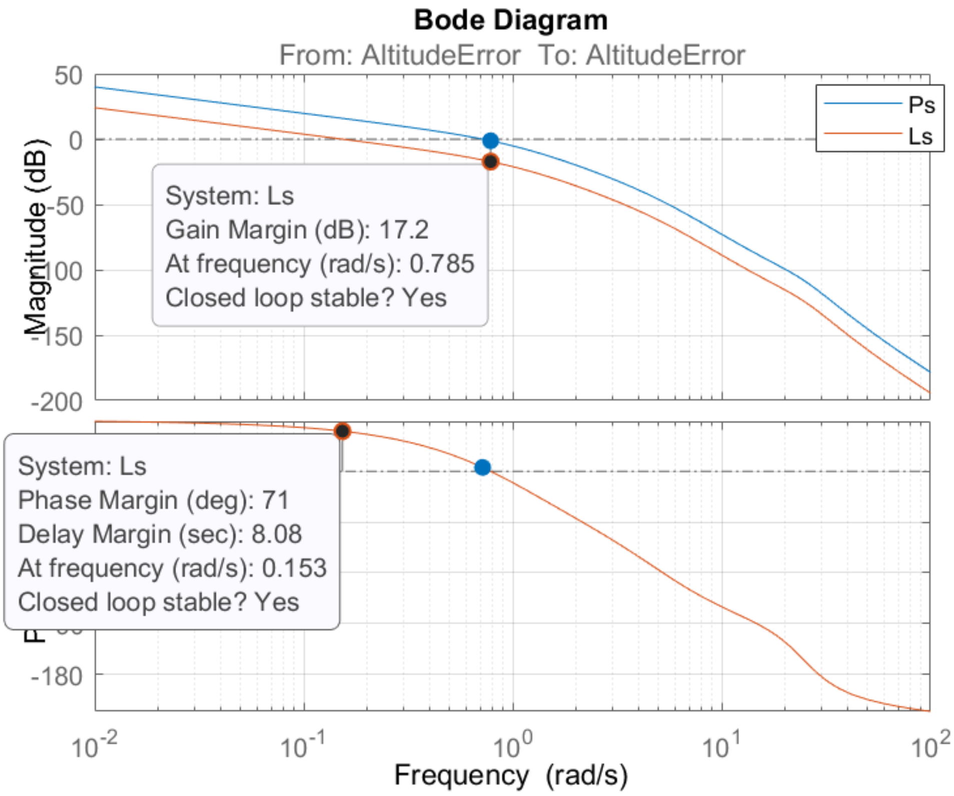

Figure 3.11 : Frequency Response – Altitude Loop.

As can be seen in Figure 3.11, the system has a gain margin of 17.2dB, a phase margin of 71° and meets the specified design requirements.

The same design processes is repeated for the entire envelope in which the autopilots will operate, and the following Table 3.2 is obtained.

Table 3.2 : Altitude Autopilot Controller Gain Table.

Air Speed (m/s)	$K_{P,\theta}$	$K_{P,\dot{h}}$	$K_{I,\dot{h}}$	$K_{P,h}$
30	1.2100	0.0750	0.6000	0.1600
35	1.2500	0.0750	0.6000	0.1600
40	1.2700	0.0750	0.6000	0.1600
45	1.3000	0.0750	0.6000	0.1600
50	1.3200	0.0625	0.5000	0.1600
55	1.4000	0.0625	0.5000	0.1600
60	1.4100	0.0625	0.5000	0.1600

Figure 3.12 shows the linear model responses of the altitude autopilot at intervals of 5m/s from 30 to 60m/s.

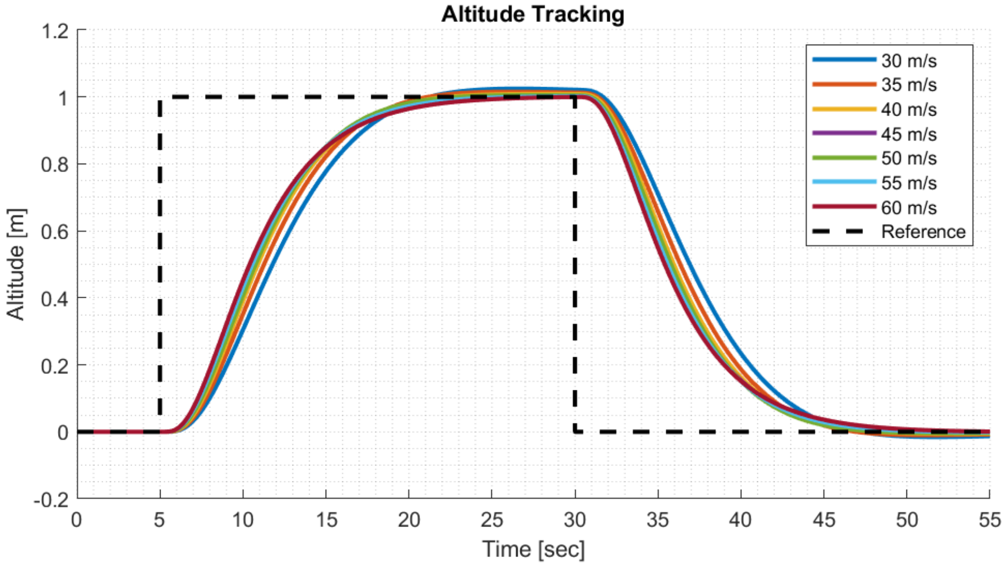

Figure 3.12 : Linear Model Singlet Response – Altitude Loop.

3.1.3 Speed autopilot

Another important state on the longitudinal axis is airspeed. The Speed Autopilot simply controls the airspeed of the aircraft with a PI controller. This autopilot is designed to ensure that the commanded airspeed is tracked. Speed Autopilot is designed to support Hold and Select mode. The requirements for the Speed Autopilot function are given as the Speed Autopilot shall maintain the existing true airspeed when the autopilot is engaged, the Speed Autopilot shall capture and maintain the selected true airspeed when the autopilot is engaged, and the Speed Autopilot shall ensure accuracy of ±1 m/s true airspeed.

The architecture of the Speed Autopilot is given in Figure 3.13. Pitch Attitude Autopilot is also included in the architecture because, as mentioned before, there is a dynamic relationship between the pitch angle and the airspeed of the aircraft. In order to design a more stable and robust speed autopilot, Pitch Attitude Autopilot is also included in the design processes.

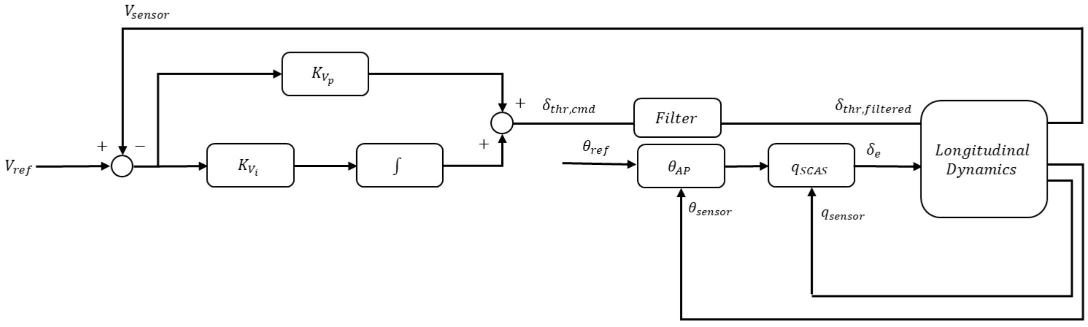

Figure 3.13 : Speed Autopilot Structure.

Sudden changes, sudden increases/decreases in the throttle command to the jet engine can cause problems such as engine throttle cut, and the formation of air bubbles in the fuel tank. To prevent these problems, a first order filter is added to the output of the Speed Autopilot. This first order filter aims to avoid various engine failures by smoothing the throttle command generated by the Speed Autopilot. The structure of the filter is given below.

$$TF = \frac{N_1 s + N_0}{D_1 s + D_0} \tag{3.9}$$

In Figure 3.13, V_{ref} refers to the reference airspeed and V_{sensor} refers to the current true airspeed of the aircraft. $\delta_{thr,cmd}$ and $\delta_{thr,filtered}$ are the throttle command generated by the Speed Autopilot and the first order filtered version of this command respectively.

In the design of Speed Autopilot, proportional integral (PI) control law is used. The control law for the Speed Autopilot is given in Eq. 3.10.

$$\delta_{thr,cmd} = K_{I,V} \int (V_{ref} - V_{sensor})\, dt + K_{P,V}(V_{ref} - V_{sensor}) \tag{3.10}$$

With the loop breaking method, the closed loop is broken from the V error signal. Then, with the help of the control system design tool in MATLAB, the controller gains are found by considering the frequency and time responses. The open loop and closed loop frequency responses are shown in Figure 3.14 for 45m/s trim condition. $\mathrm{K}_{\mathrm{P},V}$, and $K_{I,V}$ gains are found as follows.

$$\mathrm{K}_{\mathrm{P},V} = 15, \mathrm{K}_{\mathrm{I},V} = 2 \tag{3.11}$$

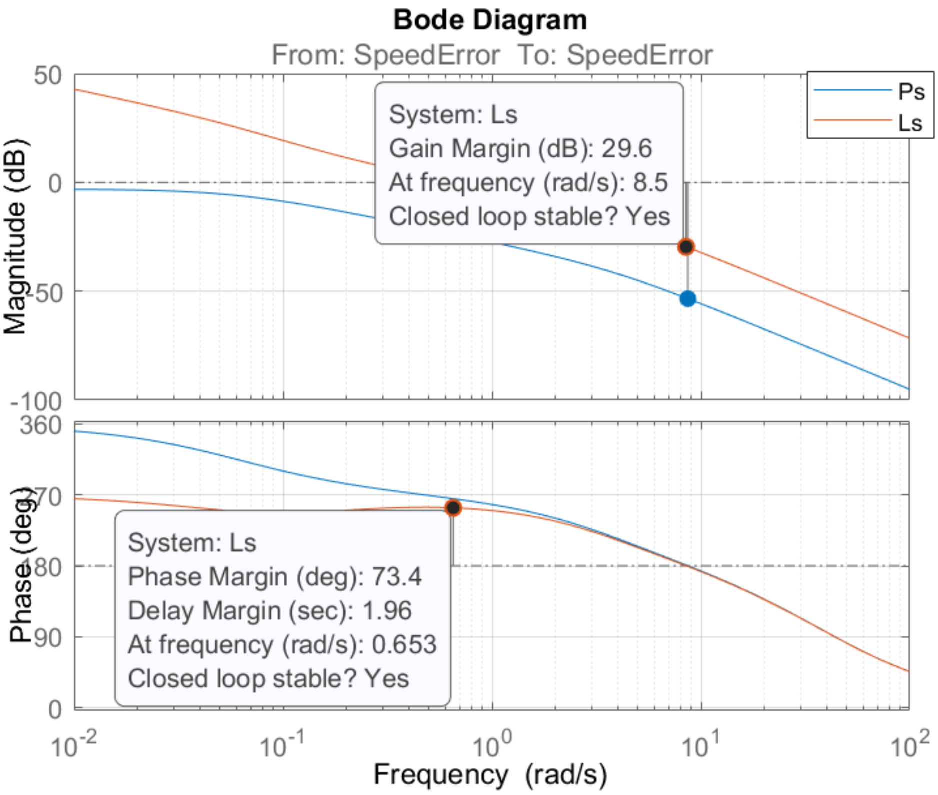

Figure 3.14 : Frequency Response – Speed Autopilot.

As can be seen in Figure 3.14, the system has a gain margin of 29.6dB, a phase margin of 73.4° and meets the specified design requirements. Figure 3.15 shows the linear model responses of the speed autopilot at intervals of 5m/s from 30 to 60m/s.

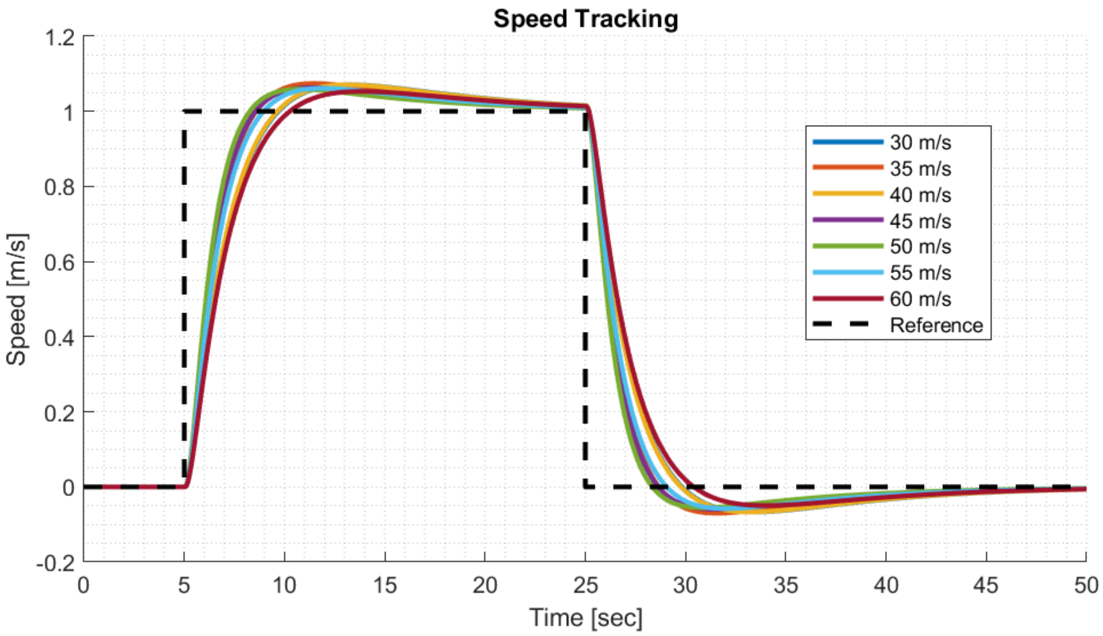

Figure 3.15 : Linear Model Singlet Response – Speed Autopilot.

Time constant of the first order filter is selected as $\tau = 0.25$. There is an example of the difference between auto throttle commands with and without first order filter in Figure 3.16.

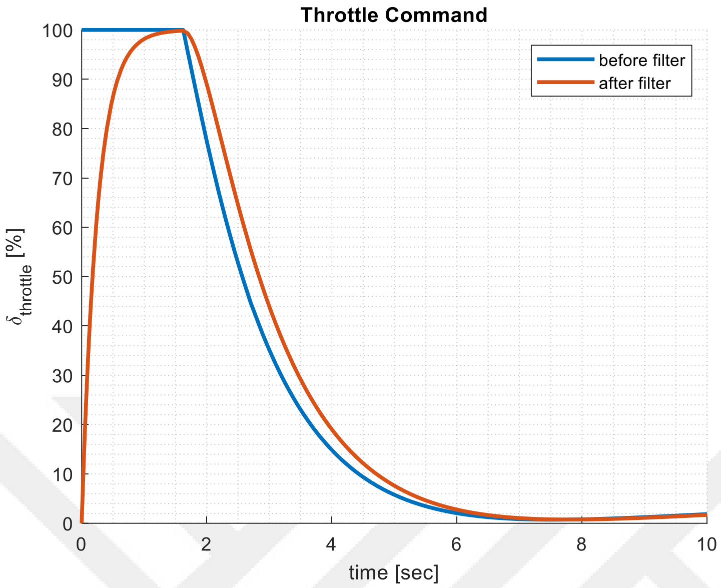

Figure 3.16 : Autothrottle Command with and without Filter.

3.2 TECS

The energy equations used in the TECS are based on classical Newtonian physics. For an aircraft modeled as a point mass, total energy, E_T is the sum of the kinetic and potential energy, and the rate of change is affected by both altitude and speed. The equation for total energy is given below. Where m is the mass, g is the acceleration due to gravity, h is the altitude, and v is the airspeed:

$$E_T = \frac{1}{2}mv^2 + mgh \tag{3.12}$$

By taking the time derivatives of this equation, the total energy rate, $\dot{E}_T$ to be used in the control system are obtained:

$$\dot{E}_T = mv\dot{v} + mg\dot{h} \tag{3.13}$$

Dividing total energy rate by mgv gives the non-dimensional specific energy rate, $\dot{E}_s$, which is controlled by the throttle:

$$\dot{E}_s = \frac{\dot{v}}{g} + \frac{\dot{h}}{v} = \frac{\dot{v}}{g} + \gamma \tag{3.14}$$

To manage the energy distribution between kinetic and potential forms, TECS uses the Lagrangian L, defined as the difference between kinetic and potential energy:

$$L = \frac{1}{2}mv^2 - mgh \tag{3.15}$$

$$\dot{L} = mv\dot{v} - mg\dot{h} \tag{3.16}$$

Dividing by mgv gives the specific energy distribution rate, $\dot{L_s}$, controlled by the elevator:

$$\dot{L_s} = \frac{\dot{v}}{g} - \frac{\dot{h}}{v} = \frac{\dot{v}}{g} - \gamma \tag{3.17}$$

where γ is the flight path angle. TECS employs proportional-integral (PI) control loops to minimize the errors in both $\dot{E}_s$ and $\dot{L_s}$. The throttle command δ_{T_c} and pitch command θ_c are determined as follows:

$$\delta_{T_c} = K_{TI}\int(\dot{E_{s_{cmd}}} - \dot{E}_s)dt + K_{TP}\dot{E}_s \tag{3.18}$$

$$\theta_c = K_{EI}\int(\dot{L_{s_{cmd}}} - \dot{L_s})dt + K_{EP}\dot{L_s} \tag{3.19}$$

where K_{TI} and K_{EI} are integral gains, and K_{TP} and K_{EP} are proportional gains. This dual-loop architecture ensures that the aircraft's total energy is adjusted primarily through throttle modulation, while the elevator redistributes energy between kinetic and potential forms, reducing cross-coupling effects. When these equations are integrated into the longitudinal controller architecture, the structure in Figure 3.17 is obtained.

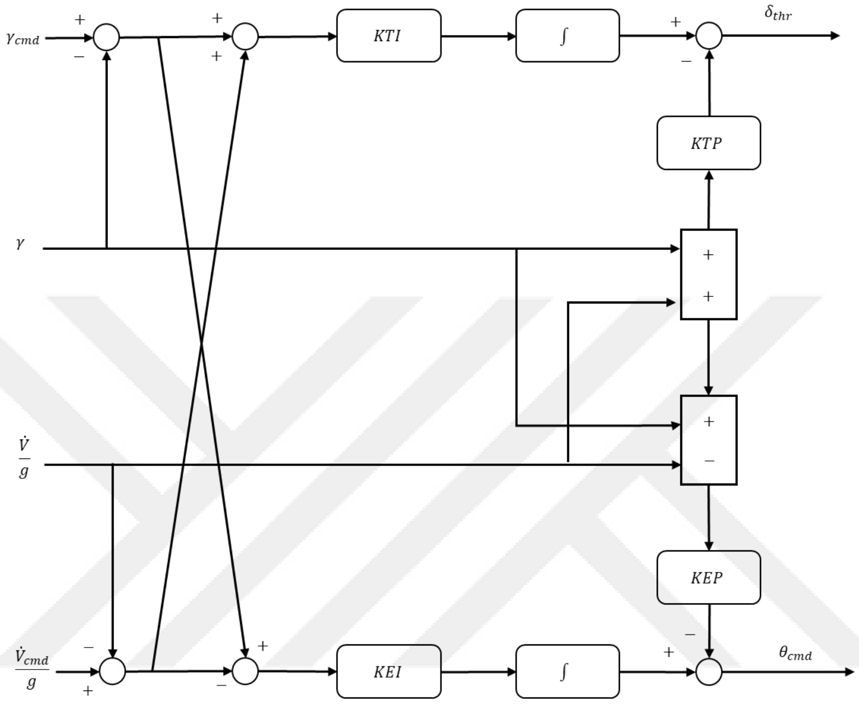

Figure 3.17 : TECS Structure.

The control loop created with these reference values synchronizes the throttle and elevator movements consistently in energy space. As the system monitors the speed and altitude commands simultaneously, there is no conflict of tasks between the control surfaces. With this architecture, the throttle only increases/decreases energy, while the elevator functions to direct energy components in favor of airspeed or altitude. Lambregts noted that this separation of duties allows the system to respond both faster and more stable (Lambregts, 1987).

The design method for finding the controller gains for TECS is loop breaking, as in the autopilot design section. Unlike the autopilot design, TECS has a multi-input multi-output system, and more complex design steps are followed. As mentioned before, TECS generates throttle and pitch angle commands. The γ loop generates the throttle command, δ_{thr}, and the $\dot{V}$ loop generates the pitch angle command, θ_{ref} which is a reference to Pitch Attitude Autopilot. When designing, both $\dot{\mathrm{V}}$ and γ loops are designed separately by following the one-loop-at-a-time method.

While designing the controller for TECS, the following steps were analyzed.

Step 1 – Couple effects are not included. Freeze the throttle input and leave only the $\dot{\mathrm{V}}$ loop active and obtain the first $\mathrm{K_{EI}}$ and $\mathrm{K_{EP}}$ gains that meet the design criteria.

Step 2 – Couple effects not included. Freeze the pitch angle command and leave only the γ loop active and obtain the first $\mathrm{K_{TI}}$ and $\mathrm{K_{TP}}$ gains that meet the design criteria.

Step 3 – Couple effects are not included. both loops are active. for the $\dot{\mathrm{V}}$ loop that generates the pitch angle command, the $\mathrm{K_{EI}}$ and $\mathrm{K_{EP}}$ gains obtained in the first step are used. $\mathrm{K_{TI}}$ and $\mathrm{K_{TP}}$ gains are re-tuned in the new system.

Step 4 – Couple effects are not included. Both cycles are active. For the γ loop that generates the Throttle command, the $\mathrm{K_{TI}}$ and $\mathrm{K_{TP}}$ gains obtained in the third step are used. $\mathrm{K_{EI}}$ and $\mathrm{K_{EP}}$ gains are re-tuned in the new system.

Step 5 – Couple effects included. Both loops are active. For the γ loop that generates the Throttle command, the $\mathrm{K_{TI}}$ and $\mathrm{K_{TP}}$ gains obtained in step four are used. $\mathrm{K_{EI}}$ and $\mathrm{K_{EP}}$ gains are re-tuned in the new system.

Step 6 – Couple effects included. Both loops are active. For the $\dot{\mathrm{V}}$ loop that generates the pitch angle command, the $\mathrm{K_{EI}}$ and $\mathrm{K_{EP}}$ gains obtained in step 5 are used. $\mathrm{K_{TI}}$ and $\mathrm{K_{TP}}$ gains are re-tuned in the new system.

Step 7 – Couple effects included. Both loops are active. For the γ loop that generates the Throttle command, the $\mathrm{K_{TI}}$ and $\mathrm{K_{TP}}$ gains obtained in step six are used. $\mathrm{K_{EI}}$ and $\mathrm{K_{EP}}$ gains are re-tuned in the new system.

Step 8 – After tuning $\mathrm{K_{EI}}$, K_{EP}, K_{TI}, and $\mathrm{K_{TP}}$ gains, $\mathrm{K_h}$ and $\mathrm{K_v}$ gains are designed to control the altitude and airspeed, respectively.

After these eight steps, DEHA's TECS controller design is finalized. In the rest of the study the design method is explained in detail with bode plots and stability margins.

In the first step, unlike conventional autopilots, the couple effects are first ignored. Then, the throttle input to the longitudinal axis system dynamics is set to 0. The controller is designed only for the $\dot{\mathrm{V}}$ loop. Figure 3.18 shows the architecture used for the first design step.

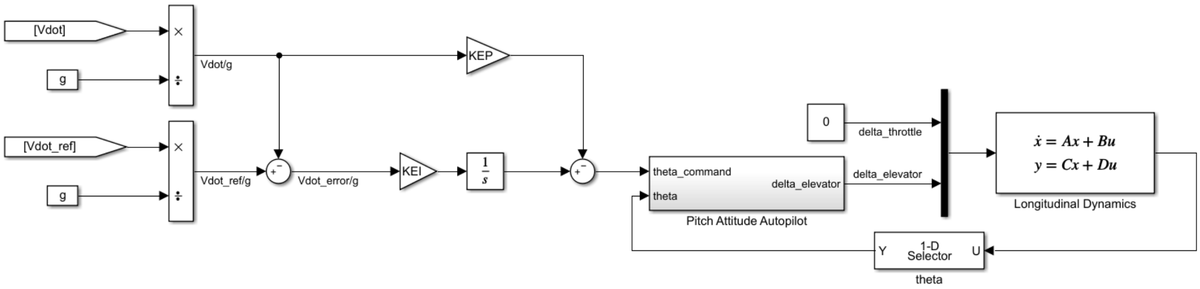

Figure 3.18 : Controller Design Architecture – Step 1.

With the loop-breaking method, the closed loop is broken from the $\dot{\mathrm{V}}$ error signal. Then, with the help of the control system design tool in MATLAB, the controller gains are found by considering the frequency and time responses. The open loop and closed loop frequency responses of the system are shown in Figure 3.19. At the end of the first step, $\mathrm{K_{EI}}$ and $\mathrm{K_{EP}}$ gains are found as follows.

$$\mathrm{K_{EI}} = 1.4, \mathrm{K_{EP}} = 0.5 \tag{3.20}$$

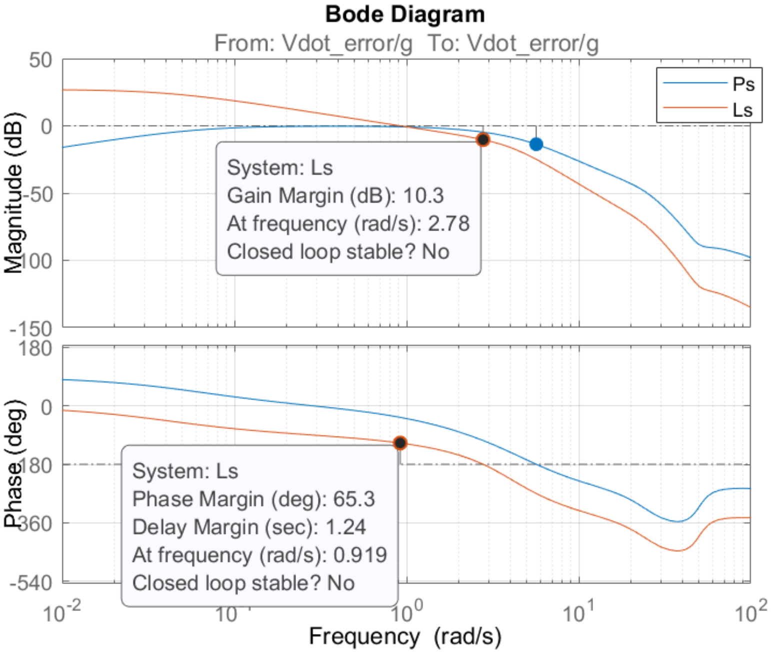

Figure 3.19 : Frequency Response – Step 1.

As can be seen in Figure 3.19, the system has a gain margin of 10.3dB, a phase margin of 65.3° and meets the specified design requirements.

In the second step, the couple effects are ignored as in the first step. Then, the pitch angle input to the longitudinal axis system dynamics is set to 0. The controller is designed only for the γ loop. Figure 3.20 shows the architecture used for the second design step.

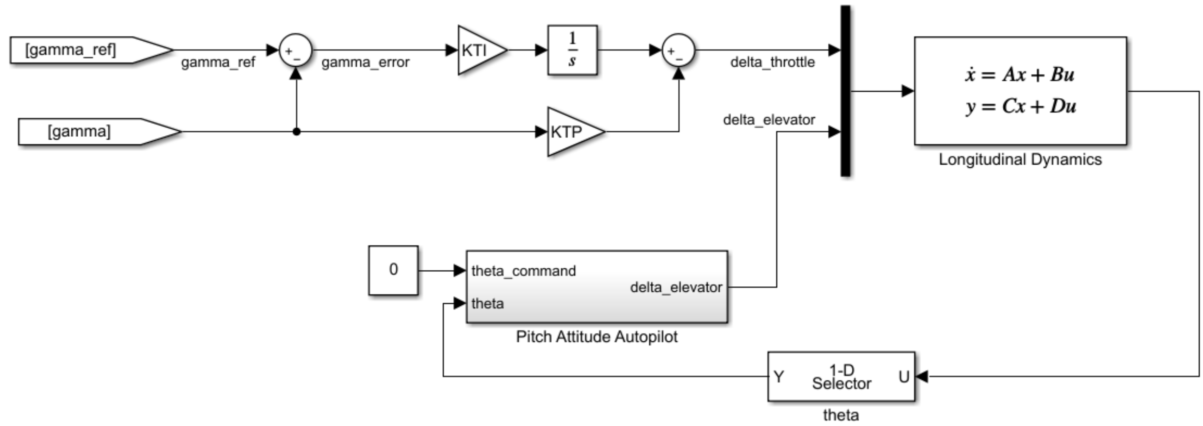

Figure 3.20 : Controller Design Architecture – Step 2.

With the loop-breaking method, the closed loop is broken from the γ error signal. Then, with the help of design tool in MATLAB, the controller gains are found by considering the frequency and time responses. The open loop and closed loop frequency responses of the system are shown in Figure 3.21. At the end of the second step, $\mathrm{K_{TI}}$ and $\mathrm{K_{TP}}$ gains are found as follows.

$$\mathrm{K_{TI}} = 8, \mathrm{K_{TP}} = 40 \tag{3.21}$$

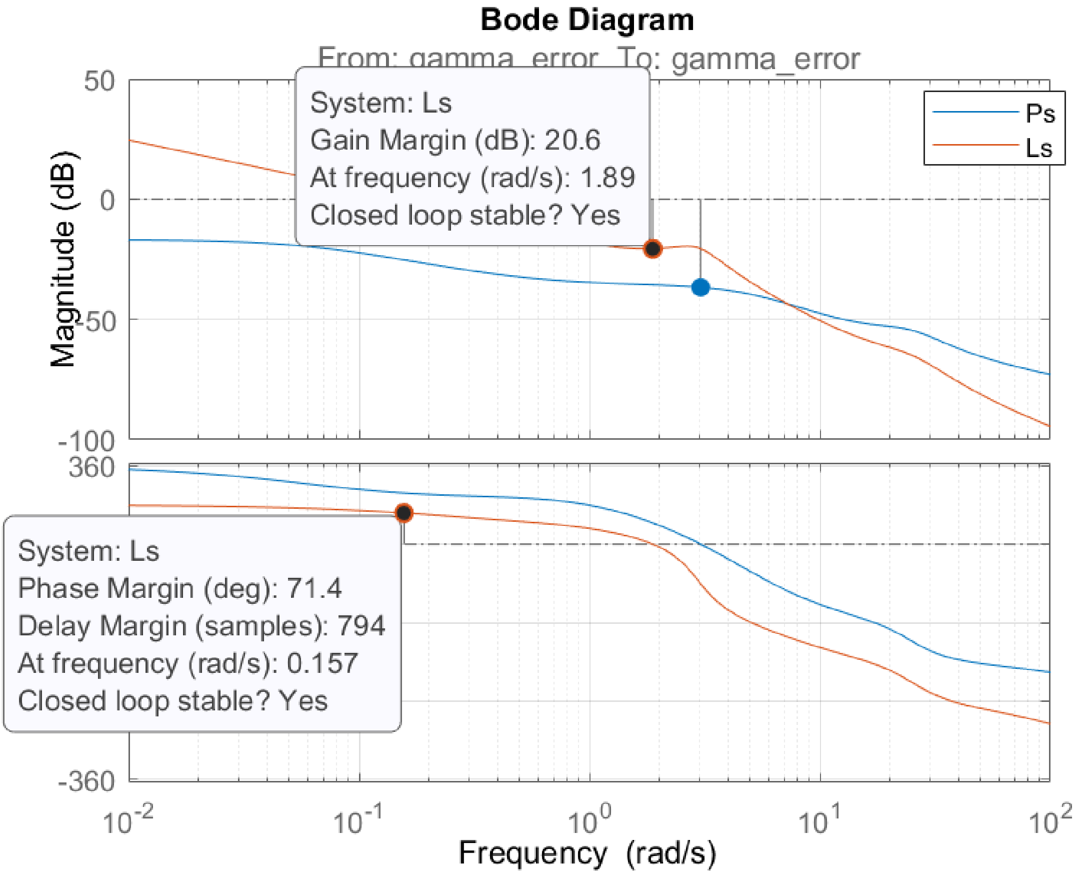

Figure 3.21 : Frequency Response – Step 2.

As can be seen in Figure 3.21, the system has a gain margin of 20.6dB, a phase margin of 71.4° and meets the specified design requirements.

In the third step, the couple effects are also ignored in this step. Both $\dot{V}$ and γ controllers are in the loop. For the $\dot{V}$ loop that generates the pitch angle command, $K_{EI} = 1.4$ and $K_{EP} = 0.5$ obtained in the first step are used. The controller is designed only for the γ loop. Figure 3.22 shows the architecture used for the third design step.

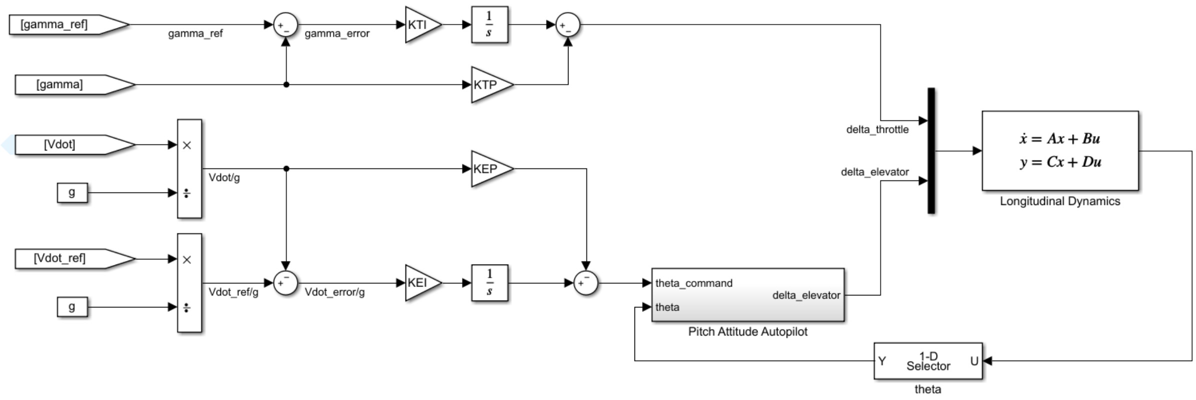

Figure 3.22 : Controller Design Architecture – Step 3.

With the loop-breaking method, the closed loop is broken from the γ error signal. Then, with the help of the design tool in MATLAB, the controller gains are found by considering the frequency and time responses. The open loop and closed loop frequency responses of the system are shown in Figure 3.23. At the end of the third step, K_{TI} and K_{TP} gains are found as follows.

$$K_{TI} = 1, K_{TP} = 0.2 \tag{3.22}$$

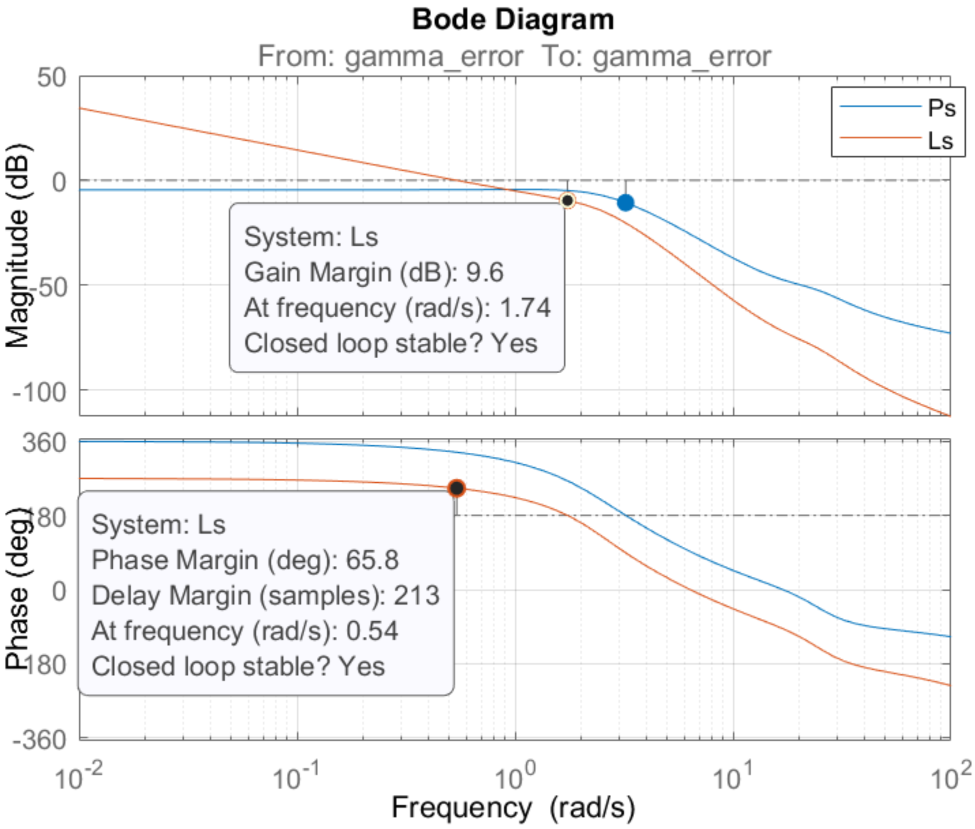

Figure 3.23 : Frequency Response – Step 3.

As can be seen in Figure 3.23, the system has a gain margin of 9.6dB, a phase margin of 65.8° and meets the specified design requirements.

In the fourth step, the couple effects are also ignored in this step. Both $\dot{V}$ and γ controllers are in the loop. For the γ loop that generates the throttle command, $K_{TI} = 1$ and $K_{TP} = 0.2$ obtained in the third step are used. The controller is designed only for the $\dot{V}$ loop. The architecture used in the previous step is also used in this step.

With the loop-breaking method, the closed loop is broken from the $\dot{V}$ error signal. Then, with the help of the design tool in MATLAB, the controller gains are found by considering the frequency and time responses. The open loop and closed loop frequency responses of the system are shown in Figure 3.24. At the end of the third step, K_{TI} and K_{TP} gains are found as follows.

$$K_{EI} = 0.8, K_{EP} = 1.45 \tag{3.23}$$

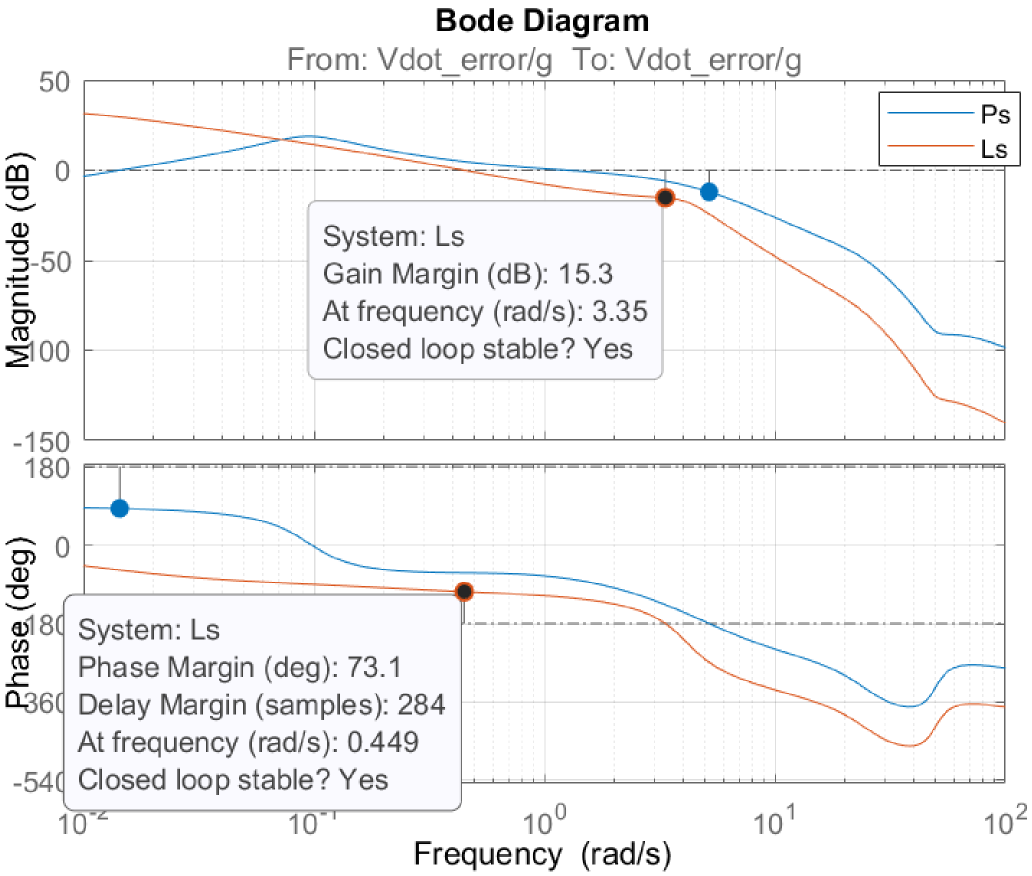

Figure 3.24 : Frequency Response – Step 4.

As can be seen in Figure 3.24, the system has a gain margin of $15.3dB$, a phase margin of 73.1° and meets the specified design requirements.

In the fifth step, both $\dot{V}$ and γ controllers and the couple effects are in the loop. For the γ loop that generates the throttle command, $K_{TI} = 1$ and $\mathrm{K_{TP}} = 0.2$ obtained in the third step are used. The controller is designed only for the $\dot{\mathrm{V}}$ loop. Figure 3.25 shows the architecture used for the fifth design step.

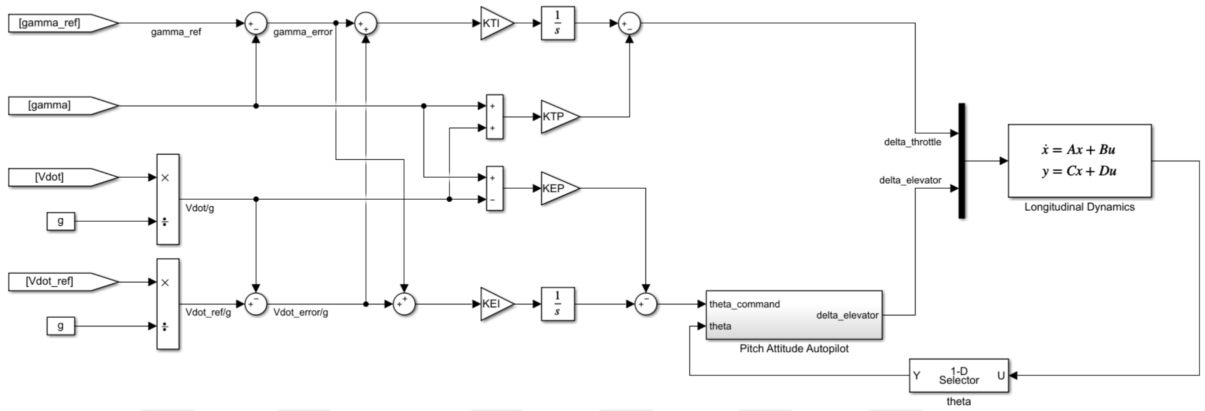

Figure 3.25 : Controller Design Architecture – Step 5.

With the loop-breaking method, the closed loop is broken from the $\dot{\mathrm{V}}$ error signal. Then, with the help of the design tool in MATLAB, the controller gains are found by considering the frequency and time responses. The open loop and closed loop frequency responses of the system are shown in Figure 3.26. At the end of the fifth step, $\mathrm{K_{EI}}$ and $\mathrm{K_{EP}}$ gains are re-tuned as follows.

$$\mathrm{K_{EI}} = 0.56, \mathrm{K_{EP}} = 0.3 \tag{3.24}$$

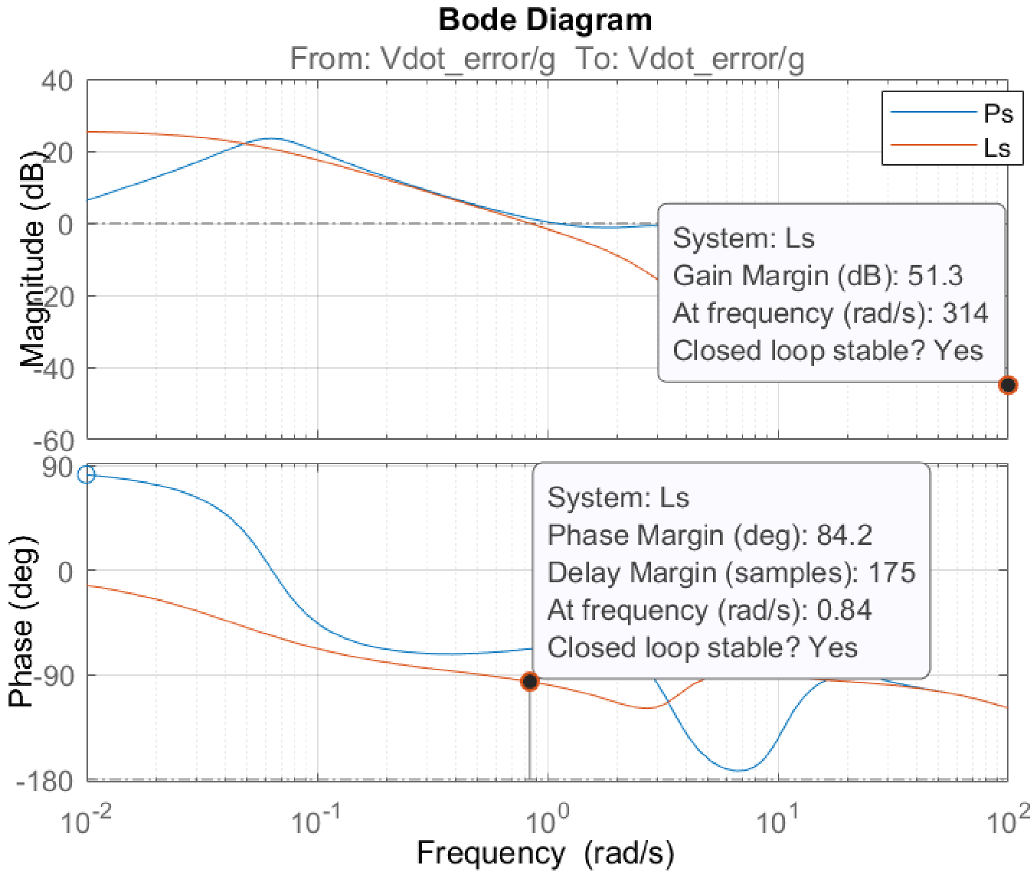

Figure 3.26 : Frequency Response – Step 5.

As can be seen in Figure 3.26, the system has a gain margin of 51.3dB, a phase margin of 84.2° and meets the specified design requirements.

In the sixth step, both $\dot{V}$ and γ controllers and the couple effects are in the loop. For the $\dot{V}$ loop that generates the pitch angle command, $K_{EI} = 0.56$ and $K_{EP} = 0.3$ obtained in the previous step are used. The controller is designed only for the γ loop. The architecture used in the previous step is also used in this step.

With the loop-breaking method, the closed loop is broken from the γ error signal. Then, with the help of the design tool in MATLAB, the controller gains are found by considering the frequency and time responses. The open loop and closed loop frequency responses of the system are shown in Figure 3.27. At the end of the fifth step, K_{TI} and K_{TP} gains are re-tuned as follows.

$$K_{TI} = 4, K_{TP} = 1.6 \tag{3.25}$$

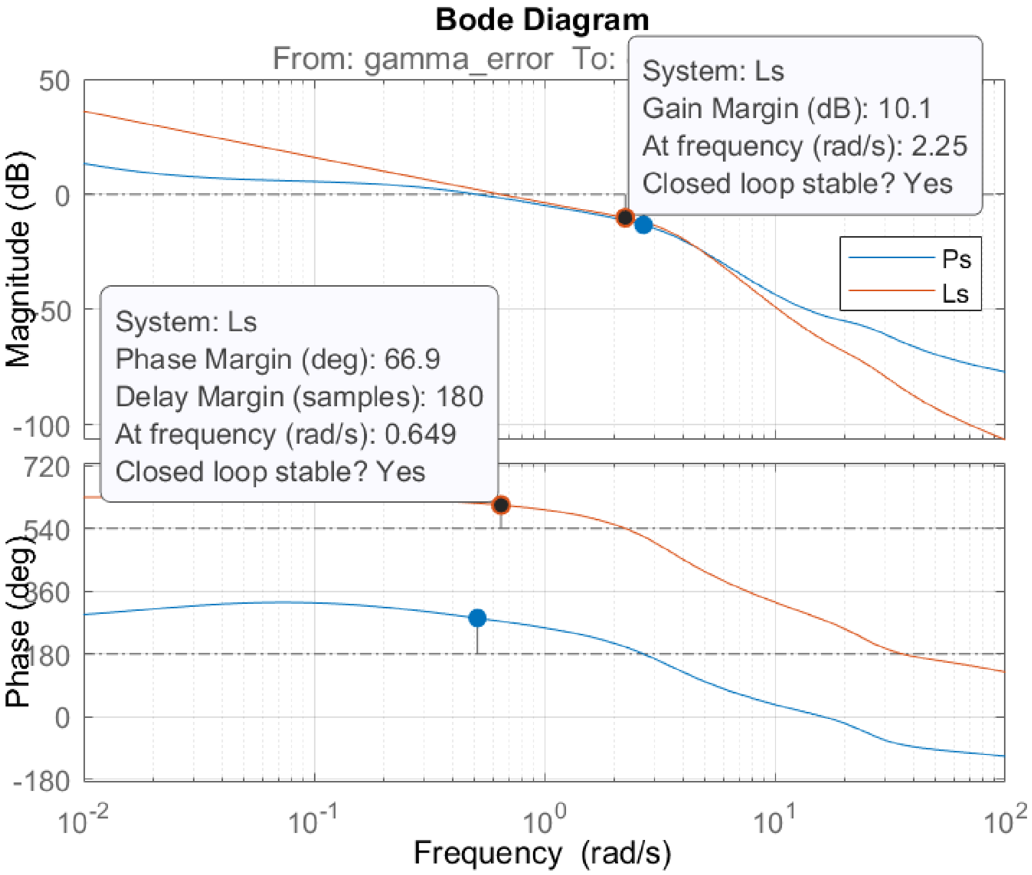

Figure 3.27 : Frequency Response – Step 6.

As can be seen in Figure 3.27, the system has a gain margin of 10.1dB, a phase margin of 66.9° and meets the specified design requirements.

In the final step, both $\dot{V}$ and γ controllers and the couple effects are in the loop. For the γ loop that generates the throttle command, $K_{TI} = 4$ and $K_{TP} = 1.6$ obtained in the previous step are used. The controller is designed only for the $\dot{V}$ loop. The architecture used in the fifth step is also used in this step.

With the loop-breaking method, the closed loop is broken from the $\dot{V}$ error signal. Then, with the help of the design tool in MATLAB, the controller gains are found by considering the frequency and time responses. The open loop and closed loop frequency responses of the system are shown in Figure 3.28. At the end of the fifth step, K_{EI} and K_{EP} gains are re-tuned as follows.

$$K_{EI} = 0.6, K_{EP} = 0.3 \qquad (3.26)$$

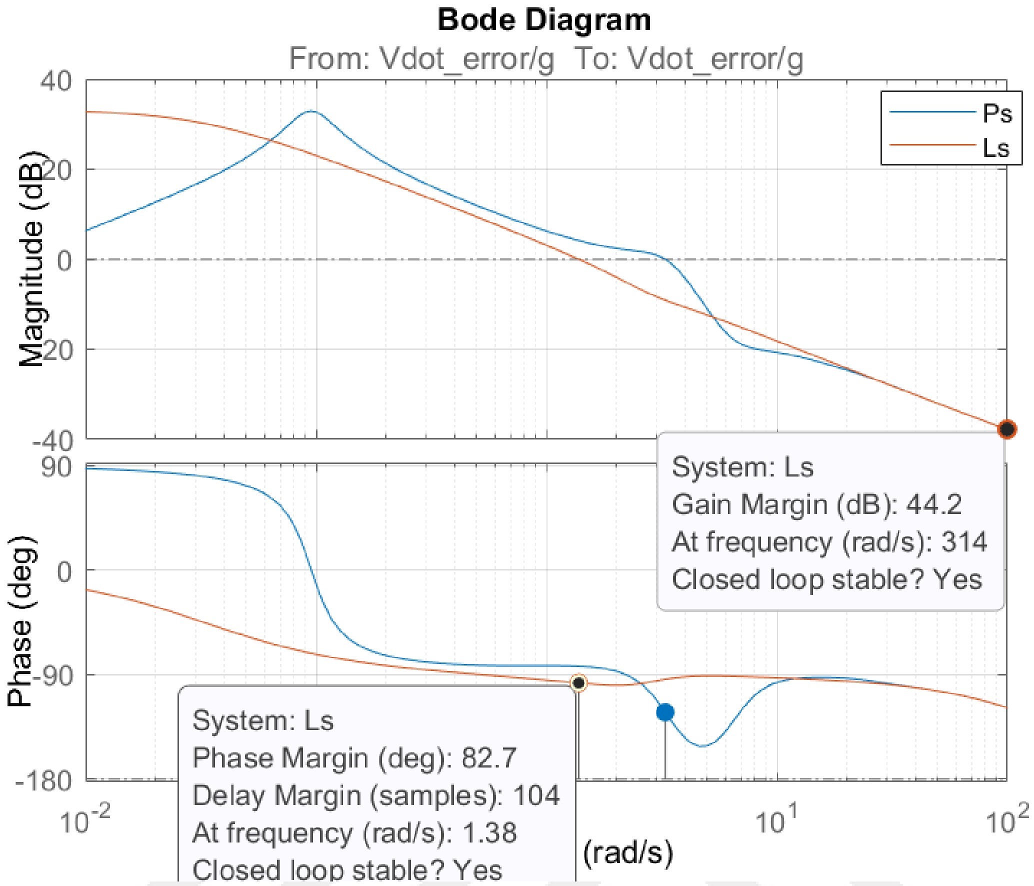

Figure 3.28 : Frequency Response – Step 7.

As can be seen in Figure 3.28, the system has a gain margin of 44.2dB, a phase margin of 82.7° and meets the specified design requirements.

After designing the $\dot{V}$ and γ controllers, the next step is to find the altitude and airspeed controllers. When it is desired to use TECS architecture to hold the altitude and airspeed of the aircraft, $\dot{V}$ and γ commands should be generated using altitude and airspeed data. The following equations and controller architecture as shown in Figure 3.29, are used to generate these commands.

$$\gamma_{ref} = \frac{(h_{ref} - h)K_h}{V} \tag{3.27}$$

$$\dot{V}_{ref} = (V_{ref} - V)K_v \tag{3.28}$$

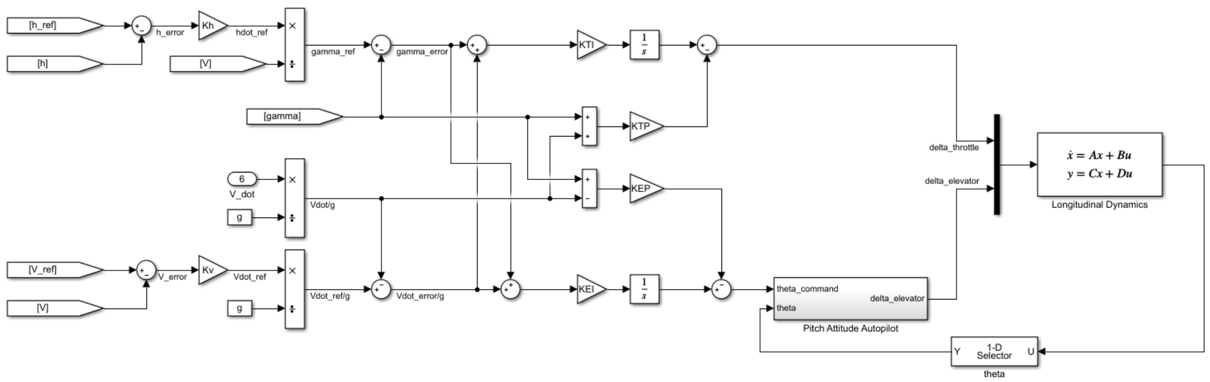

Figure 3.29 : Altitude and Speed Autopilot with TECS.

The same loop breaking method is used to find the gain for the altitude and speed controllers. First, the altitude loop is broken from h_{error} signal and a proportional controller is designed. The same design requirements are suitable for the altitude controller. In Figure 3.30, the open loop and closed loop frequency response can be seen. After the design with the help of MATLAB Control System Tool, altitude controller gain is set to $K_h = 0.3$. In Figure 3.31 linear model singlet response of the altitude loop can be seen for the flight envelope.

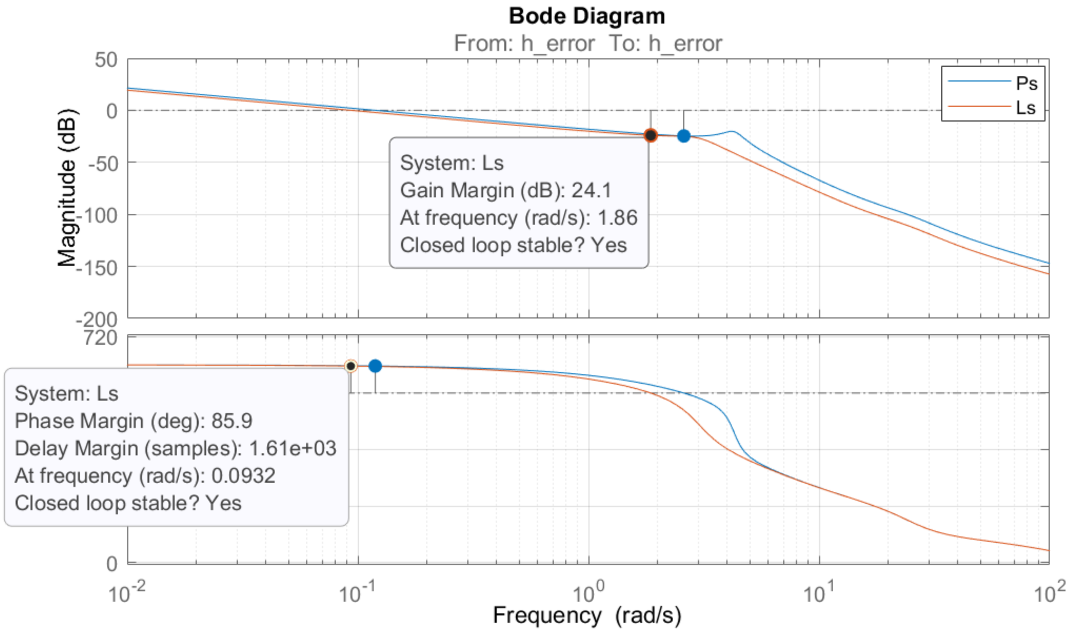

Figure 3.30 : Frequency Response – Step 8, Altitude Loop.

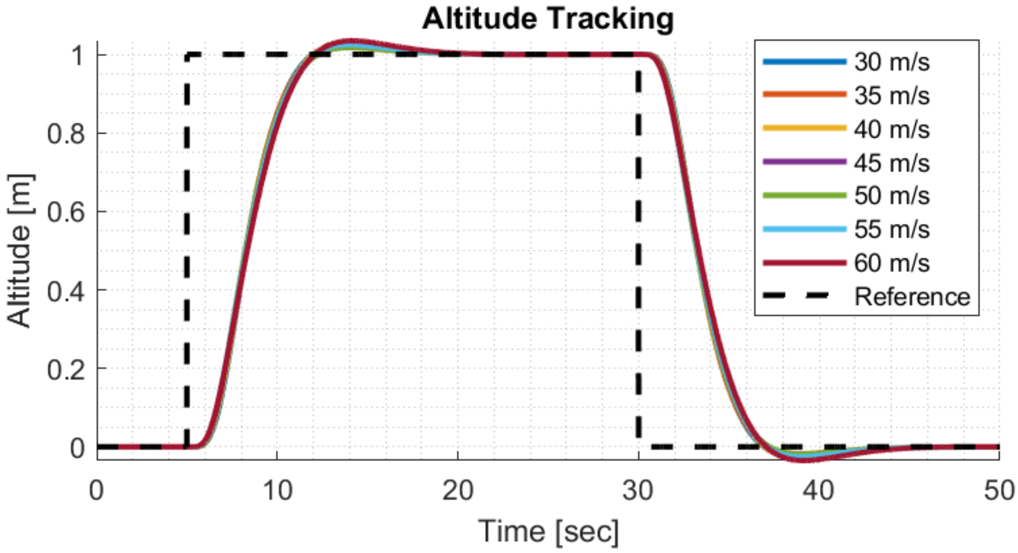

Figure 3.31 : Linear Model Singlet Response – Step 8, Altitude Loop.

Finally, the speed loop is broken from V_{error} signal and a proportional controller is also designed for speed controller. In Figure 3.32, the open loop and closed loop frequency response can be seen. After the design, speed controller gain is set to $K_v = 0.6$. In Figure 3.33 linear model singlet response of the speed loop can be seen for the flight envelope.

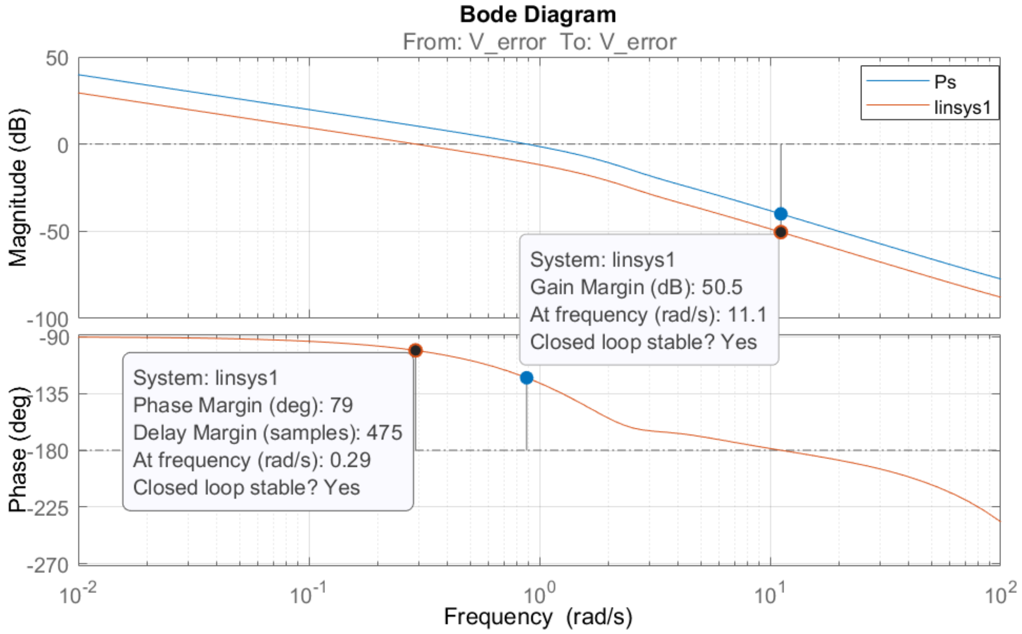

Figure 3.32 : Frequency Response – Step 8, Speed Loop.

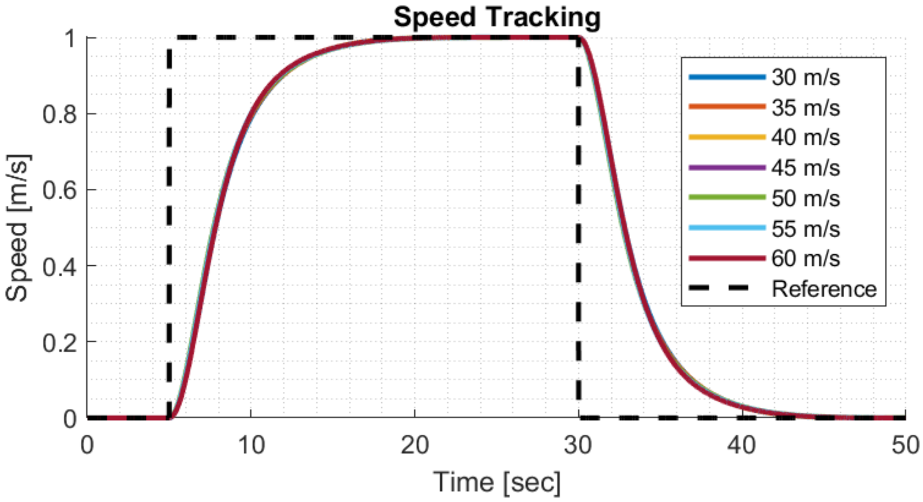

Figure 3.33 : Linear Model Singlet Response – Step 8, Speed Loop.

Here, as each trim condition started from a different initial speed, the corresponding speed responses were normalized by subtracting their respective trim values. This allowed all results to be plotted on a common reference axis, providing a clearer comparison by eliminating the visual complexity caused by the varying starting points.

According to Lambregts, the control strategy aims to reduce altitude and speed errors exponentially, using a time constant τ that decreases as K_h and K_v increase. To keep altitude and speed control decoupled and ensure the correct balance of energy between them, K_h and K_v should be set equal (Lambregts, A., 1983). Based on this information and nonlinear tuning processes, it was decided that $K_h = K_v = 0.3$.

4. SIMULATION RESULTS

In this section, simulation results of conventional longitudinal autopilots and TECS will be compared. The maneuvers listed in Table 4.1 were simulated to compare these two architectures.

Table 4.1 : Test Maneuvers.

Maneuver Number	Maneuver Description
1	Constant Altitude & Constant Speed
2	Constant Altitude & 5m/s Acceleration
3	Constant Altitude & 5m/s Deceleration
4	10m Climb & Constant Speed
5	10m Climb & 5m/s Acceleration
6	10m Climb & 5m/s Deceleration
7	10m Descent & Constant Speed
8	10m Descent & 5m/s Acceleration
9	10m Descent & 5m/s Deceleration
10	Constant Altitude & Constant Speed & 60º Bank

The most suitable maneuvers for comparing controller performance in the longitudinal axis are test scenarios in which speed and altitude change simultaneously in the same or different directions. Because these two states are energetically correlated and compete with each other, so to speak. Therefore, when comparing the two architectures, maneuvers in Table 4.1 were tested. Also, a test scenario where the aircraft suddenly banked was added to the maneuvers. When the aircraft is in high bank maneuver, it may lose altitude due to the reduction of lift in the longitudinal axis, and in this case, the performance of the longitudinal axis controllers should be analyzed additionally.

In the given test scenario in Figure 4.1, the altitude of the aircraft is held while the airspeed is decreased by 5 m/s with a bank angle 60 º. Here, the blue solid line represents TECS, and the orange solid line represents the conventional autopilots. Also, the dashed black lines represent the references, and the transparent areas show the accuracy bands. When the test results are analyzed, it is seen that both control architectures are designed to complete the maneuver in a way that meets the design requirements. However, the performance of TECS is slightly better compared to conventional autopilots. In TECS results, both altitude and airspeed data deviated less from the reference values, and the throttle command generated by TECS is less than that of conventional autopilots.

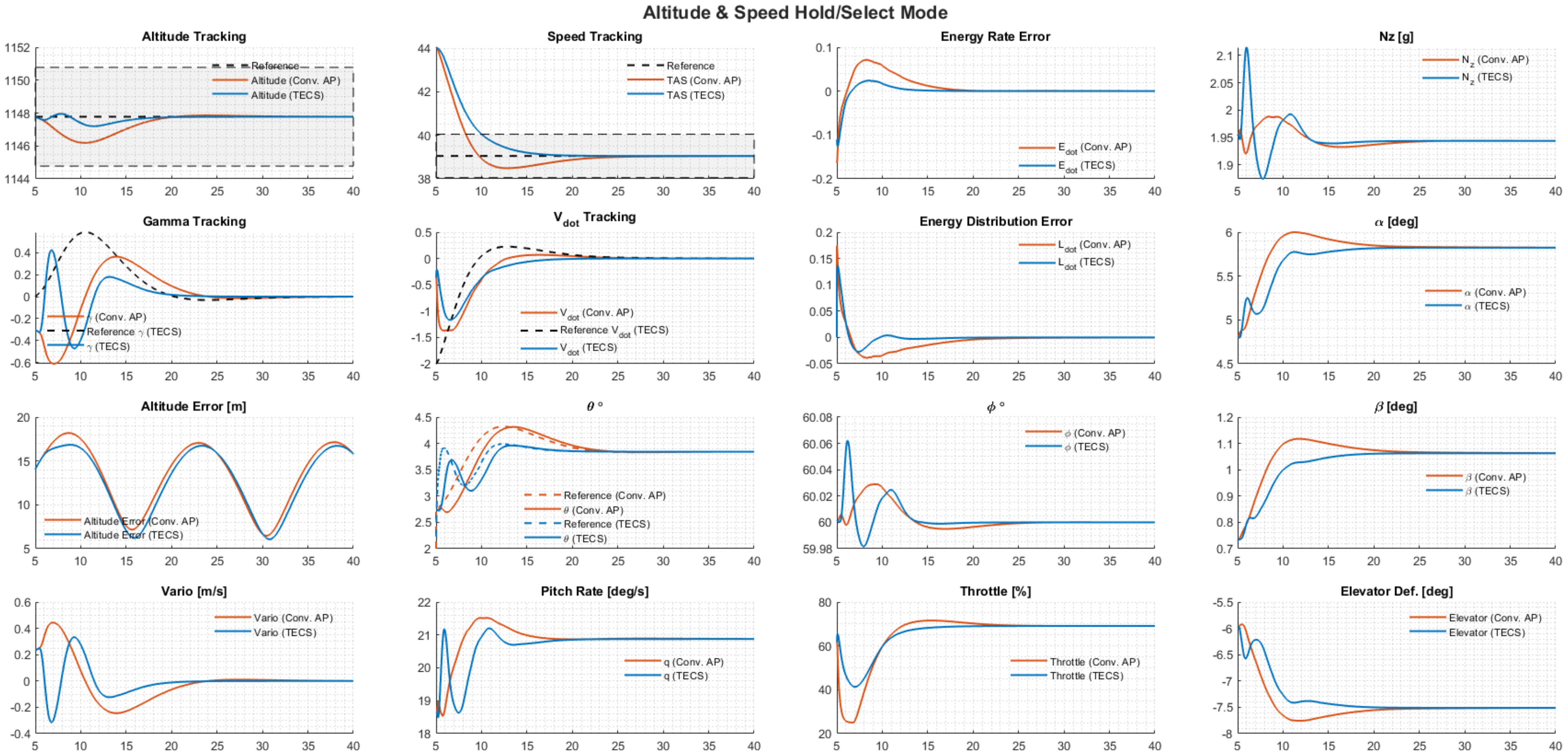

Figure 4.1 : Altitude Hold & 5m/s Deceleration.

In the given test scenario in Figure 4.2 and Figure 4.3, it was requested to increase the altitude of the aircraft by 10m while decreasing the airspeed by 5m/s in cruise and the presence of a 60º bank angle, respectively. As can be seen in Figure 4.2 and Figure 4.3, both architectures completed the maneuver to meet the requirements. The altitude is within the accuracy band of ±3m, and the airspeed is within the accuracy band of ±1m/s. The biggest difference between them is in the energy rate and the throttle command. The energy rate of TECS is going to zero, which is much easier compared to conventional autopilots. Also, the throttle command generated by TECS is lower than that of conventional autopilots. This is because TECS uses throttle and elevator together for these maneuvers, while conventional autopilots manage the airspeed and altitude loops separately. The aircraft is requested to decelerate while climbing, the aircraft is already losing airspeed due to the energy conversation, but since conventional autopilots control these two states separately with elevator and throttle channels, they ignore the energy change and focus on the two loops independently. However, TECS uses throttle and elevator as a couple with energy rates and successfully completes the maneuvers by generating fewer throttle commands.

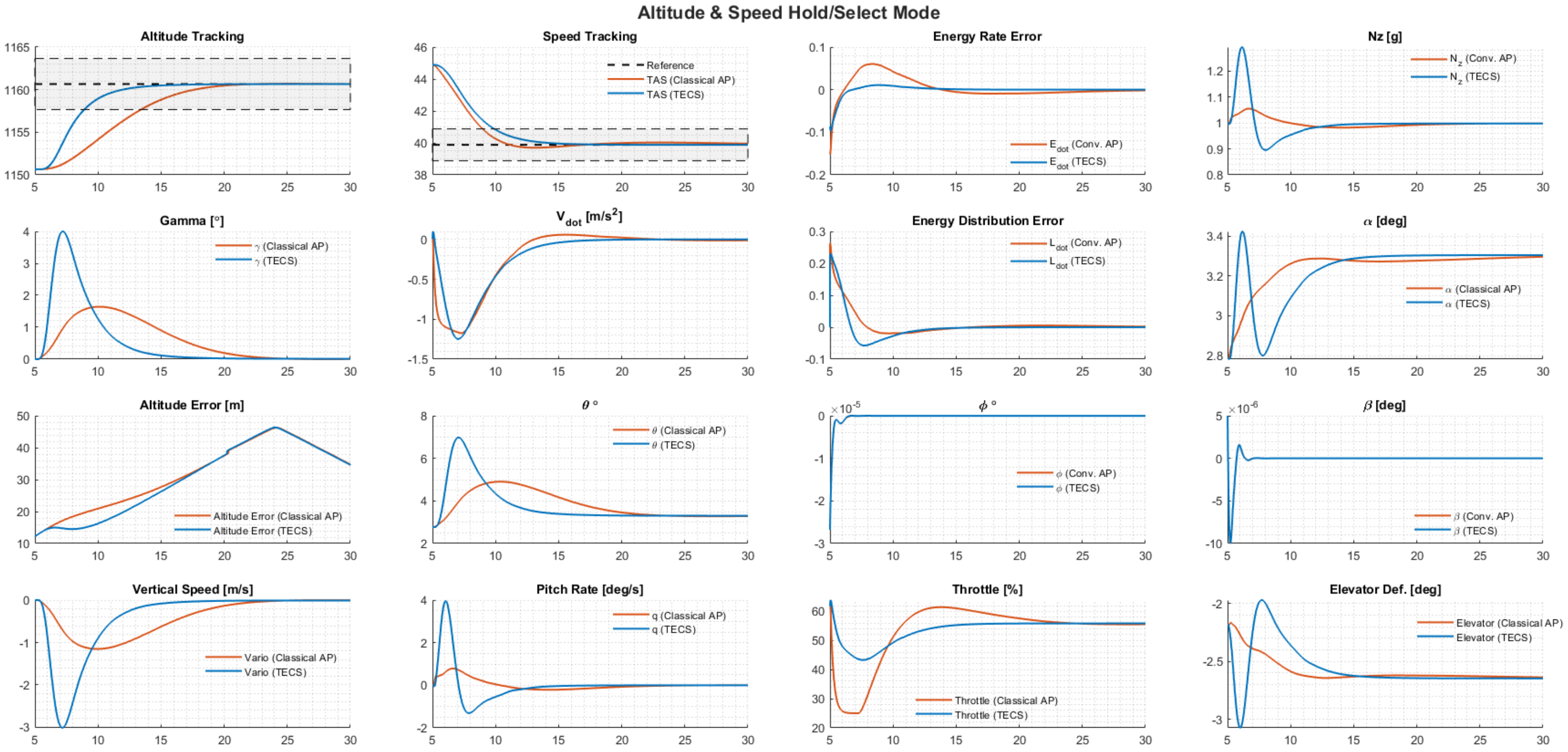

Figure 4.2 : 10m Climb & 5m/s Deceleration – Cruise.

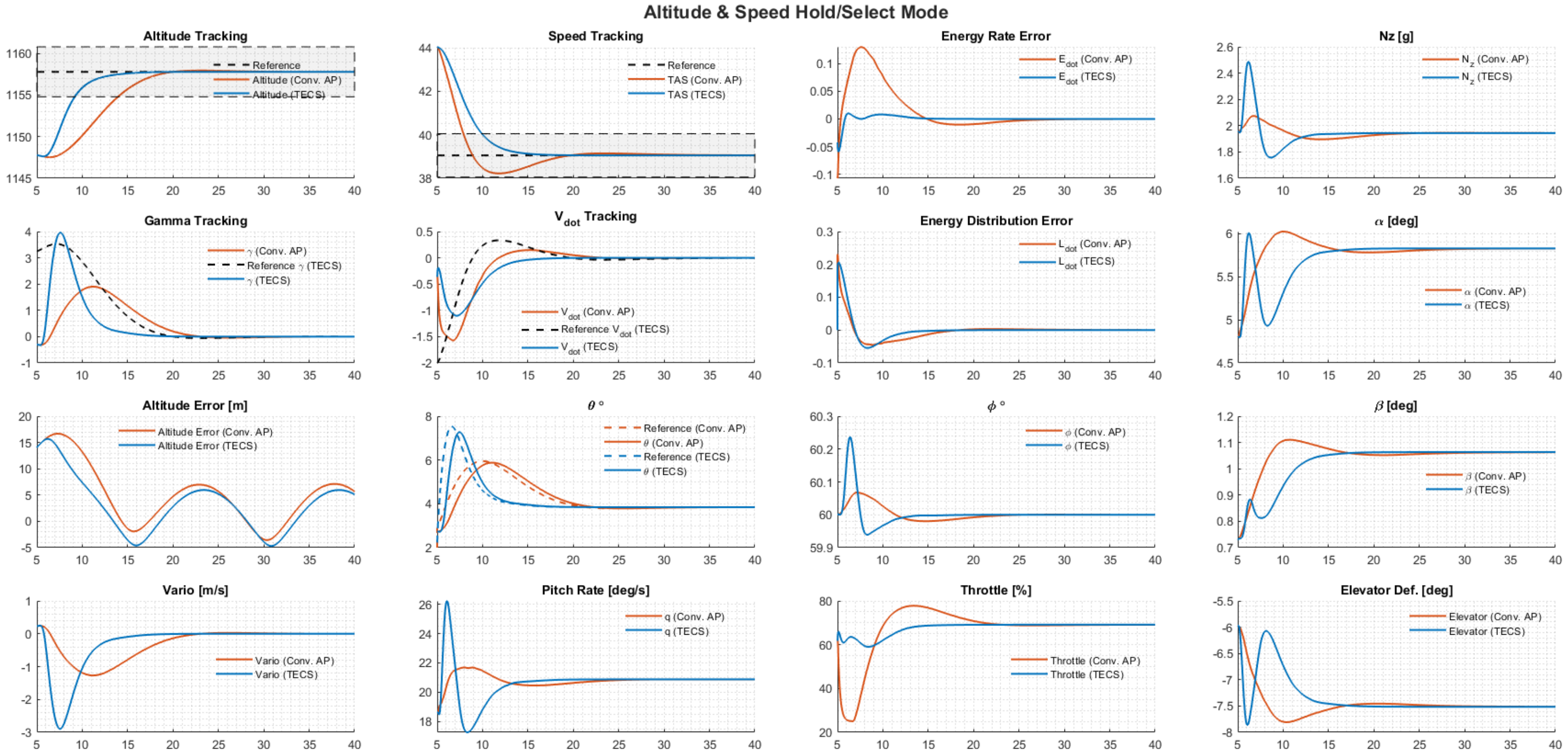

Figure 4.3 : 10m Climb & 5m/s Deceleration – 60 degree bank angle.

In the given test scenario in Figure 4.4 and Figure 4.5, DEHA’s navigation autopilots are used (Gökduman, E.Y., 2025). In this navigation mission, the aircraft makes turns in a square path while holding both altitude and speed. These turns are performed with a roll angle of 60 degrees. This test scenario is very convenient for comparing both architectures. During a turn maneuver, the aircraft loses altitude, and an energy change is caused and the autopilots try to compensate this altitude loss and energy change. Figure 4.4 and Figure 4.5 show that both architectures successfully completed the mission, although TECS performed significantly better. TECS results show that the loss of both altitude and speed during turns is considerably less compared to conventional autopilots.

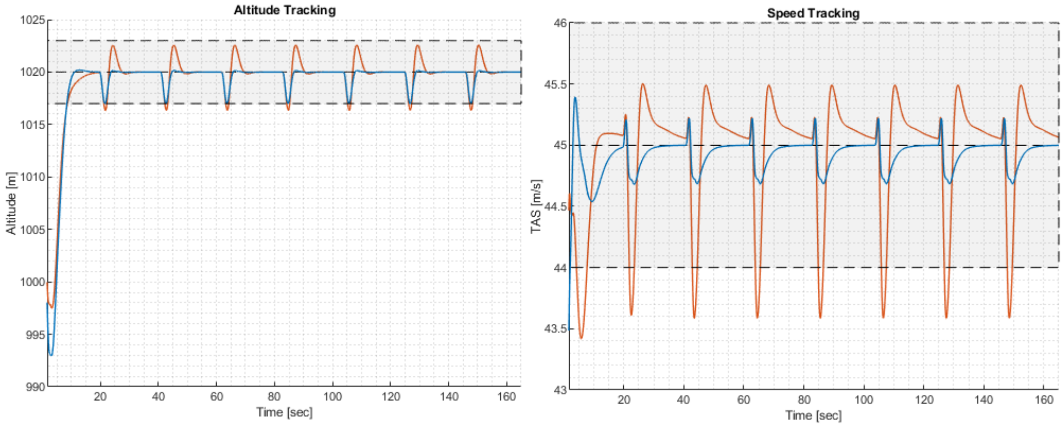

Figure 4.4 : Navigation Mission – Altitude & Speed Tracking.

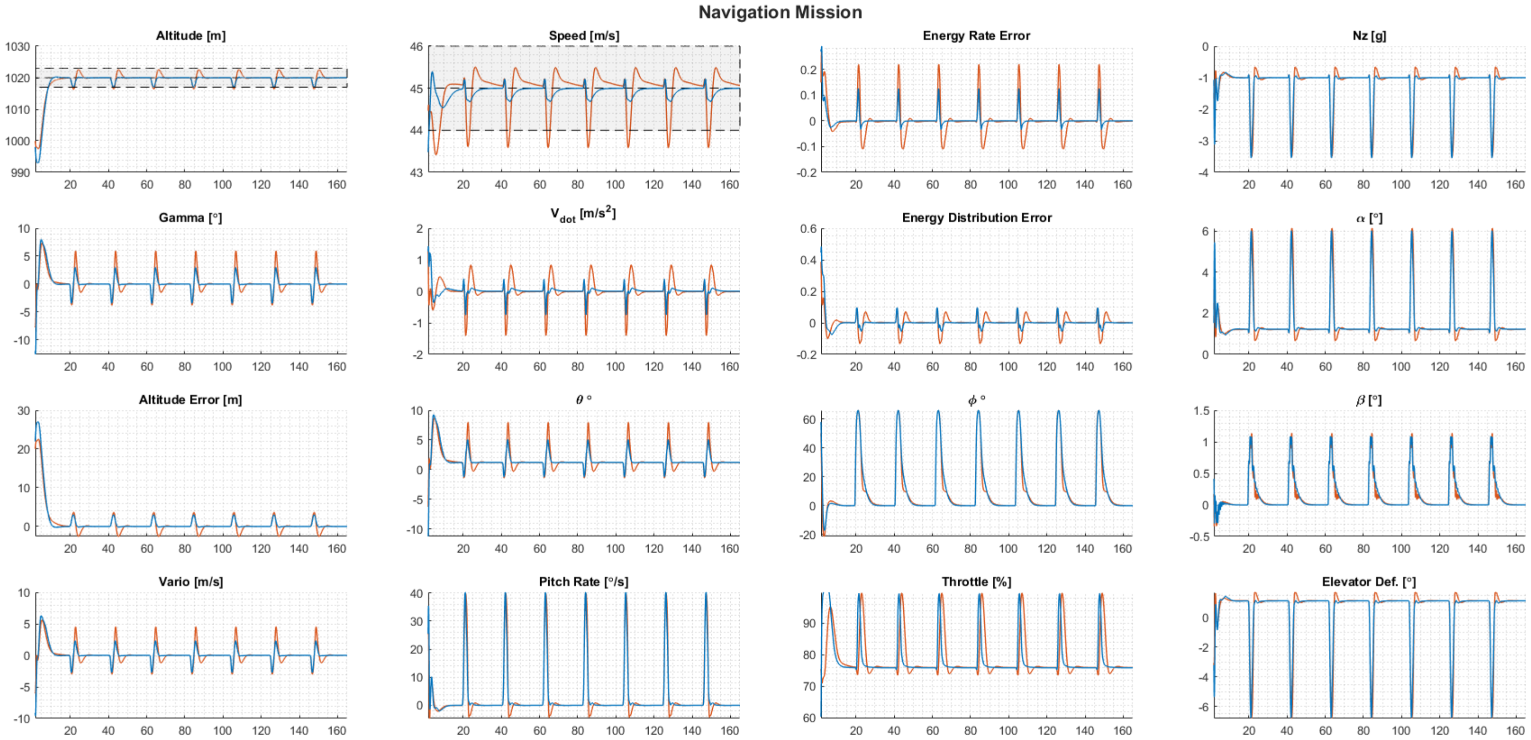

Figure 4.5 : Navigation Mission – Full States.

5. CONCLUSION

This thesis aims to conduct a comprehensive evaluation of the performance of longitudinal autopilot systems used in modern aircraft by comparing the performance of conventional control architectures and TECS (Total Energy Control System), an energy based approach. Throughout the study, the theoretical foundations of both control architectures are explained, and the behavior of these systems against flight scenarios is investigated and systematically compared on an experimental jet powered unmanned aerial vehicle model named DEHA (Deneysel Hava Aracı). The findings reveal the strengths and weaknesses of both structures and clearly show which system is more advantageous under different mission profiles and flight conditions.

Conventional autopilot systems have been used for a long time in aviation history; their structure is well understood and has been successfully implemented on many different platforms. These structures are usually SISO based and attempt to regulate quantities such as airspeed and altitude through independent control loops. For example, airspeed is controlled by the throttle, while altitude is controlled by the elevator movement. These approaches simplify controller design, increase modularity, and provide the user with clear engineering processes. As a matter of fact, these systems have found a wide range of applications, from commercial passenger aircraft to small UAVs, and have become standard in engineering applications.

In this thesis, conventional autopilot systems are modeled with three main components: pitch attitude autopilot, altitude autopilot, and speed autopilot. The pitch attitude controller holds only the pitch angle constant in the inner loop. The altitude autopilot operates with two sub loops: altitude and vertical speed control. This structure first converts the reference altitude command into a vertical speed command and then generates a pitch angle command. Speed autopilot, on the other hand, generates a throttle signal with a PI controller structure and tracks the true airspeed. These systems were tuned in MATLAB/Simulink environment by using a loop breaking technique to control their frequency responses and stability margins were ensured.

The main advantage of conventional architecture is its simple control structure and clear separation of functions. The individual design of each control loop makes it possible to adjust controller gains more easily. Especially in the simpler phases of flight – where constant speed and altitude must be maintained – the performance of conventional systems is quite good. In addition, these systems have been in use for many years and have established a high level of trust and user experience in the industry.

However, conventional architecture is not without its limitations. Due to the physical interconnectedness of states, such as airspeed and altitude in the longitudinal axis, controlling these states separately can lead to cross-interactions. Especially in maneuvers where energy transfer is intense – for example, deceleration during climb or acceleration during sudden descent – the throttle and elevator control signals may operate in opposite directions, and instabilities or deviations may occur in the system. This can increase the load on the control surfaces, reduce fuel efficiency, and jeopardize flight safety. The analysis conducted in this thesis shows that in such complex missions, the control errors of conventional autopilots increase, the time to reach target values increases, and the control commands fluctuate more.

TECS architecture offers an energy based control approach that aims to solve these problems. Based on the total energy of the aircraft (kinetic and potential), TECS regulates the energy rate with the throttle and how this energy is distributed between airspeed and altitude with the elevator. Thus, instead of controlling airspeed and altitude separately, the system creates a more consistent control strategy by linking these two states together. This structure is a MIMO system and provides a more integrated decision structure, unlike conventional autopilots.

In this thesis, TECS architecture is structured through an eight-step design process. Initially, the throttle and elevator loops were designed separately, and then both control loops were reorganized together by considering couple effects. At each step in this process, the open loop responses were analyzed using the loop breaking method, and the controller gains were optimized to meet the stability and performance criteria of the system. In addition, proportional control loops were added at the end of the system to achieve the reference altitude and airspeed values, completing the overall flight control architecture of TECS.

The biggest advantage of TECS emerges in complex maneuvers where energy transitions are intense. In cases such as climbing, descending, and loss of altitude during turns, TECS generates throttle and elevator control signals more evenly, reaching reference values in less time and with less deviation. In addition, throttle deflections are lower compared to the conventional system, which has a positive effect in terms of both engine performance and fuel consumption. Since the system operates on energy rate and distribution, it is able to react more robustly to environmental effects and exhibits more stable behavior, especially in turbulence or sudden altitude changes.

However, TECS also has some challenges. First, the controller architecture is more complex, increasing design and implementation time and making debugging difficult. The system is also highly dependent on accurate energy calculations and sensor data. Especially on platforms where sensor calibration is inadequate, the performance of TECS can be degraded. While conventional systems can operate with simple PID structures, the implementation of TECS requires more sophisticated architecture. Therefore, the integration of the system may demand more engineering resources and time.

The simulations conducted in this thesis extensively tested both control architectures under different mission scenarios. In simple missions (constant speed and altitude), both systems performed well, but in complex missions (speed transitions with altitude change, turns with high bank angle), the performance of TECS was clearly superior. In this context, it can be said that conventional autopilots are quite efficient within certain limits within the envelope, but in complex mission scenarios, the energy based decision mechanism offered by TECS becomes more advantageous.

In conclusion, this thesis has shown that a “one controller fits all” approach is not sufficient in the design of flight control systems; different architectures should be evaluated according to the mission, platform characteristics, and control expectations. While conventional autopilot systems offer significant advantages in terms of simplicity, robustness, and industrial maturity, energy based approaches such as TECS offer next generation solutions in terms of higher mission accuracy, more balanced control responses, and efficiency.

In this context, it is understood that both structures have their place in flight control systems, but it is vital to choose the right control architecture depending on the mission definition. Especially in complex, dynamic, and mission-critical scenarios, the integration of architectures such as TECS is recommended, whereas conventional autopilot structures are considered sufficient for simple or routine flights.

In the future, the hybrid architectures may be proposed. These architectures can operate with conventional autopilot under normal conditions while switching to TECS architecture during complex missions. Likewise, it is important for future research to make the TECS structure more flexible with machine learning based adaptive control approaches and to increase its robustness against sensor failures. In conclusion, this study has not only compared existing control architectures but also provided an important perspective on how future control systems may be shaped.

REFERENCES

Abrahart, R. J., and See, L. (1998). Neural Network vs. ARMA Modelling: Constructing Benchmark Case Studies of River Flow Prediction. In GeoComputation '98. *Proceedings of the Third International Conference on GeoComputation*, University of Bristol, United Kingdom, 17–19 September (CD-ROM).

Altay, H. (2021). Jet İnsansız Hava Aracı İçin Model Tabanlı Tasarım Yaklaşımı ile Benzetim Modelinin Oluşturulması ve Deneysel Doğrulanması (İstanbul: İstanbul Technical University, Lisansüstü Enstitüsü, Yüksek Lisans Tezi, 2021).

Anderson, J. D. (2002). Introduction to flight (5th ed.). McGraw-Hill.Bilstein, 2001.

Argyle, M. E., & Beard, R. W. (2016). Nonlinear Total Energy Control for the Longitudinal Dynamics of an Aircraft. ACC 2016.

Bauer, P. (2003). A Modified Total Energy Control Scheme for Unmanned Aircraft. Journal of Intelligent & Robotic Systems ISA.

Bry, S., Richter, C., Bachrach, A., & Roy, N. (2015). Aggressive flight of fixed-wing and quadrotor aircraft in dense indoor environments. The International Journal of Robotics Research, 34(7), 969–1002.

Chakraborty, D., Choudhury, D., Roy, D. & Kar, I. (2021). Total Energy Based Flight Control System Design for a Longitudinally-Projected Control UAV Model. AIAA Scitech 2021 Forum. https://doi.org/10.2514/6.2021-1899.

Chudy, M., & Rzucidlo, P. (2012). TECS/THCS Based Flight Control System for General Aviation Aircraft. Journal of Theoretical and Applied Mechanics, 50(1), 83–93.

Cook, M. V. (2007). *Flight Dynamics Principles: A Linear Systems Approach to Aircraft Stability and Control* (2nd ed.). Butterworth-Heinemann.

Cook, M. V. (2013). Flight Dynamics Principles: A Linear Systems Approach to Aircraft Stability and Control (3rd ed.). Butterworth-Heinemann.

Dorf, R. C., & Bishop, R. H. (2011). Modern Control Systems. Pearson.

Gökduman, E. Y., & Demirkiran, R. (2025). Waypoint Navigation Capability Achievement: L1 Guidance Approach Applied to Jet-Powered Scaled UAV, AIAA 2025-2085.c1. AIAA SCITECH 2025 Forum. January 2025.

FAA. (2020). Airplane Flying Handbook (FAA-H-8083-3C). U.S. Department of Transportation, Federal Aviation Administration.

Giusti, A., Degaspare, M., & Kienitz, K. H. (2020). A More Integrated Total Energy Control Law for Longitudinal Automatic Flight Control System. Aerospace Science and Technology, 104, 105962.

International Civil Aviation Organization. (1993). Manual of the ICAO standard atmosphere: Extended to 80 kilometres (262,500 feet) (3rd ed.). Montreal, Canada: ICAO.

Lambregts, A. A. (1993). Vertical Flight Path and Speed Control Autopilot Design Using Total Energy Principles. NASA-TM-83820.

Lambregts, A. A. (1987). Integrated System Design for Flight and Propulsion Control Using Total Energy Principles. NASA TM-89098.

Mahony, R., Kumar, V., & Corke, P. (2012). Multirotor aerial vehicles: Modeling, estimation, and control of quadrotor. IEEE Robotics & Automation Magazine, 19(3), 20–32.

McLean, D. (1990). Automatic Flight Control Systems. Prentice Hall.

Demirkiran, R., & Isci, H. (2023). Application of Robust Dynamic Inversion to Longitudinal Control of an Aircraft, AIAA 2023-1637. AIAA SCITECH 2023 Forum. January 2023.

Demirkiran, R., & Isci, H., & Turkoglu, E. (2024). Handling Quality-Oriented Tuning Procedure of a Dynamic Inversion Control Law via a Robust Control Technique, AIAA 2024-2563. AIAA SCITECH 2024 Forum. January 2024.

Roland, J. W. (1985). The Autopilot: A History of Aircraft Automatic Flight Control.

Skogestad, S., & Postlethwaite, I. (2007). Multivariable Feedback Control: Analysis and Design (2nd ed.). Wiley.

Stevens, B. L., & Lewis, F. L. (2003). Aircraft control and simulation (2nd ed.). Wiley.

Stevens, B. L., & Lewis, F. L., & Johnson, E. N. (2016). Aircraft control and simulation: Dynamics, controls design, and autonomous systems (3rd ed.). Wiley.

U.S. Department of Defense. (1980). Flight control systems—Design, installation, and test of piloted aircraft, general specification for (MIL-F-9490D). Washington, DC: Department of Defense.

Weiser, C., Looye, G., & Ossmann, D. (2024). light Testing Total Energy Control Autopilot Functionalities for High Altitude Aircraft. AIAA.

CURRICULUM VITAE

Name Surname : **Rumeysa Kökoğlu**

EDUCATION :

- **B.Sc.** **:** 2021, Istanbul Technical University, Faculty of Electric and Electronics, Control and Automation Engineering Department

PROFESSIONAL EXPERIENCE AND REWARDS:

- 2021-Present Design Engineer, Turkish Aerospave Indutry.

PUBLICATIONS, PRESENTATIONS AND PATENTS ON THE THESIS:

- **Kökoğlu, R.,** Güzelkaya M. 2025: Enhancing Aircraft Control Through Total Energy Control Systems: A Comperative Study with Conventional Autopilots, *4th International Graduate Symposium*, 2025 Istanbul, Turkey.