FAULT TOLERANT CONTROL OF A MISSILE AUTOPILOT SYSTEM

FÜZE OTOPİLOT SİSTEMİNİN ARIZA TOLERANSLI KONTROLÜ

YUSUF ÖNAY

PROF. DR. S. ÇAĞLAR BAŞLAMIŞLI
Supervisor

Submitted to
Graduate School of Science and Engineering of Hacettepe University
as a Partial Fulfillment to the Requirements
for the Award of the Degree of Master of Science
in Mechanical Engineering

June 2024

ABSTRACT

FAULT TOLERANT CONTROL OF A MISSILE AUTOPILOT SYSTEM

Yusuf Önay
Master of Science, Mechanical Engineering
Supervisor: Prof. Dr. Selahattin Çağlar Başlamışlı
June 2024, 84 pages

In aerospace defense industry, missile systems have been an important asset for more than half a century. As technological advancements take place, this field also continues to improve and develop today. A missile is simply a type of controlled aircraft with the purpose of carrying a payload (warhead) to a designated location. In order to complete this mission, a controller that can handle disturbances and uncertainties is necessary. The section responsible for the flight control of missiles is called the missile autopilot. The autopilot contains pre-designed controllers and other logical coding that concern any type of control of the missile. In this thesis, the main controllers of the autopilot are designed benefiting from model predictive control theory and then missile autopilot system is equipped with fault tolerance capability against actuator failure. Fault-tolerant control is a control approach that aims to recover controller performance in case of a fault occurring in the system. These faults can happen in the plant, sensors, or actuators. Fault-tolerant control takes place when the basic controller would fail in such a scenario. In missile systems, actuators are vital for the control of the missile. They rotate control surfaces of the missile so that the missile can carry out required maneuvers during flight. If one of the actuators beocme dysfunctional this can lead failure of the whole missile system. In order to prevent losing an entire expensive system due to actuator failure, an active fault tolerant control method is required. In this study, firstly the equations of motion for the system are obtained and then a linearization process is followed. Adaptive Model Predictive Control is applied to the linearized system model. The accuracy of linearization is shown by comparing time responses of linear and nonlinear system dynamics. Then a nonlinear analysis model is used to analyze the robustness and stability of the controller. Then the effect of a fault scenario is analyzed where one of the four actuators has failed and stuck at its latest position (also called total loss of effectiveness). Then a fault diagnosis logic is coded in order to locate and identify the faulty actuator. After the diagnosis, an appropriate fault-tolerant control process is applied to the remaining three actuators. Finally, nonlinear simulation results are obtained in order to compare the fault scenarios with and without fault-tolerant control and demonstrate the effectiveness of the method used.

Keywords: missile, autopilot, fault tolerant, model predictive, control

ÖZET

FÜZE OTOPİLOT SİSTEMİNİN ARIZA TOLERANSLI KONTROLÜ

Yusuf Önay
Yüksek Lisans, Makina Mühendisliği
Danışman: Prof. Dr. Selahattin Çağlar Başlamışlı
Haziran 2024, 84 sayfa

Havacılık savunma sanayii sektöründe füze sistemleri yarım yüzyılı aşkın süredir önemli bir araç olmuştur. Dünyada teknolojik gelişmeler oldukça bu sektör de aynı şekilde gelişmeye devam etmiştir ve günümüzde de devam etmektedir. En basit şekliyle füze bir yükü (harp başlığı) istenen noktaya götürmeye yarayan bir hava aracıdır. Bu görevi başarmak için belirsizlik ve bozucu etkilerin üstesinden gelecek bir kontrolcü elzemdir. Füzenin kontrolünden sorumlu bölüme ise füze otopilotu denilmektedir. Otopilot önceden tasarlanmış kontrolcü ve füzenin uçuş sırasındaki kontrolüyle ilgili tüm kodlamaların yer aldığı sistemdir. Bu tez çalışmasında, füze otopilot sisteminin ana kontrolcüleri model öngörülü kontrolden faydalınalarak tasarlanmış ve ardından otopilot sistemine eyleyici arızasına karşı tolerans yeteneği kazandırılmıştır. Arıza toleranslı kontrol, sistemde bir hata meydana gelmesi durumunda kontrolcü performansını geri kazanmak amacını güden bir kontrol yaklaşımıdır. Bu hatalar, ana sistemde, sensörlerde veya eyleyicilerde meydana gelebilir. Hata toleranslı kontrol normal bir kontrolcünün arıza durumunda başarısız olacağı noktada devreye girer. Füze sisteminde eyleyiciler kontrol için son derece önemli elemanlardır. Eyleyiciler füzenin havada gerekli manevraları yapabilmesi için kontrol yüzeylerini döndürmeye yarar. Eyleyicilerden bir tanesinin arızalanması tüm sistemin başarısız olmasına yol açabilir. Böyle bir eyleyici hatası sebebiyle tüm sistemi kaybetmemek için aktif bir arıza tolerans metoduna ihtiyaç vardır. İlk olarak, sistemin hareket denklemleri elde edilmiş ve doğrusallaştırma süreci takip edilmiştir. Adaptif model öngörülü kontrolcü doğrusallaştırılmış sisteme uygulanmıştır. Doğrusallaştırmanın yeterliliği açık döngü cevaplarının doğrusal olmayan sistem dinamiği sonuçlarıyla karşılaştırması ile yapılmıştır. Ardından doğrusal olmayan bir analiz modeliyle kontrolcünün gürbüzlüğü ve kararlılığı incelenmiştir. Sonra, eyleyicilerden birinin arızalanıp yerinde takılı kaldığı hata senaryosunun (etkinliğin tamamen kaybı) sisteme etkisi incelenmiştir. Ardından, arızanın varlığını ve yerini belirlemek için arıza teşhis algoritması kodlanmıştır. Teşhis ardından arızaya uygun olan arıza tolerans kontrol metodu kalan eyleyiceler için aktive edilmiştir. Son olarak, doğrusal olmayan simülasyon ortamında arıza tolerans kontrolünün varlığının hata senaryosuna nasıl etki ettiği sonuçlarla gösterilmiştir.

Keywords: füze, kontrolcü, otopilot, arıza tolerans, model öngürülü kontrol

ACKNOWLEDGEMENTS

I would like to express my gratitude to my supervisor Prof. Dr. S. Çağlar BAŞLAMIŞLI for his valuable support throughout the study.

I dedicate this thesis to my dear family...

CONTENTS

Page

TABLES

FIGURES

ABBREVIATIONS

MPC	:	**M**odel **P**redictive **C**ontrol
FTC	:	**F**ault **T**olerant **C**ontrol
CAS	:	**C**ontrol **A**ctuation **S**ystem
IMU	:	**I**nertial **M**easurement **U**nit
DoF	:	**D**egree **o**f **F**reedom
PLOE	:	**P**artial **L**oss **O**f **E**ffectiveness
TLOE	:	**T**otal **L**oss **O**f **E**ffectiveness
CFD	:	**C**omputational **F**luid **D**ynamics
C.G.	:	**C**enter of **G**ravity
ERA	:	**E**levator **R**udder **A**ileron
STT	:	**S**kid **T**o **T**urn
BTT	:	**B**ank **T**o **T**urn
RA	:	**R**olling **A**irframe
IOL	:	**I**nput **O**utput **L**inearization
UDE	:	**U**ncertainty **D**isturbance **E**stimator

1. INTRODUCTION

In this thesis work, an autopilot for a short-range air defense missile will be designed with fault-tolerant control. Short-range air defense missiles contain subsystems such as the seeker head, guidance unit, fuse, warhead, rocket engine, and control unit. Subsystems that are directly involved in this study are autopilot, inertial measurement unit and control actuation. Short-range air defense missiles are systems that are required to maneuver during rapid changes of altitude, velocity, and angle of attack. Hence, these conditions must be considered during the autopilot design process. In the autopilot design process, in order to apply linear control methods, the system must be linearized at various altitudes, velocities, masses, and inertia values. In these linearization points, a state-space model is constructed so that a state feedback structure can be implemented. Another important subject for the design process is the missile maneuvering configuration. Three main classifications for missile maneuver configurations are skid-to-turn (STT), bank-to-turn (BTT), and rolling airframe (RA) [12]. For this thesis, skid-to-turn configuration will be considered. In skid-to-turn configurations, the angular rate of the missile around its longitudinal axis is stabilized at zero, and the missile can maneuver simultaneously around its lateral axes[7]. Therefore, the design of the roll controller of the missile aims to keep the roll rate at zero. This way, lateral dynamics can be decoupled and linearized with the assumption of zero roll rate so that a controller design can be made for them. The controller design should be made by taking actuator dynamics, sensor delays, and aerodynamic uncertainties into account. The main controller design will be based on Model Predictive Control (MPC) theory. 'Adaptive MPC Controller' from MPC toolbox of MATLAB/Simulink will be used. The linearized system model will be updated each sample time for the Adaptive MPC Controller. This design so far assumes that all actuators are operating as expected. Therefore, in addition, a fault-tolerant control method will be implemented considering the possibility of an actuator failure. In the control actuation system, there are actuators for rotating control surfaces of the missiles in order to generate required aerodynamic forces and moments. Missiles with four control surfaces usually have plus or cross configurations for the surface placements. The needed elevator, rudder, and aileron commands are distributed to the actuators of these control surfaces according to this configuration [7]. In this thesis, the control surface configuration will be a cross configuration. Structural weakening that may happen during time in storage or high hinge moments that can occur during the flight can cause one of the transmission elements in the actuation system to fail and get stuck at a certain angular position. For the fault-tolerant control application, a fault diagnosis algorithm and an appropriate controller design are needed. The Fault diagnosis algorithm will be used for detection and isolation of the fault. Fault-tolerant control methods can be categorized as passive and active methods [4]. For a missile with high velocity and low inertia, an actuator failure will have a great effect on system dynamics [1], therefore, an active fault-tolerant control method will be applied. Active fault-tolerant control method will detect which actuator has failed, and a proper reconfiguration will be made for the remaining three actuators. The basic controller structure will be merged with the fault-tolerant approach. Fault-tolerant control system will be tested under nonlinear simulation scenarios with various sensor delays and aerodynamic uncertainties. This way, the final robustness of the system against actuator faults will be analyzed, and analysis results will be demonstrated.

1.1. Scope Of The Thesis

This thesis mainly focuses on fault-tolerant control design of a missile autopilot system.

The main motive of this thesis is to implement fault tolerance ability to the missile autopilot system. For this purpose, a basic controller design and fault tolerant controller design will be merged together. For the faultless case, controller designs will be made for roll, pitch, and yaw channels. Roll autopilot will stabilize the roll rate of the missile at zero rate. Pitch and yaw autopilot will be designed to track required acceleration commands along lateral axes. For the autopilot design, linearizations will be made at various altitudes, velocities, inertia, and masses. After the design, actuator fault scenarios will be modeled and a fault diagnosis algorithm will be coded. The Fault diagnosis algorithm will inspect commands and responses of each actuator and detect the faulty actuator. When the fault is detected, a fault tolerant control method will be activated. The method used in this thesis work will be the control mixer reconfiguration method. This method will activate a new suitable reconfiguration for the remaining operational actuators.

For missiles with skid-to-turn configurations, the roll channel and pitch-yaw channels are regarded independently. Roll autopilot stabilizes the roll angular rate of the missile. This way, pitch and yaw autopilots can be designed by a linearization with zero roll rate assumption [11,12,13,14]. Due to the symmetrical shape of the missile, cross products can be assumed to be zero and inertia terms around lateral axes are equal to each other. Lastly, when a small angle of attack assumption is made, the linearization process can be complete. After the linearization, state-space representations for each channel will be made and an adaptive model predictive controller will be implemented using MATLAB/Simulink software toolbox. States for the state-space model will be chosen as lateral acceleration and angular rates measured by IMU sensor and control surface deflection angles received by actuator encoder readings.

For the autopilot design, the controllers are based on Model Predictive Control (MPC) theory. Model Predictive Control is a type of modern optimal control method. A prediction horizon is defined for MPC where it will minimize a cost function throughout that optimization window using the given plant model. The optimization results with a set of control inputs that minimize the tracking error. A control horizon for the control input length can also be defined. After the optimization is over, only the first element of the optimal control actions is used. In the next time step, the same procedure is repeated with more recent measurements [20]. One of the most popular features of MPC is that input or output constraints can be defined for the optimization process. This way, MPC is highly suitable for optimizing the performance of constrained systems [22].

In missiles with a plus or cross configuration of control surfaces, flight control commands must be distributed accordingly to the actuators. For the cross configuration, every actuator is used for elevator, rudder, and aileron deflections. For example, when elevator deflection is needed, every actuator is used so that vertical forces add up to each other and horizontal forces cancel each other [7]. Long periods of time in storage or high hinge moments during flight may cause structural deterioration that leads to actuator failures [1,5]. For missiles with high velocity and low inertia values, an actuator failure can have a significant impact on system performance and even cause instability. Because of this, an active fault-tolerant control should be considered [1,2,4,6]. When one of the actuators has failed, it will act as a disturbance on the system. The nominal controller's ability to reject this disturbance will also be weakened because given commands will not generate expected control surface deflections. The transformation process between autopilot commands and actuator commands (ERA to 1234) is also called 'Control Mixer'. Therefore, a control mixer reconfiguration method will be applied to utilize the remaining three actuators appropriately and recover control performance of the system. After the actuator fault is modeled, a fault diagnosis algorithm and active fault-tolerant control method with reconfiguration approach will be applied [1,2,5]. Lastly, the obtained fault-tolerant autopilot performance will be tested under various fault scenarios, aerodynamic errors, and sensor delays.

1.2. Contributions

In this research, we cover the design of a missile autopilot system that benefits from fault tolerant and model predictive control methods by proposing a novel, simple and efficient approach. The main contributions of this paper can be summarized as follows:

- We propose a fault tolerant ability implementation to missile autopilot system while benefiting from model predictive control advantages
- Unlike most of the previous works, we use MPC and FTC together for the design of a missile autopilot system
- Our simulation results show the effectiveness of the proposed approach by demonstrating results from various fault scenarios under errors and uncertainties.

Fault tolerant control research in missile autopilot design is not very common in literature [1]. When compared to existing works in literature our thesis stands out exceptional in some aspects. Our study combines model predictive control theory and fault tolerant control theory for missile autopilot system design. Apart from that, most similar papers presents design of controllers for a specific flight condition however our thesis proposes single adaptive MPC controller design that covers a wide flight envelope by itself without the need of methods such as gain scheduling. Hence, this thesis study contributes to literature by proposing an autopilot design for multiple flight conditions by MPC theory and also FTC to handle actuator failures.

1.3. Organization

The organization of the thesis is as follows:

- Chapter 1 presents our motivation, contributions and the scope of the thesis.
- Chapter 2 provides background overview on missile autopilot systems, fault tolerant control, model predictive control and missile kinematics
- Chapter 3 gives insight on some related works, their similarities and differences compared to our thesis
- Chapter 4 presents methods regarding the thesis subject while explaining the preparation work for simulation obtaining results
- Chapter 5 shows simulation results obtained by linear and nonlinear models, nonlinear analysis under uncertainties, sensor delays and various fault scenarios
- Chapter 6 states the summary of the thesis and possible future directions.

2. BACKGROUND OVERVIEW

2.1. Missile Autopilot

Missiles are a type of aircraft that fly with the objective of carrying a warhead to a specific destination. In order to accomplish this goal, they require various subsystems. These subsystems commonly include a rocket engine, seeker head, guidance unit, and control actuators. Target information obtained by the seeker head is transferred to the guidance unit, which generates the control surface deflection commands necessary for the needed maneuver for intercepting the target, all while the rocket engine provides the required kinetic energy. Required maneuvers can be done by tracking angle of attack, angular rate, or acceleration commands [3,9]. The modern guidance units contain an inertial measurement unit (IMU) and onboard computer. The onboard computer carries out multiple online tasks during flight. These include navigation (using IMU sensor data) and generating guidance commands by using target and self-navigation information. The controllers in the autopilot system compute required aero fin deflections in order to make the required guidance maneuver. Finally, the control actuation system (CAS) ensures the aero fins are tracking the commands received from the autopilot system [7].

This thesis mainly focuses on the interaction between the autopilot, control actuation systems, and their effects on the performance of the missile system.

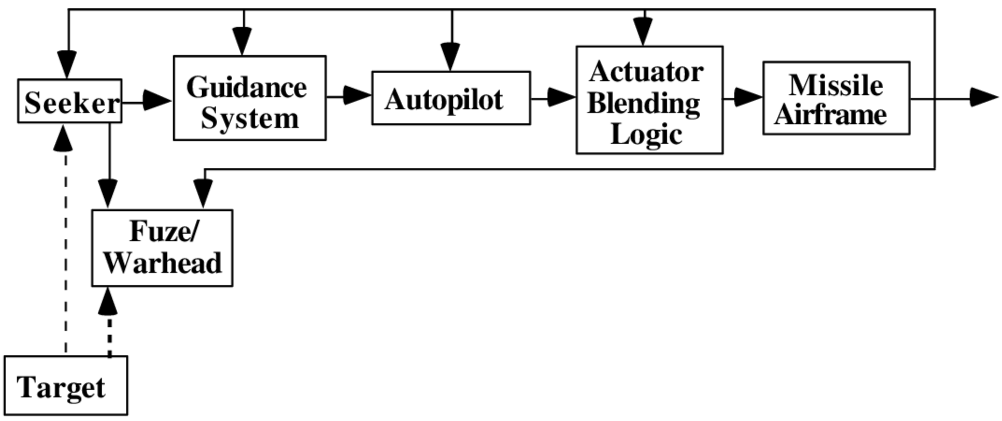


Figure 2.1 Missile Guidance Loop [25]

Menon and Ohlmeyer [25] shows complete guidance loop of the missile in figure 2.1. This figure shows where guidance and autopilot systems take place in the loop. This study will focus on the autopilot system which converts guidance commands into actuator commands.

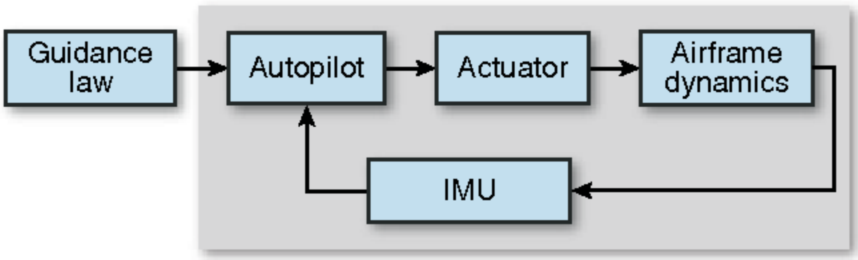


Figure 2.2 Missile Control Loop [26]

The figure 2.2 by Naidu et al [26] shows the autopilot loop in more detail which can be observed separately from guidance loop. The input of this system is guidance commands obtained from the guidance law. This command can be angle of attack or angular rate [3,9] or acceleration command [10]. The aim of the autopilot design will be to generate actuator commands that will ensure stable and robust tracking of guidance commands.

2.2. Control Actuator System

Control actuators are systems that aim to rotate control surfaces of the aircraft. There are many types of actuators such as hydraulic, electric, linear, mechanical or electromechanical. Most missiles use electromechanical actuators [7], and the actuator in this study is an electromechanical actuator too. Moment produced by an electric motor is converted into control surface mechanical rotation. The input of the system is the desired control surface/fin angle. After receiving the angle command input, the controller of the system calculates an error by feeding back the angle reading from the encoder device. Then, the error is converted into a current command for the electric motor such that the desired angular position is reached at the end of the process.

In this missile system, there are four control fins and four independent CAS, each controlling one fin. The fins are placed around the missile symmetrically. There are two possible configurations for this setup: the ’plus’ (+) and ’cross’ (x) configurations. These configurations change how elevator, rudder, and aileron deflections must be distributed into four aero fins. Each configuration can have its own advantages and disadvantages; however, such a topic is out of scope for the purpose of this thesis. In this thesis, the system has a cross (x) surface configuration. Figure 2.3 shows the relation between control surfaces and aerodynamic deflections. An actuator failure is going to have a great impact on the system by degrading control performance and even posing instability risks [1].

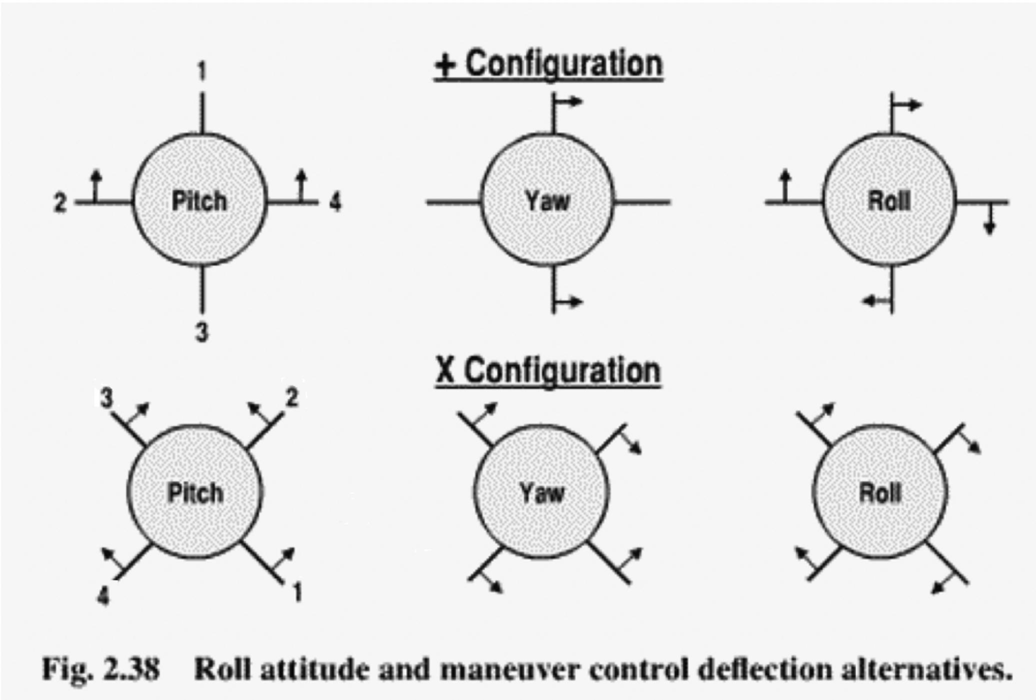


Figure 2.3 Actuator Configurations [7]

This relation between aero fins and control actuators is very vital for the purpose of the thesis because in case of an actuator failure, this distribution is going to have vital importance for the active fault-tolerant control method [6].

Control systems are classified based on the number of actuators operating in the systems. This classification is done by comparing the degrees of freedom and the number of actuators. For the missile system, there are 3 channels that are desired to be controlled using aero fins. These are pitch, yaw, and roll motions. If the system has more than 3 actuators, it is called an over-actuated system. If the number of actuators is equal to 3, it is a properly-actuated system. Lastly, if there are fewer than 3 actuators, the system is under-actuated [1].

Over-actuated systems have more actuators than needed. This increases the reliability and the flexibility of the system because even in case of actuator failure, the remaining actuators can be utilized to preserve the nominal performance of the system by applying Fault-Tolerant Control method [6].

Properly-actuated systems have the minimum number of actuators required to achieve full movement in all desired degrees of freedom.

Under-actuated systems cannot achieve full simultaneous control in some degrees of freedom. For example, if the system has only two actuators, it cannot achieve roll, pitch, and yaw control simultaneously. Therefore, at least one of these axes should be given up [7].

2.3. Fault Tolerant Control

In literature and practice, control systems are usually designed for systems operating under expected conditions. However in real-world applications, various subsystems or parts of the system can have deterioration or even complete failure. It would be highly desirable if the designed controller were able to adapt or reconfigure itself in such occasions and preserve same performance or at least an adequate portion of performance. This is the main goal of Fault Tolerant Control. FTC can be divided into two parts: Active and Passive FTC [4,5].

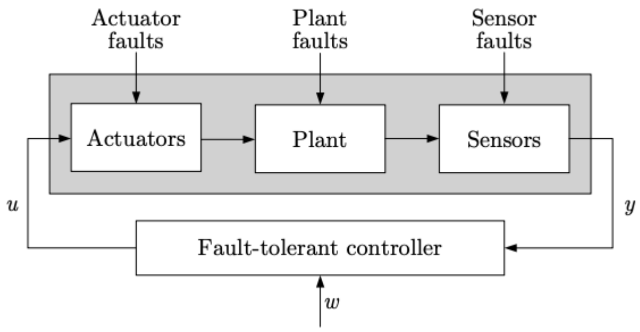


Fig. 7.5. Fault-tolerant controller

Figure 2.4 Fault-tolerant controller[4]

2.3.1. Passive Fault Tolerant Control

In this type, the robustness of the controller for the nominal system is high enough that it is able to tolerate the disturbances caused by system faults. Even though this scenario is possible, in many applications, the disturbance caused by the fault will likely exceed the robustness capability of the nominal controller.

This is also the case in the missile system in this thesis. The missile has relatively low mass and inertia while flying at supersonic speeds. Therefore, it has a fast dynamic response to control inputs or disturbances. A total loss of effectiveness for an actuator will bring significant impact on system dynamics, which the robustness of the nominal controller is not able to overcome in certain cases (shown in 6.14). For this reason, an active fault tolerant method will be applied.

2.3.2. Active Fault Tolerant Control

This type indicates that there is going to be an online action taken in the control system upon the encounter of a fault during operation.

Adaptive: In this method, an adaptive controller is designed such that it adapts to the new dynamics of the faulty system. For example, if a fault in the plant has occurred, the plant dynamics have now changed. Given enough time, the adaptive controller will adjust itself according to the new system dynamics.

Controller Redesign: In this approach, there is a pre-designed alternate action for the controller upon diagnosis of a fault. After determining the effect of a fault on the system, a recovery action for the controller is designed. This new design may be a completely new controller which is switched to after fault diagnosis (Control Reconfiguration) or it may be a local change in the controller that preserves the input-output structure of the basic controller (Fault Accommodation).

Fault tolerant control design consists of two main steps:

- Fault Diagnosis
- Control Re-design

2.3.3. Fault Diagnosis

A fault diagnosis process is necessary for applying a fault tolerant recovery action. This is because, to initiate a fault tolerant method, one must ascertain the presence, location, and type of the fault that has occurred. Fault diagnosis comprises three sub-processes within itself. These are:

- Fault Detection: Fault Tolerant Control system should have a detection algorithm which is responsible of determining the existence of a fault.
- Fault Isolation: In this step, the location of the fault is determined. If the system is shown as a control cycle that consists of transfer functions representing each element, goal of this step is to find which element in the feedback cycle is faulty. This information is important because this way the effect of fault on the system dynamics can be modeled depending on the type of fault which is diagnosed in next step.
- Fault Identification: If there are multiple possibilities for different types of fault, identifying which type has occurred is the final and most important step for a fault diagnosis algorithm. Sometimes the fault isolation step can also be included in identification process as well. After this step is complete, existence, type and location of the fault is determined. This means that the effect of the fault on system dynamics now can be modeled and analysed.

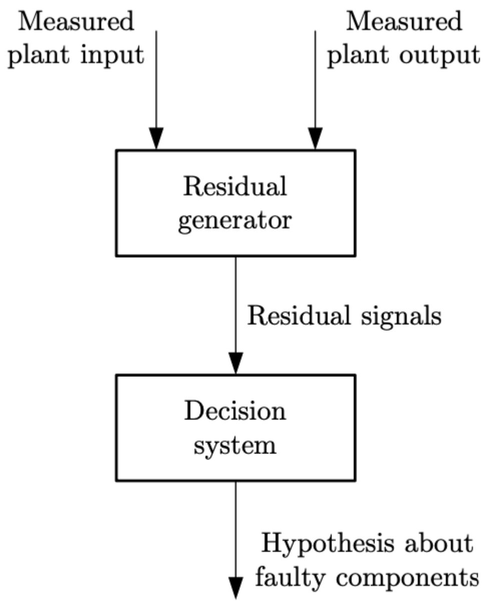


Fig. 6.1. Structure of a fault diagnosis system

Figure 2.5 Structure of a fault diagnosis system[4]

2.3.4. Control Re-design

Since the effect of the fault on the system is now known, an appropriate control recovery action can be applied. This process is called controller re-design. There are multiple possible approaches and methods. These methods can be categorized as followed:

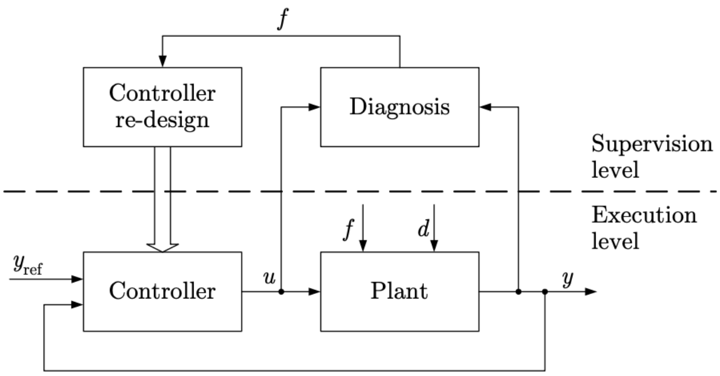


Fig. 1.6. Architecture of fault-tolerant control

Figure 2.6 Architecture of fault-tolerant control[7]

Blanke [7] demonstrates the general architecture of a fault tolerant control system as shown in Figure 2.6. The diagnosis algorithm constantly checks necessary inputs and outputs of the plant. Once a fault is identified it activates corresponding control action which is simply called 'Controller Re-design'. Since passive FTC is already covered, Controller re-design section is related to active FTC methods. These methods can be explained in 2 categories:

- Fault Accomodation: When the scale of change is limited in the controller such as parameter or structure change in the controller and input output structure between the plant and the controller is unchanged.
- Control Reconfiguraiton: The input and output structure between plant and the controller is changed. Original controller objective is achieved however a reduction in the control performance may be observed [2].

In the scope of this thesis, a single actuator fault case will be inspected. The fault scenario is failure of one of the four actuators such that the control surface is stuck and can not move as a result. Therefore the over-actuated system becomes properly-actuated. The control mixer reconfiguration FTC method will be applied in order to preserve desired system performance by utilizing the remaining three actuators while rejecting the disturbance caused by stuck control surface.

2.4. Model Predictive Control

Model Predictive Control (MPC) theory is an optimal control technique. This approach computes control actions over a finite receding horizon for a given plant and generates optimal control input which minimizes a cost function. It can also be described as a set of advanced control methods that benefits from future prediction of system by using a given system model. It solves a constraint optimization problem over a prediction horizon and determines an optimal output [24].

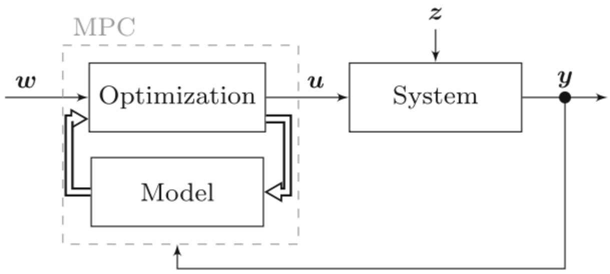


Figure 2.7 Simplified block diagram of a MPC-based control loop [24]

Main characteristics of MPC can be listed as:

- an explicit process model
- a receding horizon
- input and output constraints
- an iterative determination of controls

Being introduced at 1970s , since then there have been multiple branches of Model Predictive Control. In past years Finite Impulse Response (FIR) models were used for MPC, however in recent years state-space models have become much more popular [20]. The MPC plant model in this thesis will also be a state-space model. A brief explanation of widely used MPC types will be shown before presenting the type used in this study.

MPC can be divided into two main categories:

- Linear MPC
- Nonlinear MPC

Linear MPC runs optimization by using a linearized system model while a nonlinear model can be provided to Nonlinear MPC. The advantage of Nonlinear MPC is that it can make more precise predictions and more accurate optimization since the nonlinear dynamics of the system is not neglected. However it requires much greater computing power to operate at same period of time which could not be provided in many applications. The Adaptive MPC controller used in this research uses linear model. Linear Model Predictive Control have multiple different approaches as well [24]:

- Implicit MPC : control problem is solved online
- Explicit MPC : control problem is solved priori for all cases
- Multiple MPC : model is switched online among a set of predefined models
- Multiple Explicit MPC : control problem for all models are solved priori
- Adaptive MPC : The linear model for optimization is updated online at every sample time

For implementation of adaptive MPC the 'Adaptive MPC Controller' from MPC toolbox of MATLAB/Simulink software is used. The block runs basic linear MPC optimization in discrete-time domain. The discrete linear state-space model is updated at every time step and fed to the block for online optimization. The optimization process of MPC will be explained briefly in Methods section.

2.5. Reference Frame Notations

The missile system covered in this thesis is controlled in roll, pitch and yaw channels by means of aerodynamic force/moments generated by control surfaces. The rocket engine burns and produces a thrust according to a previously designed profile. The resulting thrust provides the missile with high velocity and dynamic pressure therefore high aerodynamic moments can be produced in order to obtain an angle of attack. Flying at high velocity with angle of attack results in high lateral acceleration capacity which provides necessary sudden maneuvers required to intercept the target [15].

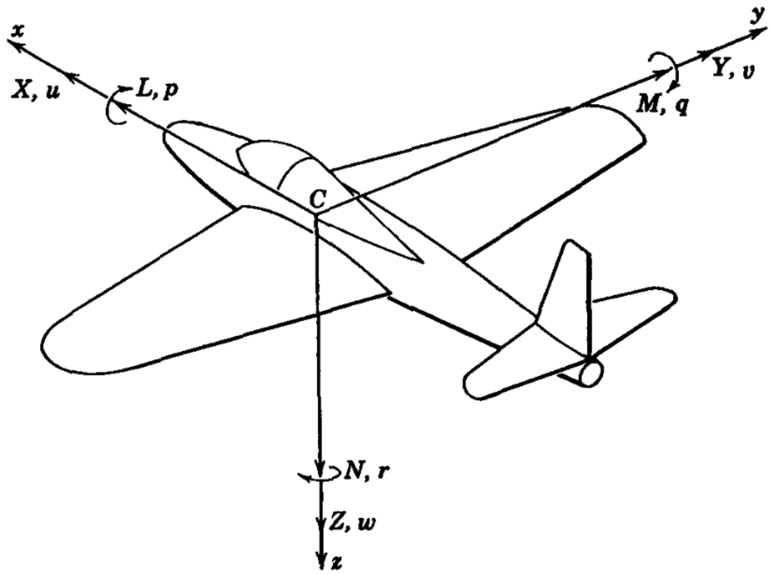


Figure 2.8 Notation for Body Axes [21]

Table 2.1 Table for Notations [9]

Name	Roll Axis	Pitch Axis	Yaw Axis
Angular Rate	p	q	r
Translational Velocity	u	v	w
Force along axis	X	Y	Z
Moment along axis	L	M	N
Moments of Inertia	I_{xx}	I_{yy}	I_{zz}

System dynamics and motion equations will be derived according to notations of this coordinate frame which is provided by Etkin and Reid [21]. In this frame, x axis is parallel to the longitudinal axis of the vehicle pointing the same direction as the nose of the missile. Aerodynamic force along this axis is notated as X and aerodynamic moment is L. Body angular and translational velocity components are written as u and p in order. Rotation around this axis is called roll motion. Y axis points rightwards and orthogonal to the X axis. Along this axis, aerodynamic force is Y, aerodynamic moment notation is M, body velocity component is v and angular velocity component is q. Rotation around this axis is called pitch motion. Lastly, z is the remaining axis that shows downward according to right hand rule. Aerodynamic force along this axis is Z force, moment is N, body translational velocity component is w, angular velocity component is r. Rotation around Z axis is called yaw motion.

This coordinate frame is carried and rotated along with the vehicle movement and rotation. In another words, it is fixed to the vehicle body, it has no relative motion with respect to the body.

3. RELATED WORK

Implementing a fault tolerant control approach in a missile system is not very common in literature. A few papers that have researched this subject will be briefly reviewed here.

Tong and Zhenyu [1] present a study regarding reconfigurable fault tolerant control for supersonic missiles with actuator faults. The paper focuses on two types of actuator faults called Partial Loss of Efficiency (PLoE) and Total Loss of Efficiency (TLoE). In the PLoE fault, the actuator responds to given commands with lower efficiency, tracking given commands with higher errors than expected. TLoE type of fault means the actuator has lost its efficiency completely and is not responding to given commands. Also, two types of faulty systems are introduced: the Properly-actuated system if only one actuator has lost efficiency, and the Under-actuated system if two actuators have failed. The system model is constructed with direct four actuator inputs. For the properly-actuated case, back-stepping control method is applied, and for the under-actuated case, a new approach called 'shape variables' is introduced. In this thesis study, the actuator angles are not directly related to the plant dynamics because aerodynamic analysis with inputs defined as elevator, rudder, and aileron deflection. There is another step in the autopilot which transforms these deflections according to four actuator locations (ERA to 1234 transformation). Also, instead of back-stepping and 'shape variables' method, we have used Model Predictive Control with an active Fault-Tolerant Control method (Command Reconfiguration [6], also called Control

Mixer Reconfiguration [28]). The paper similarly presents simulation results with various errors under actuator fault scenarios.

Another paper worth mentioning is by Louhuan et al. [6]. This paper proposes a deep learning approach to active fault-tolerant control for missile actuators. Similar to this thesis study, they have implemented the control mixer configuration method by reconfiguring flight control command distributions to the remaining fork (cross configuration) actuators. Unlike our thesis study, they mainly focus on the effectiveness of deep learning for identifying various types of actuator faults rather than the controller design of the autopilot. Deep learning training is carried out using received sensor data so that the trained model can correctly identify a fault type and switch to the appropriate control configuration. Whereas in our study, we have focused on the autopilot design of the missile by making it fault-tolerant to a specific common actuator failure scenario.

Lastly, the master's thesis study by Kırımlıoğlu [19] regards fault-tolerant control of a missile system with actuator failures. Kırımlıoğlu models actuator failure as an acceleration disturbance to the missile system. The main goal of the study is to achieve the desired impact angle with the target object by rejecting the disturbance of the failed actuator in the guidance loop. They have focused on the guidance loop of the missile system and studied target engagement scenarios, whereas in our work, we have focused on autopilot control and modeled actuator faults by modeling the transformation of ERA to each actuator. Kırımlıoğlu implements an adaptive sliding mode guidance law in order to reject the effect of acceleration disturbance on the guidance loop caused by actuator fault. In our research, we have used the control mixer reconfiguration method and updated Model Predictive Control constraints to preserve controller performance in case of an actuator failure.

4. PROPOSED METHOD

4.1. Equations of Motion

In this section, the nonlinear equations of motion governing the system will be presented.

Using Newtonian approach and Euler's equations for rigid body motion, the nonlinear relations between forces and moments in inertial frame are represented in terms of translational and angular velocities in body frame as shown below (ignoring change in mass). These are the nonlinear 6-DoF equations of motion of the missile system:

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = m\dot{V} + \omega \times V \tag{1}$$

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = m \begin{bmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{bmatrix} + m \begin{bmatrix} p \\ q \\ r \end{bmatrix} \times \begin{bmatrix} u \\ v \\ w \end{bmatrix} \tag{2}$$

$$\begin{bmatrix} L \\ M \\ N \end{bmatrix} = I\dot{\omega} + \omega \times (I\omega) \tag{3}$$

$$\begin{bmatrix} L \\ M \\ N \end{bmatrix} = \begin{bmatrix} I_{xx} & I_{xy} & I_{xz} \\ I_{xy} & I_{yy} & I_{yz} \\ I_{xz} & I_{yz} & I_{zz} \end{bmatrix} \begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} + \begin{bmatrix} p \\ q \\ r \end{bmatrix} \times \begin{bmatrix} I_{xx} & I_{xy} & I_{xz} \\ I_{xy} & I_{yy} & I_{yz} \\ I_{xz} & I_{yz} & I_{zz} \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix} \tag{4}$$

$$\begin{aligned} u &= V cos\alpha cos\beta \\ v &= V sin\beta \\ w &= V cos\beta sin\alpha \end{aligned} \tag{5}$$

These equations will be the basis for constructing the state-space model after linearization so that a controller can be designed afterwards. The designed controller (autopilot) will measure system states using IMU sensor data. The autopilot will receive command from guidance system (which is out of scope of this thesis) and generate commands for control surface deflections in order to track guidance command inputs.

Now, for the purpose of linearization, two main steps will be taken. Firstly, the equations (2) and (4) will be simplified by certain assumptions. Tong et al. [1] propose a simplified set of equations due to the symmetrical configuration of the missile geometry. Ryu et al. [3] propose assumptions such as rigid body, no gravity, no roll rate, and zero roll angles for the purpose of the simplification process. Urban [9] also uses Euler's equations for modeling the 6-DoF system and lists assumptions such as symmetric mass and neglection of gravity for reducing the Euler's equations. The assumptions made in this study are similar:

- Symmetrical airframe: $I_{yy} = I_{zz}$
- Cross products of inertia are assumed to be zero: $I_{xy} = I_{yz} = Ixz = 0$
- Roll rate (p) is zero
- Gravitational effects are neglected
- Small angle assumption for angle of attack ($sin\alpha \approx \alpha, \tan\alpha \approx \alpha$)

Now the remaining equations are:

$$X = m(\dot{u} + qw - rv) \tag{6}$$

$$Y/m = a_y = \dot{v} + ru \tag{7}$$

$$Z/m = a_z = \dot{w} - qu \tag{8}$$

$$L = I_{xx}\dot{p} \tag{9}$$

$$M = I_{yy}\dot{q} \tag{10}$$

$$N = I_{zz}\dot{r} \tag{11}$$

Aerodynamic forces and moment terms will now be represented in terms of aerodynamic coefficients and variables such as angle of attack, angular velocity and control surface deflection for the purpose of linearization.

$$L = L_p p + L_{\delta_a}\delta_a \tag{12}$$

$$Y = Y_\beta \beta + Y_r r + Y_{\delta_r}\delta_r \tag{13}$$

$$Z = Z_\alpha \alpha + Z_q q + Z_{\delta_e}\delta_e \tag{14}$$

$$M = M_\alpha \alpha + M_q q + M_{\delta_e}\delta_e \tag{15}$$

$$N = N_\beta \beta + N_r r + N_{\delta_r}\delta_r \tag{16}$$

$$L_p = 1/2\rho V^2 S d C_{l_p} d/2V I_{xx}$$

$$L_{\delta_a} = 1/2\rho V^2 S d C_{l_{\delta_a}}/I_{xx}$$

$$Z_\alpha = 1/2\rho V^2 S C_{Z\alpha}/m$$

$$Z_q = 1/2\rho V^2 S C_{Z_q} d/2mV$$

$$Z_{\delta_e} = 1/2\rho V^2 S C_{Z_{\delta_e}}/m$$

$$M_\alpha = 1/2\rho V^2 S d C_{m_\alpha}/I_{yy}$$

$$M_q = 1/2\rho V^2 S C_{m_q} d^2/2I_{yy}V$$

$$M_{\delta_e} = 1/2\rho V^2 S d C_{m_{\delta_e}}/I_{yy}$$

$$Y_\beta = 1/2\rho V^2 S C_{Y_\beta}/m$$

$$Y_r = 1/2\rho V^2 S C_{Y_r} d/2mV$$

$$Y_{\delta_r} = 1/2\rho V^2 S C_{Y_{\delta_r}}/m$$

$$N_\beta = 1/2\rho V^2 S d C_{n_\beta}/I_{zz}$$

$$N_r = 1/2\rho V^2 S C_{n_r} d^2/2I_{zz}V$$

$$N_{\delta_r} = 1/2\rho V^2 S d C_{n_{\delta_r}}/I_{zz}$$

ρ: air density, V: speed of projectile, S: reference area, d: moment arm length, m: mass, a_y, a_z :inertial translational acceleration in y and z axes in body frame.

4.2. Linearized Plant

For a skid-to-turn missile autopilot roll channel is decoupled from pitch and yaw channels. Stabilization of roll axis is considered separately because the aim of roll autopilot is to keep roll rate at zero along the flight. This is because zero roll rate assumption enables decoupling of pitch and yaw equations and makes the linearization process possible. Gezer and Kutay [11] also states that a roll stabilizing autopilot is used for skid-to-turn (STT) missiles and proposes a model following control design for the linearized roll dynamics. Mohammadi et al [12] proposes a Input Output Linearization (IOL) + Uncertainty and Disturbance Estimation (UDE) approach to roll autopilot design similarly for a symmetrical skid-to-turn missile. Kohli and Chandar [13] also follows a linearization process before designing an UDE based roll autopilot for the missile.

When equations (9) and (12) are combined and transformed into a state space representation for roll channel we obtain:

$$\begin{bmatrix} \dot{p} \\ \dot{\delta_a} \\ \ddot{\delta_a} \\ p \end{bmatrix} = \begin{bmatrix} Lp & L_{\delta_a} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & -\omega_n^2 & -2\omega_n\zeta & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} p \\ \delta_a \\ \dot{\delta_a} \\ \int p \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \omega_n^2 \\ 0 \end{bmatrix} \delta_{a_{com}} \tag{17}$$

In this representation the states are :

- p: roll angular rate
- δ_a: aileron deflection angle
- $\dot{\delta_a}$: aileron deflection angular rate
- $\int p$:integral state of p

Since short range air defense missiles are usually long and thin geometry they have very low roll inertia therefore a small roll disturbance cause large angular rates. Hence, an integral state of roll angular rate is added in order to increase rejection of roll disturbances which is an important feature of a roll autopilot as also mentioned in [11,12,13].

The input of this state-space model is deflection commands. The actuator dynamics is included in the plant which can be seen in last two rows of A matrix. The actuator dynamics is modeled as second order system with natural frequency ω_n and damping ratio of ζ.

Now the pitch channel linearization can be carried out by combining (8),(10),(14), (15). Considering that $\dot{w} \approx \dot{V}\alpha + V\dot{\alpha}$ due to small angle assumption and solving for derivatives of states a_z, q, δ_e and $\dot{\delta_e}$. Finally the obtained A matrix for the state-space representation is:

$$A_{pitch} = \begin{bmatrix} \frac{Z_\alpha}{V} + \frac{Z_q M_\alpha}{Z_\alpha} + \frac{\dot{u}}{V} & Z_\alpha + \frac{\dot{u} Z_q}{V} - \frac{Z_q^2 M_\alpha}{Z_\alpha} + Z_q M_q & \frac{Z_{\delta_e} \dot{u}}{V} - \frac{Z_q Z_{\delta_e} M_\alpha}{Z_\alpha} + Z_q M_{\delta_e} & Z_{\delta_e} & 0 \\ \frac{M_\alpha}{Z_\alpha} & M_q - \frac{Z_q M_\alpha}{Z_\alpha} & M_{\delta_e} - \frac{Z_{\delta_e} M_\alpha}{Z_\alpha} & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & -\omega_n^2 & -2\omega_n \zeta & 0 \\ 1 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{18}$$

Since pitch and yaw channels have same dynamics due to symmetry when same operations are followed the A matrix for yaw channel is obtained:

$$A_{yaw} = \begin{bmatrix} \frac{Y_\beta}{V} + \frac{Y_r N_\beta}{Y_\beta} - \frac{\dot{u}}{V} & -Y_\beta + \frac{\dot{u} Y_r}{V} - \frac{Y_r^2 N_\beta}{Y_\beta} + Y_r N_r & \frac{Y_{\delta_r} \dot{u}}{V} - \frac{Y_r Y_{\delta_r} N_\beta}{Y_\beta} + Y_r N_{\delta_r} & Y_{\delta_r} & 0 \\ \frac{N_\beta}{Y_\beta} & N_r - \frac{Y_r N_\beta}{Y_\beta} & N_{\delta_r} - \frac{Y_{\delta_r} N_\beta}{Y_\beta} & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & -\omega_n^2 & -2\omega_n \zeta & 0 \\ 1 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{19}$$

If pitch and yaw channels are wanted to be represented by a single state-space representation they can be combined and written as:

$$
\begin{bmatrix} \dot{a}_z \\ \dot{q} \\ \dot{\delta}_e \\ \ddot{\delta}_e \\ a_z \\ \dot{a}_y \\ \dot{r} \\ \dot{\delta}_r \\ \ddot{\delta}_r \\ a_y \end{bmatrix} = \begin{bmatrix} A_{pitch} & 0 \\ 0 & A_{yaw} \end{bmatrix} \begin{bmatrix} a_z \\ q \\ \delta_e \\ \dot{\delta}_e \\ \int a_z \\ a_y \\ r \\ \delta_r \\ \dot{\delta}_r \\ \int a_y \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ \omega_n^2 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & \omega_n^2 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \delta_{e_{com}} \\ \delta_{r_{com}} \end{bmatrix} \tag{20}
$$

In this study acceleration is chosen as a state variable to control instead of angle of attack. This way all states are measured through the IMU sensor and encoder angle sensors in CAS. The states can be listed as:

- a_z : IMU acceleration measurement in body Z-axis
- q : IMU angular rate measurement around Y-axis
- δ_e : Elevator deflection angle calculated by actuator encoder measaurements
- $\dot{\delta}_e$: Elevator deflection angular rate
- $\int a_z$: Integral of IMU acceleration measurement in body Z-axis
- a_y : IMU acceleration measurement in body Y-axis
- r : IMU angular rate measurement around Z-axis
- δ_r : Rudder deflection angle calculated by actuator encoder measaurements
- $\dot{\delta}_r$: Rudder deflection angular rate
- $\int a_y$: Integral of IMU acceleration measurement in body Y-axis

4.3. Adaptive Model Predictive Control

The idea behind the model predictive control theory has been introduced in Section 2.4. In this section, the mathematical representation of the MPC cost optimization will be briefly shown based on works of Wang [20], Schwenzer et al [24] and Abhishek [27] before moving on to Adaptive MPC.

The mathematical formulation for the optimization can be shown for an example predictive system within an optimization window in order to calculate predicted plant output [20].

$$x(k+1) = f(x(k), u(k))$$

,

$$y(k) = h(x(k))$$

MPC minimizes a cost function J for minimizing the error between reference and predicted model output [24]:

$$\underset{u}{min} \quad J(x(k), u(.))$$

$$\underset{u}{min} \quad \sum_{i=N_1}^{N_2} ||r(k+i|k) - y(k+i|k)||$$

This optimization process used in MATLAB can be presented with more detail based on Wang's book [20].

$$J = (R_s - Y)^T (R_s - Y) + \Delta U^T \bar{R} \Delta U$$

First term in this cost function is to minimize tracking error and the second term is to reduce control input steps. $\bar{R} = r_w I_{N_c x N_c}$ $(r_w \geq 0)$ where r_w is a tuning parameter for closed-loop performance.

Previous expression can be re-written in order to find the optimal ΔU that will minimize J:

$$J = (R_s - Fx(k_i))^T(R_s - Fx(k_i)) - 2\Delta U^T\phi^T(R_s - Fx(k_i)) + \Delta U^T(\phi^T\phi + \bar{R})\Delta U$$

For the condition of minimum J :

$$\frac{\partial J}{\partial \Delta U} = -2\phi^T(R_s - Fx(k_i)) + 2(\phi^T\phi + \bar{R})\Delta U = 0$$

Then the optimal solution for the control signal is:

$$\Delta U = (\phi^T\phi + \bar{R})^{-1}\phi^T(R_s - Fx(k_i))$$

Even though the optimization process calculate series of optimal control inputs over a prediction horizon only the first step of control inputs is used. In the next time step the same process is repeated with updated measurements[20].

There are various types of MPC such as a single linear MPC, Multiple Explicit MPC, Adaptive MPC, Nonlinear MPC. Aboelela [22] also implements MPC for missile control design by linearizing the model for a specific flight condition and testing the linear MPC controller in an LTI environment. Asfihani [23] has also proposed a linear MPC control design for a missile guidance loop. The engagement kinematic between target and missile is linearized for MPC and the designed controller generates acceleration commands needed to intercept the target. In our thesis study, as mentioned before we have researched autopilot design rather than guidance section however both studies have followed a linearization process for the model in order to use linear MPC.

Unlike [22] and [23] this study proposes a MPC controller that covers a wider flight envelope. The missile will be operating in different flight regions which can have significantly different system dynamics. Considering this, a single linear MPC is not able to handle such a wide operation range. Because there needs to be multiple linearization points to cover all flight regions if it is decided to use a linear control approach. A nonlinear MPC would not have such a limitation however that would require high computation time which exceeds the onboard computer capabilities. Therefore, since a smooth transition between linearized systems is crucial an Adaptive MPC controller is selected.

Adaptive MPC requires linear state space system matrices which can dynamically change in discrete form. This means that every time step the MPC controller runs optimization for the currently linearized system. Therefore, state space matrices are computed inside autopilot system every time step using data received from IMU sensor.

Inputs of Adaptive MPC controller are:

- discretized state space model matrices
- prediction horizon
- control horizon
- control input minimum and maximum values
- control input minimum and maximum rate values
- output weights

Fawzy et al [22] chooses sampling time of MPC controller with trial and error based on tracking performance and proposes 0.01 seconds. Taking computation efficiency into account, the step time for the MPC in this thesis is also chosen to be 0.01 seconds (10 milliseconds, i.e. 100 Hz frequency). Prediction horizon defines the length of the optimization window that projects future behavior of the system. MATLAB recommends that prediction horizon should be chosen such that it covers settling time of the system. In this study, it is desired to have approximately 0.5 seconds settling time therefore prediction horizon is set to be 50 steps. Control horizon defines the length of control action steps in each prediction cycle. MATLAB recommends a control horizon length between 10 and 20 percent of prediction horizon. Considering that suggestion and also through trial and error process the control horizon is set to be 5 steps.

One of the most beneficial advantages of MPC theory is that it takes input and output constraints into consideration for optimization [27]. In this study hard constraints on control input variables are set considering that the actuators have physical limits. Therefore input constraints are set to be 20 degrees in nominal case. Upon a case of actuator failure, the total capacity of remaining actuators will drop therefore input constraint of the MPC will also change online accordingly. This way we could benefit from MPC theory by combining it with FTC approach. Output weights for optimization are chosen from interpolation table with respect to mach number, weight values at mach breakpoints are tuned with trial and error process.

When given above inputs, after each sample time (0.01s) the controller generates deflection command as output.

The system models are designed as two separate state-space models. Pitch and yaw channels are combined as single state space model. When this model is fed to adaptive MPC controller it generates two deflection commands: elevator and rudder deflection commands. The other state space model is for roll channel. The reason why roll channel is not included and designed separately is because such a system could not be represented linearly due to nonlinear relation between system states (shown in next sections).Roll rate (p) which is also system state would have to be multiplied with other states such as q, r, ay, az and that would not be possible in linear state space representation. Therefore, roll rate (p) is taken as a constant in pitch-yaw channel, only in roll channel it is considered as a state variable. This means that the nonlinear relation between the roll channel and pitch-yaw channels are not included in system dynamics model presented to the controller. Hence, this relation will act as a disturbance to the both channels. In order to minimize this effect, roll controller will have to stabilize the roll rate to zero along the flight.

Prediction horizon is chosen such that it covers the expected time of settling.

4.4. Fault Tolerant Control

4.4.1. Fault Diagnosis

In the autopilot system a fault diagnosis algorithm is coded. This algorithm computes tracking errors for each actuator over time. If the error exceeds given limit for over 50 milliseconds, the faulty actuator is then detected. Once the faulty actuator is detected FTC method called Control Mixer Reconfiguration is activated according to the remaining actuators.

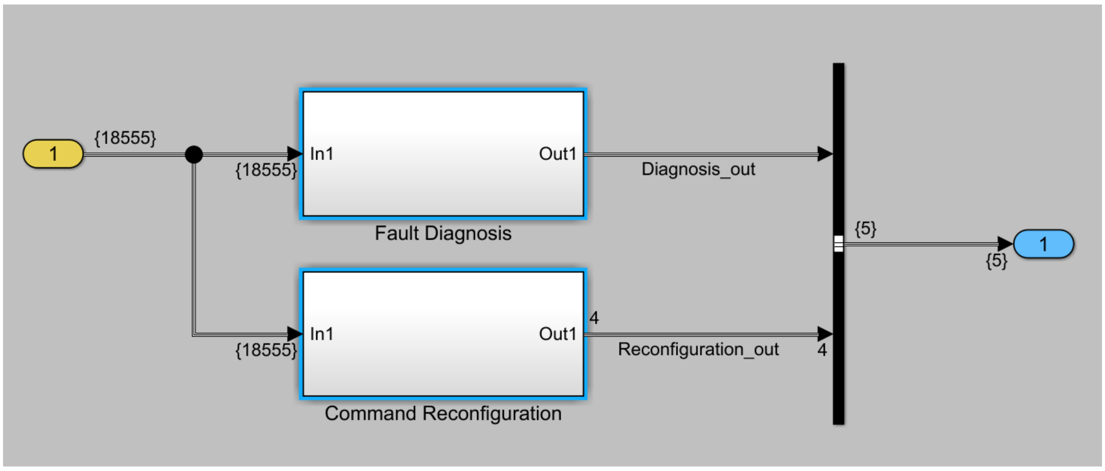


Figure 4.1 FTC Block Modeled in Autopilot

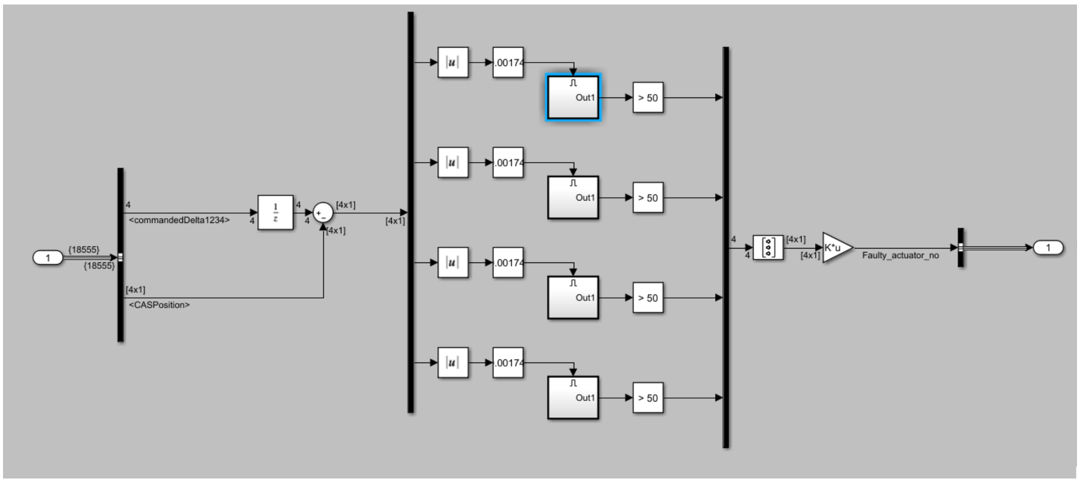


Figure 4.2 Fault Diagnosis Algorithm in Simulink Model

4.4.2. Reconfiguration of Control Mixer

In section 2.3 various types and steps of Fault Tolerant Control were introduced. For the purpose of tolerating actuator faults different applications have been demonstrated in literature. For instance, Tong et al [1] proposes an adaptive approach with back-stepping and another method described as 'shape variables' in order to handle under-actuation failure for a supersonic missile. Liue et al [29] proposes actuator reconfiguration active FTC method for actuator faults. In this method the effect of faulty actuator is defined with a corresponding new B matrix for state-space representation. And in order to keep preserve state-feedback controller design the gain vector 'K' is also adjusted accordingly ($BK = B_f K_f$). Louhan et al [6] proposes a deep learning algorithm that identifies multiple fault cases and implements command reconfiguration (same as control mixer reconfiguration) for the case of single actuator failure. Bajpai et al [28] also proposes configuration of control mixer as an active fault tolerance method for actuator failure scenario. In this study, control mixer is reconfigurated (also called command reconfiguration [6]) for the purpose of active fault tolerant control upon detection of actuator failures.

Various examples for fault tolerant methods in literature can also be listed in Table 4.1 below.

Table 4.1 Various FTC Methods for Actuator Faults

Ref. No	Actuator Fault	FTC Method	For Missile
6	TLoE/PLoE	Command Reconfiguration	Yes
1	TLoE/PLoE	Adaptive	Yes
30	PLoE	H_∞ Bumpless Transfer	Yes
31	TLoE/PLoE	Adaptive	No
28	TLoE	Control Mixer Reconfig.	No
32	TLoE/PLoE	Adaptive	No
33	PLoE	Adaptive	No
29	TLoE/PLoE	Actuation Reconfig.	No
34	PLoE	Gain Scheduling	No
35	PLoE	LPV	No

The FTC method applied in this study is an active method which involves reconfiguring the distribution of deflection commands into actuator commands. The reason to choose this method is because in case of a singular actuator failure the system still has 3 remaining actuators and 3 control channels (roll, pitch yaw). This means that the system is properly-actuated when missing one actuator. Also the aerodynamic CFD data is obtained relative to elevator, rudder and aileron deflections instead of actuator positions. Therefore reconfiguring the relation between ERA and actuator positions is the most practical solution that also promise full recovery of the controller.

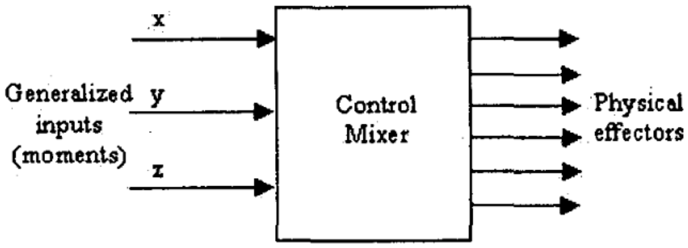


Figure 4.3 Control Mixer [28]

In this study failure of a single actuator is studied. Considering each actuator having independent electric motor an electrical dysfunction or a mechanical failure can cause such a situation. In that case, the actuator will stop responding to given commands and the control surface will stay still at last angular position. Since there are three desired motions to control (roll, pitch, yaw), when one of the four actuators fail the system is properly-actuated with remaining three actuators. This means that theoretically roll, pitch and yaw channels can still be controlled however a new distribution to remaining actuators is necessary. By implementing the proposed control mixer reconfiguration capability to control all three channels will be preserved.

Transformation between ERA and 1234 actuator commands for cross configuration is shown below:

$$\begin{bmatrix} \delta_1 \\ \delta_2 \\ \delta_3 \\ \delta_4 \end{bmatrix} = \begin{bmatrix} -1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & -1 & 1 \\ -1 & -1 & 1 \end{bmatrix} \begin{bmatrix} \delta_e \\ \delta_r \\ \delta_a \end{bmatrix} \tag{21}$$

$$\begin{bmatrix} \delta_e \\ \delta_r \\ \delta_a \end{bmatrix} = 1/4 \begin{bmatrix} -1 & 1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} \delta_1 \\ \delta_2 \\ \delta_3 \\ \delta_4 \end{bmatrix} \tag{22}$$

Now, depending on which actuator fails there exists a new transformation for the remaining three actuators. Also, the angle of the locked actuator will also be important for the transformation because the remaining three actuators need to cancel the effect of failed one in addition to required deflection commands. Below each actuator failure is demonstrated.

If 1st actuator fails there exists a new transformation between ERA and 1234 delta commands which is shown below:

$$\delta_e' = \delta_e - f/4$$
$$\delta_r' = \delta_r - f/4$$
$$\delta_a' = \delta_a - f/4$$

f is the angular position of failed actuator which is locked.

$$\begin{bmatrix} \delta_2 \\ \delta_3 \\ \delta_4 \end{bmatrix} = 2 \begin{bmatrix} 1 & 0 & 1 \\ 1 & -1 & 0 \\ 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} \delta_e' \\ \delta_r' \\ \delta_a' \end{bmatrix} \tag{23}$$

If 2nd actuator fails:

$$\delta_e' = \delta_e - f/4$$
$$\delta_r' = \delta_r + f/4$$
$$\delta_a' = \delta_a - f/4$$

$$\begin{bmatrix} \delta_1 \\ \delta_3 \\ \delta_4 \end{bmatrix} = 2 \begin{bmatrix} 1 & 0 & 1 \\ 0 & -1 & 1 \\ -1 & 1 & 0 \end{bmatrix} \begin{bmatrix} \delta_e' \\ \delta_r' \\ \delta_a' \end{bmatrix} \tag{24}$$

If 3rd actuator fails:

$$\delta_e' = \delta_e + f/4$$
$$\delta_r' = \delta_r + f/4$$
$$\delta_a' = \delta_a - f/4$$

$$\begin{bmatrix} \delta_1 \\ \delta_2 \\ \delta_4 \end{bmatrix} = 2 \begin{bmatrix} 1 & 1 & 0 \\ 0 & -1 & 1 \\ -1 & 0 & 1 \end{bmatrix} \begin{bmatrix} \delta_e' \\ \delta_r' \\ \delta_a' \end{bmatrix} \tag{25}$$

If 4th actuator fails:

$$
\begin{aligned}
\delta_e' &= \delta_e + f/4 \\
\delta_r' &= \delta_r - f/4 \\
\delta_a' &= \delta_a - f/4
\end{aligned}
$$

$$
\begin{bmatrix} \delta_1 \\ \delta_2 \\ \delta_3 \end{bmatrix} = 2 \begin{bmatrix} 0 & 1 & 1 \\ 1 & -1 & 0 \\ -1 & 0 & 1 \end{bmatrix} \begin{bmatrix} \delta_e' \\ \delta_r' \\ \delta_a' \end{bmatrix} \tag{26}
$$

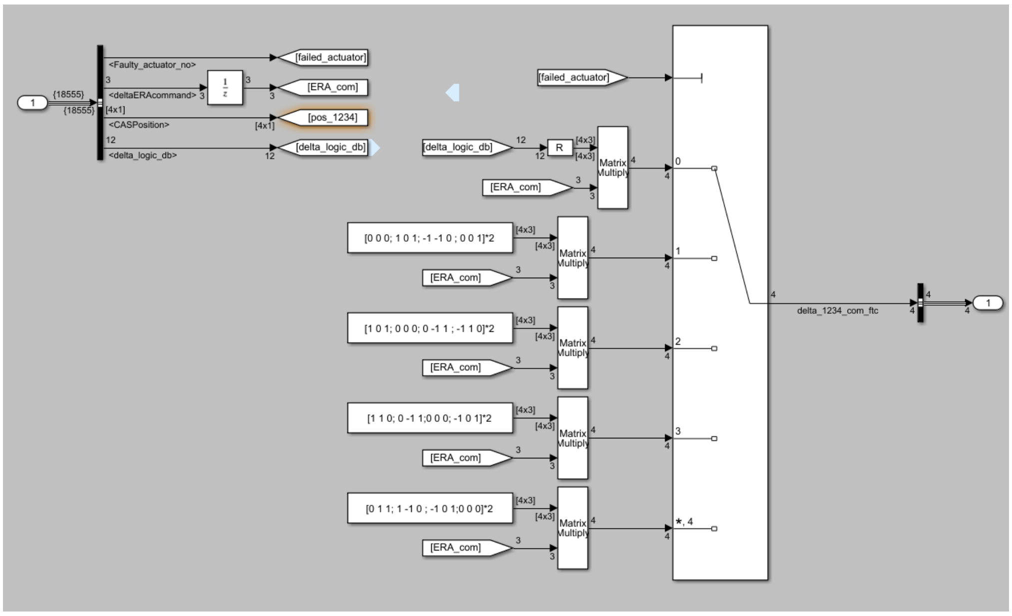


Figure 4.4 Command Reconfiguration Block in Simulink

4.5. Nonlinear Analysis Model

In order to test the effectiveness of designed controllers a nonlinear analysis tool is modeled in Simulink. The main models of the tool is listed below:

- Atmosphere Model

- Propulsion Model
- Mass-Inertia Model
- Aerodynamics Model
- Flight Dynamics Model
- Missile Subsystems Model (IMU,CAS,OBC)

4.5.1. Atmosphere Model

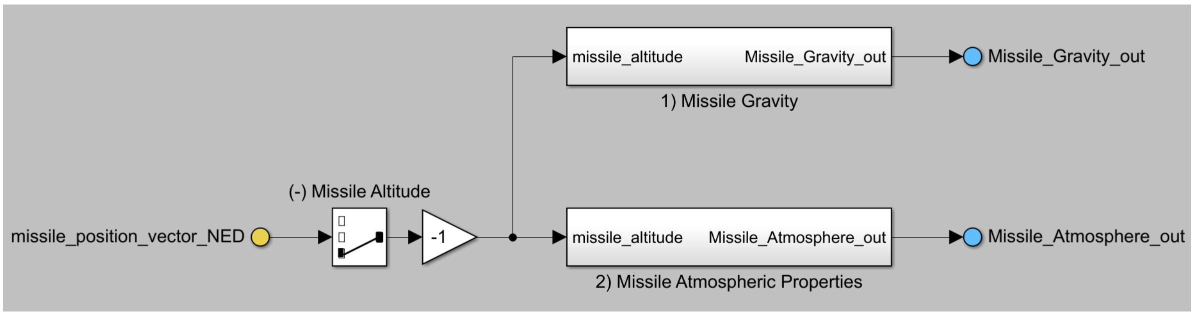


Figure 4.5 Atmosphere Model

In environment model there are gravity and atmospheric properties models. Input of these models is missile altitude. These models make calculations for gravitational acceleration, air density, air temperature, air pressure and speed of sound. Gravitational acceleration is used for finding gravitational force acting on missile body. Air density is used for calculating dynamic air pressure which is necessary for determining aerodynamic forces and moments acting on body. Speed of sound is used for calculating missile's mach number which is used for finding aerodynamic coefficients in that flight conditions.

4.5.2. Propulsion Model

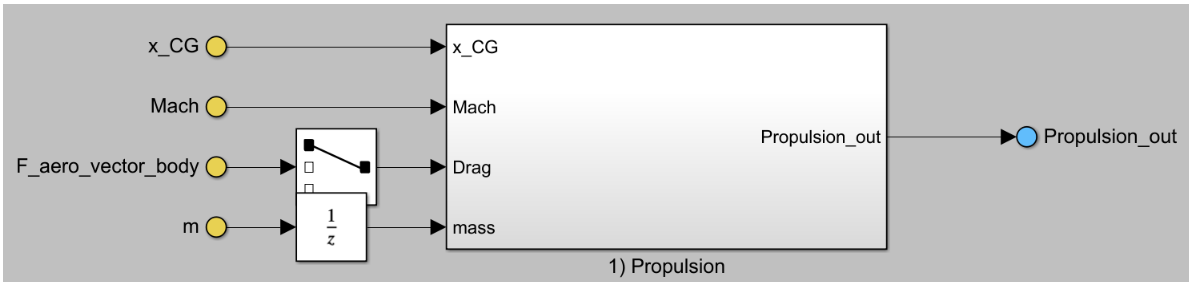


Figure 4.6 Propulsion Model

Propulsion block models propulsive forces and moments acting on missile body. Missile has a solid fuel rocket engine, therefore once it is ignited a pre-designed thrust profile is obtained. In the model, an axial thrust force profile is implemented which changes over time. If a constant speed analysis is intended, thrust force is equalized to negative aerodynamic drag force. Inputs of this block are center of gravity location along longitudinal axis (xCG), mach number and aerodynamic force along x-axis(drag force). The outputs are propulsive forces and moments along each three axes.

4.5.3. Mass-Inertia Model

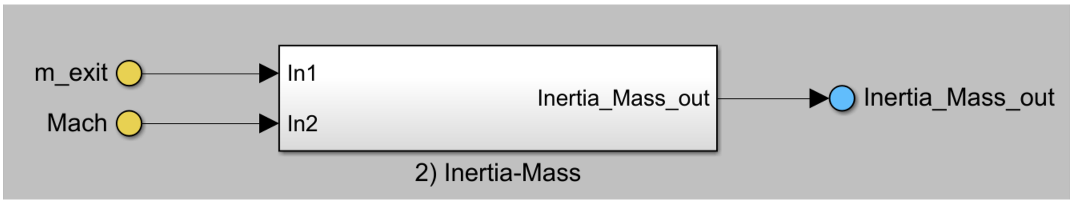


Figure 4.7 Mass-Inertia Model

As the rocket engine expel burned fuel out the mass and inertia of the body decrease. Mass and inertia values for full and empty rocket fuel are known. A linear decrease over time is assumed and modeled in this block. As mass change inertia values also change using a look-up table.

4.5.4. Aerodynamics Model

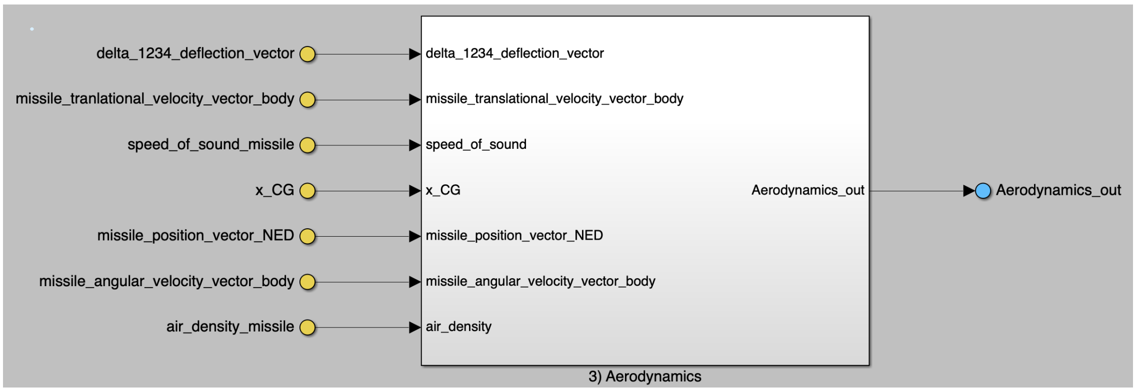


Figure 4.8 Aerodynamics Model

Aerodynamics block receives information about flight conditions and generates aerodynamic coefficients, aerodynamic force and moments acting on the body. Firstly, body translational velocity components are used to calculate angle of attack and sideslip angles. These parameters along with mach number and control surface deflections make the basis of flight conditions. Then, these flight conditions are fed to lookup tables. The aerodynamic coefficient data for these lookup tables are obtained by CFD analysis. CFD analysis are made at certain mach numbers, angles of attack and deflection angles. The lookup tables provide interpolation between these points during simulation therefore at each time step aerodynamic coefficients in that moment can be estimated continuously through out the simulation run. Dynamic pressure acting on missile is calculated since air pressure and missile velocities are known, thereby total aerodynamic force and moments acting on the body are computed.

4.5.5. Flight Dynamics Model

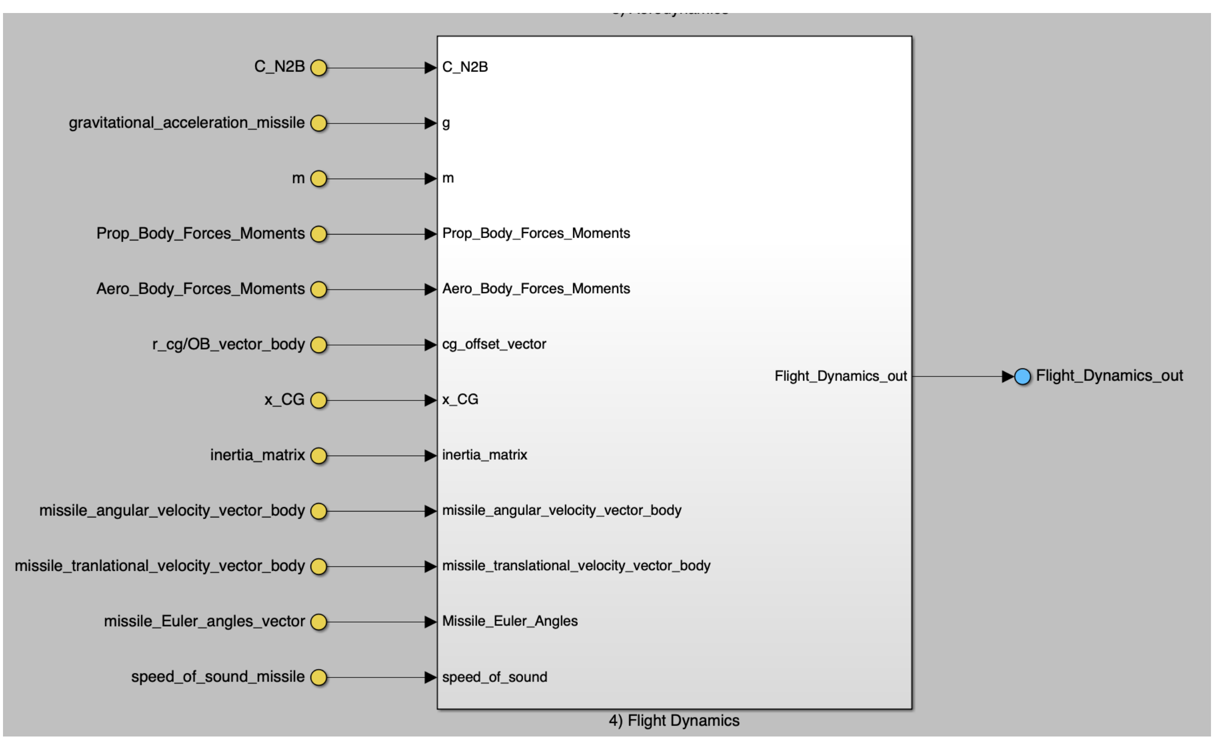


Figure 4.9 Flight Dynamics Model

Flight Dynamics block implements nonlinear equations of motion shown in (2) and (4) and given the initial conditions it computes translational and angular velocities acting on the missile body. These velocities are then used to calculate position and orientation of the missile.

4.5.6. Missile Subsystems Model

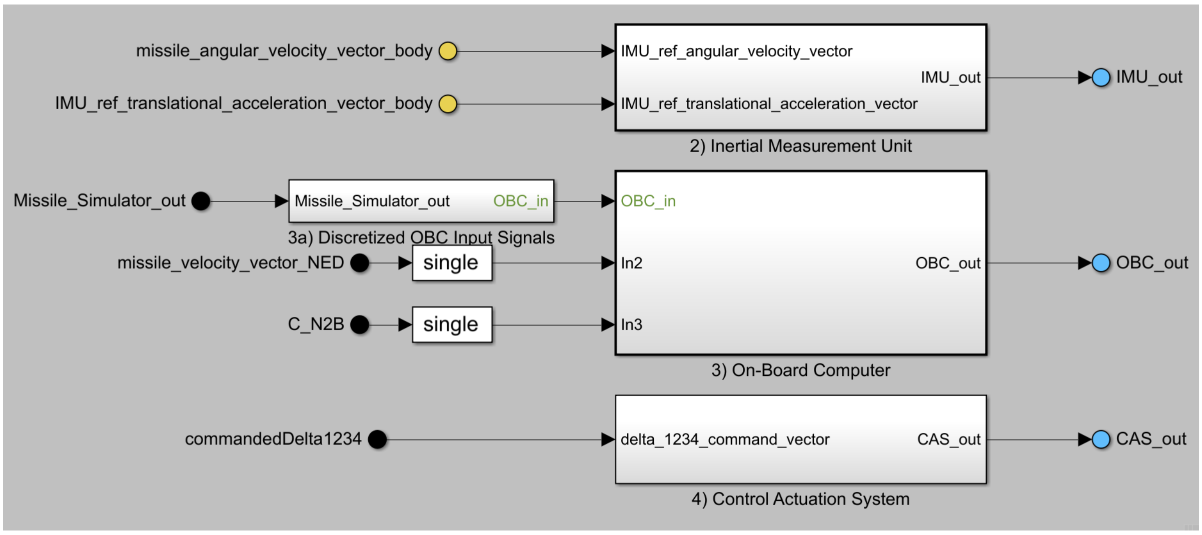


Figure 4.10 Missile Subsystems Model

In scope of this study, only three onboard missile systems have significance. These are Control Actuation System (CAS), Inertial Measurement Unit (IMU) and onboard computer which contains autopilot system. CAS block receives deflection angle commands from the autopilot system. Actuator dynamic is modeled as second order transfer function. Inertial Measurement Unit block models the acceleration and angular rates that would be measured. In this block, IMU's location with respect to missile c.g. and rate of change of c.g. location are received in order to consider coriolis, centrepedal acceleration terms. Also sensor resolution and noise is added in this block. In both CAS and IMU blocks sensor measurement delays are modeled with simulink delay blocks.

Actuator dynamics is be modeled as a second order transfer function as shown below:

$$\frac{\delta_a}{\delta_{a_{com}}} = \frac{\omega_A^2}{s^2 + 2\zeta_A\omega_A + \omega_A^2}$$

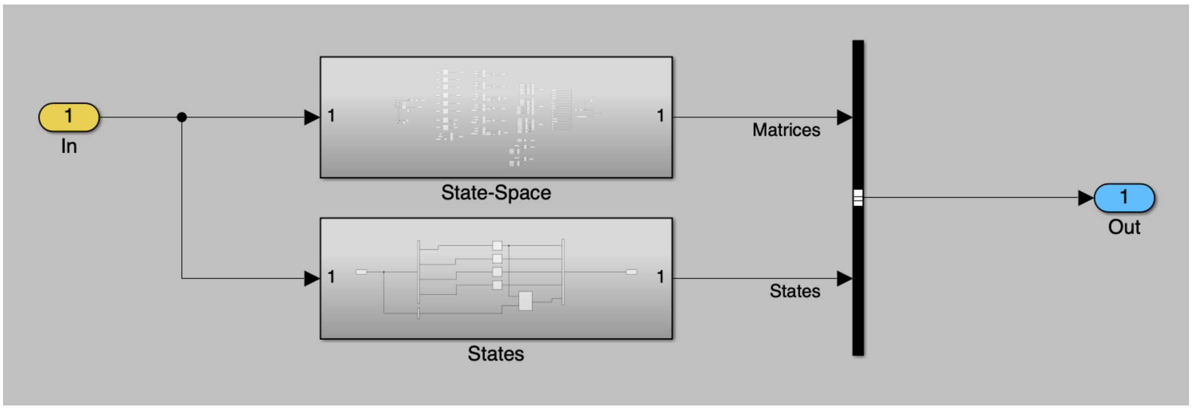


Figure 4.11 Autopilot Control Parameters Block

Inside autopilot block there are two important blocks for the controller. Control Parameters and Control Law blocks. In Control Parameters, state space matrices and states are prepared using the model obtained in Section 5.2. In order to compute A matrix components velocity, mass, inertia, axial acceleration and aerodynamic derivative estimates are needed. The velocity is calculated using IMU acceleration measurements. Mass and inertia are estimated using a lookup table with flight time input. Since mass and inertia change profiles are pre-known they can be estimated with this method. Axial acceleration value is provided by IMU sensor again. Aerodynamic derivatives were previously estimated by Taylor's first order expansion linearization(shown in Equations of Motion (12-16)). These estimations were obtained at certain flight condition breakpoints. Again lookup tables are used for interpolation. Mach number, alpha, beta and deflection angles are values already provided to OBC and using them as an input aerodynamic derivatives are interpolated with lookup tables.

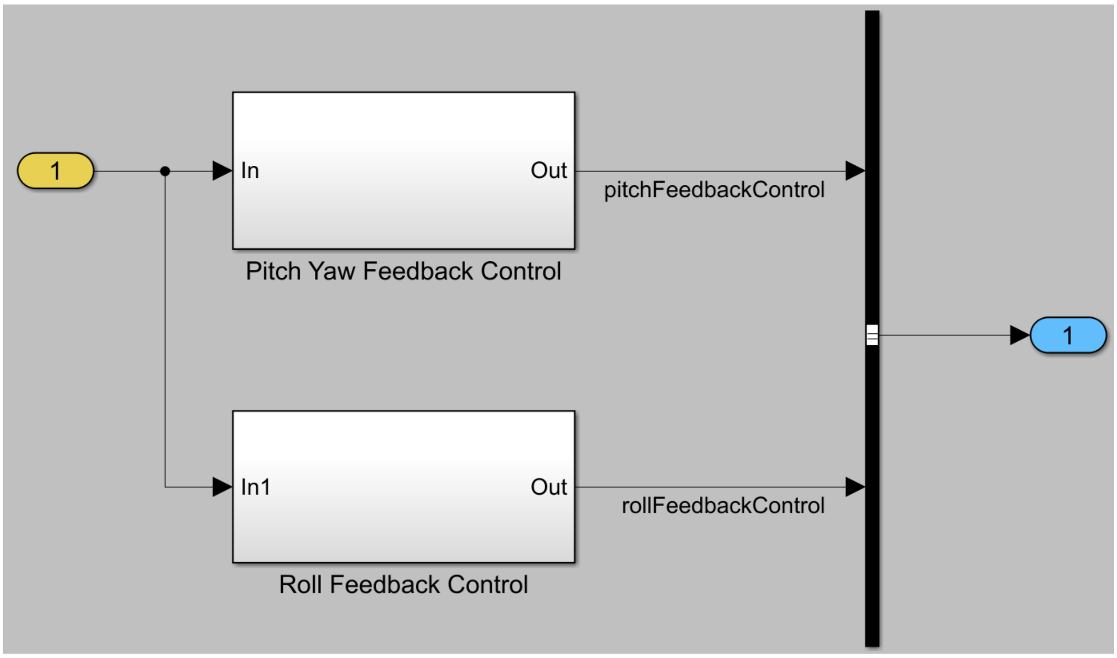
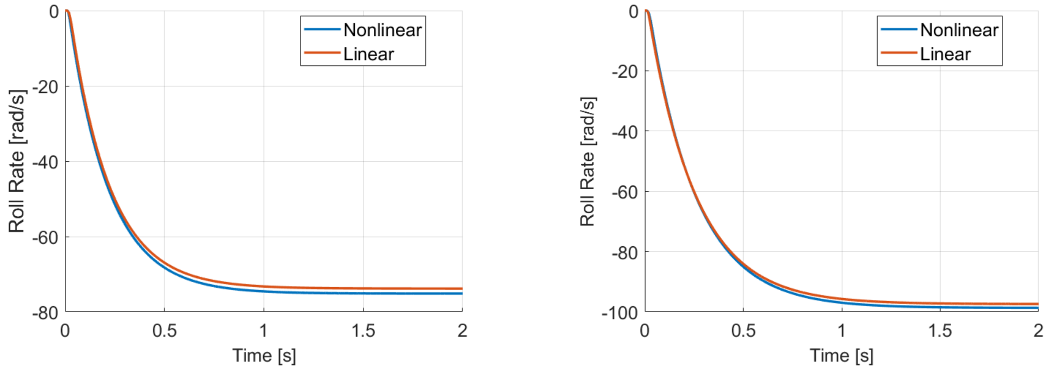


Figure 4.12 Autopilot Control Law Block

5. SIMULATION RESULTS

5.1. Linear and Nonlinear Models Comparison

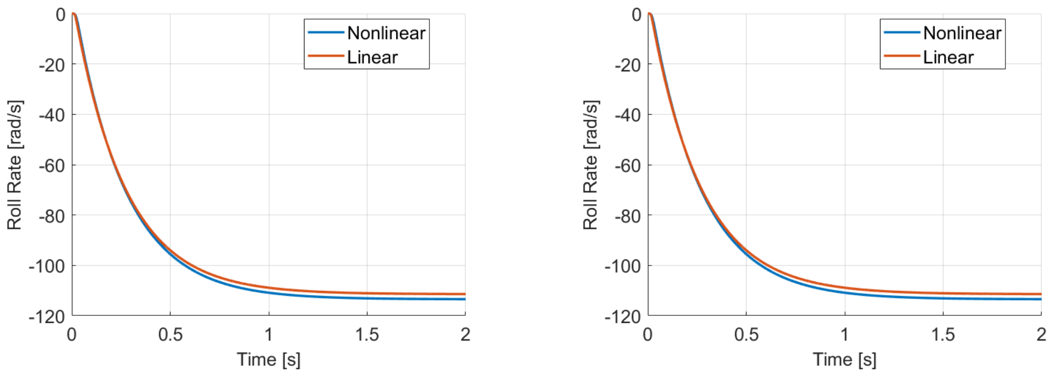
Before implementing adaptive MPC controller, the validity of the linearization should be analysed. In this analysis, step actuator deflections are given in both linear and nonlinear model. For roll channel 1 degree aileron step deflection is given and the roll angular rate responses are compared. For pitch channel, 1 degree elevator step deflection is given and afterwards acceleration and angular rate responses are comparerd. Since the pitch and yaw channels contain identical dynamics due to symmetry of the missile geometry only the pitch channel results will be demonstrated to represent lateral dynamics in results. The linear model response is obtained by a discrete state space model which is based on the model shown in section 5.2. The sample time for the discrete system is equal to the sample time of the MPC controller (0.01 seconds). The nonlinear response is obtained by the model described in section 5.3 The results show that linear model response show highly similar dynamics as in the nonlinear model.



(a) Linear vs Nonlinear Roll Rate Response to Step Aileron at Mach=1.2

(b) Linear vs Nonlinear Roll Rate Response to Step Aileron at Mach=1.4

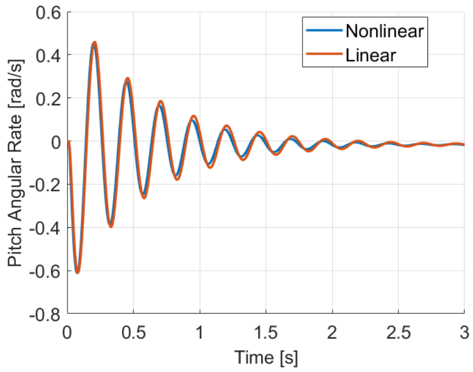
Figure 5.1 Roll Rate Responses to Step Aileron in Linear and Nonlinear Models for Mach=1.2 and Mach=1.4



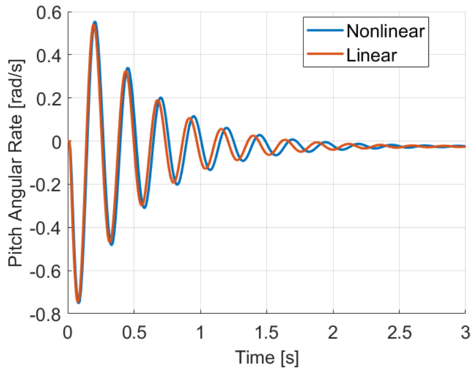
(a) Linear vs Nonlinear Roll Rate Response to Step Aileron at Mach=1.6

(b) Linear vs Nonlinear Roll Rate Response to Step Aileron at Mach=1.8

Figure 5.2 Roll Rate Responses to Step Aileron in Linear and Nonlinear Models for Mach=1.6 and Mach=1.8

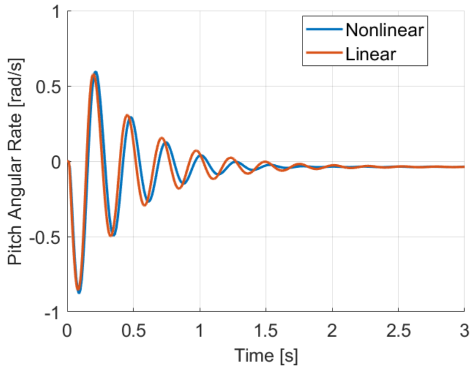


(a) Linear vs Nonlinear Acceleration Response to Step Elevator at Mach=1.2

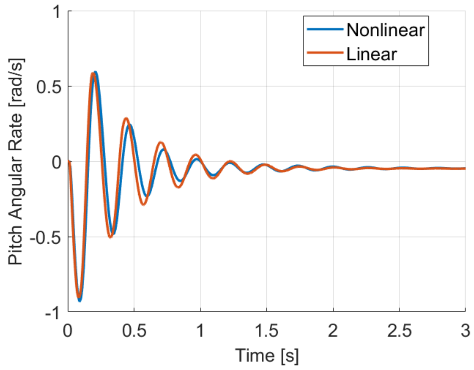


(b) Linear vs Nonlinear Acceleration Response to Step Elevator at Mach=1.4

Figure 5.3 Pitch Rate Responses to Step Elevator in Linear and Nonlinear Models for Mach=1.2 and Mach=1.4

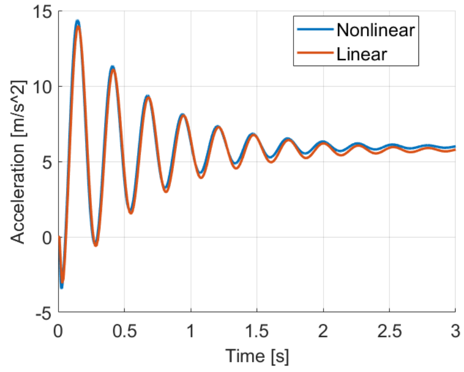


(a) Linear vs Nonlinear Acceleration Response to Step Elevator at Mach=1.2

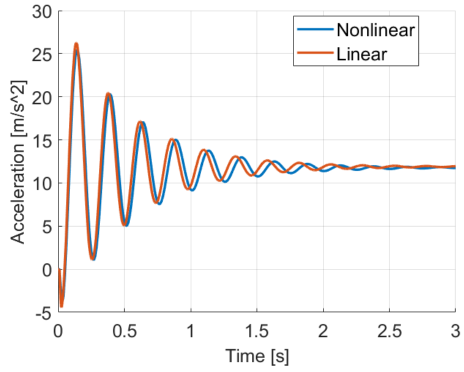


(b) Linear vs Nonlinear Acceleration Response to Step Elevator at Mach=1.4

Figure 5.4 Pitch Responses to Step Elevator in Linear and Nonlinear Models for Mach=1.6 and Mach=1.8

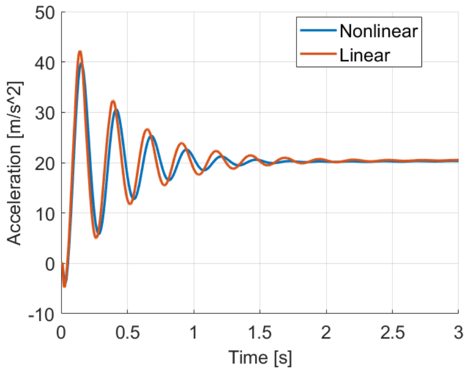


(a) Linear vs Nonlinear Acceleration Response to Step Elevator at Mach=1.2

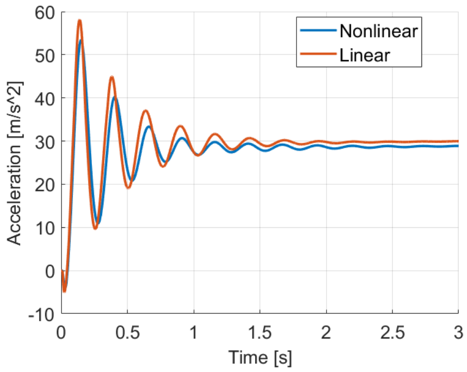


(b) Linear vs Nonlinear Acceleration Response to Step Elevator at Mach=1.4

Figure 5.5 Acceleration Responses to Step Elevator in Linear and Nonlinear Models for Mach=1.2 and Mach=1.4



(a) Linear vs Nonlinear Acceleration Response to Step Elevator at Mach=1.6



(b) Linear vs Nonlinear Acceleration Response to Step Elevator at Mach=1.8

Figure 5.6 Acceleration Responses to Step Elevator in Linear and Nonlinear Models for Mach=1.6 and Mach=1.8

5.2. Autopilot Test Results in Nonlinear Analysis Model

The nonlinear analysis model shown in section 5.3 is used for obtaining results. In this analysis, flight conditions at multiple points are chosen. These conditions are flight altitude and mach number. At each mach number corresponding mass, inertia and c.g. values are also updated. The flight conditions that were analysed can be shown in table below.

Table 5.1 Analysis Cases

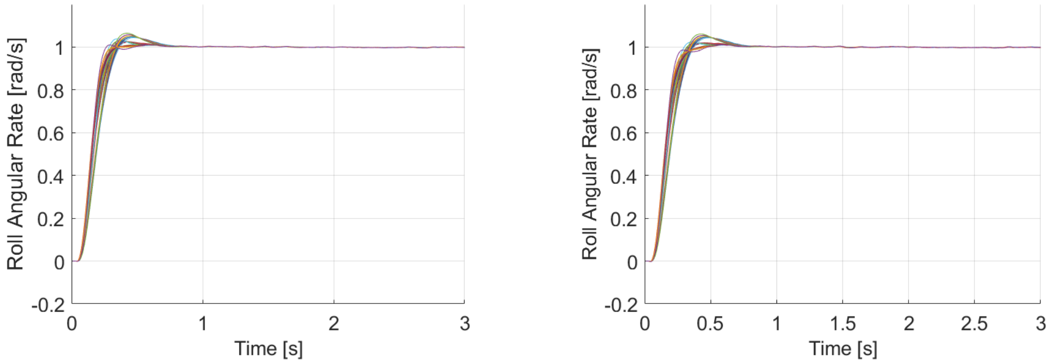
Analysis	Mach Points	Altitude Points	Fault	FTC
1	[1.2 1.4 1.6 1.8]	[1 2 3] km	No	No
2	[1.2 1.4 1.6 1.8]	[1 2 3] km	Yes	No
3	[1.2 1.4 1.6 1.8]	[1 2 3] km	Yes	Yes

In each analysis there exist 81 runs with different errors and delays. There are three cases for c.p. locations, three different sensor delays for acceleration, gyro and actuator position sensors. All these conditions make 81 runs when summed. Thereby, with this analysis nonlinear response of the system under % 25 percent c.p. error and 4,8,12 milliseconds sensor delays. The aim of the analysis is to show the controller design is robust enough preserve stability under these model errors and sensor delays. In the case of roll autopilot analysis c.p. error is replaced with Cl coefficient error.

Table 5.2 Errors and Delays

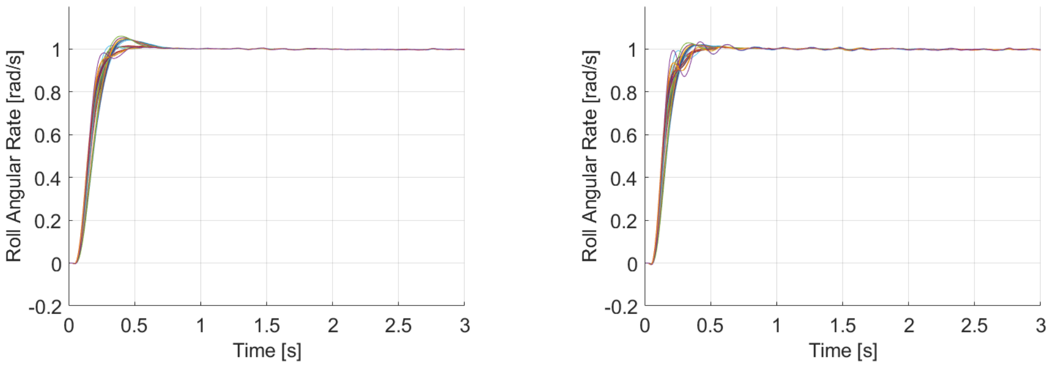
Analysis	Aerodynamic Error	Accelerometer Delay	Gyro Delay	CAS Delay
Pitch	[-%25 0 %25] C.P.	[4 8 12] ms	[4 8 12] ms	[4 8 12] ms
Roll	[-%25 0 %25] Cl	[4 8 12] ms	[4 8 12] ms	[4 8 12] ms

The analysis is done for three different cases. Firstly the roll and pitch autopilots are tested in above test conditions without any actuator failure case. Then, tests are repeated with one of the actuators being completely dysfunctional (TLoE). 1000 meters altitude tests are repeated under failure of actuator no 1, 2000 meters analysis is repeated under failure of actuator no 2 and 3000 meters analysis is conducted again under failure of actuator no 3. Finally, these tests are repeated one more time with active FTC method implementation.



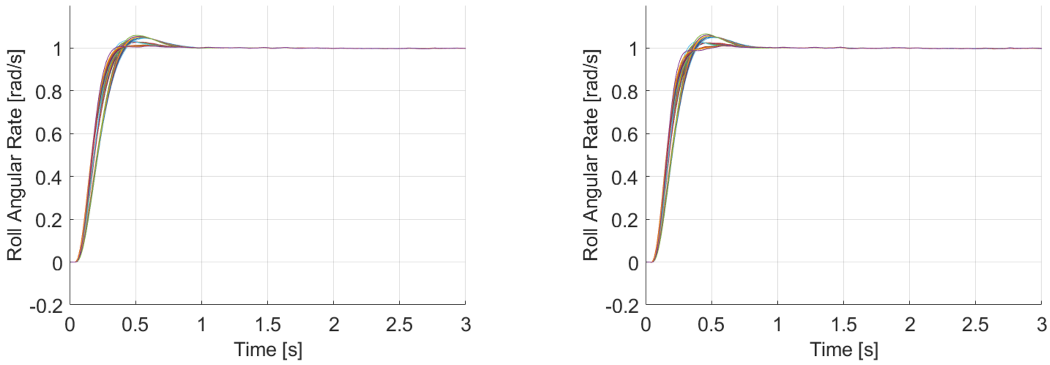
(a) Roll Step Command at Mach=1.2 Altitude=1000m (b) Roll Step Command at Mach=1.4 Altitude=1000m

Figure 5.7 Roll Channel Nonlinear Model Analysis at 1000m Altitude



(a) Roll Step Command at Mach=1.6 Altitude=1000m (b) Roll Step Command at Mach=1.8 Altitude=1000m

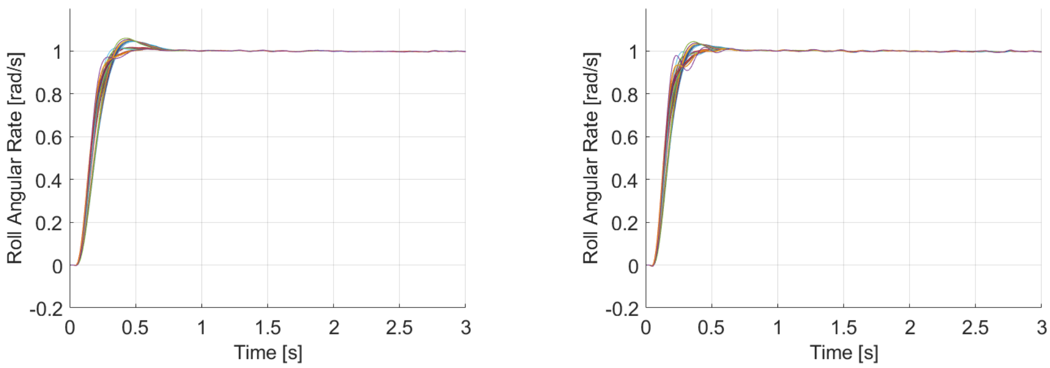
Figure 5.8 Roll Channel Nonlinear Model Analysis at 1000m Altitude



(a) Roll Step Command at Mach=1.2 Altitude=2000m

(b) Roll Step Command at Mach=1.4 Altitude=2000m

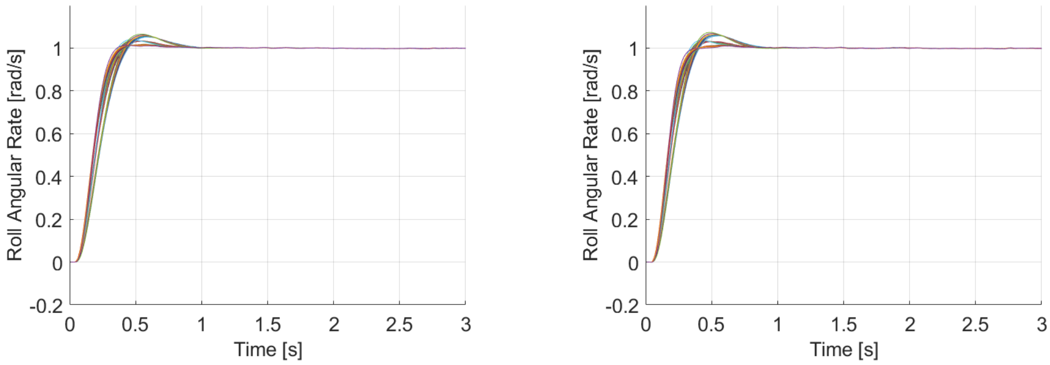
Figure 5.9 Roll Channel Nonlinear Model Analysis at 2000m Altitude



(a) Roll Step Command at Mach=1.6 Altitude=2000m

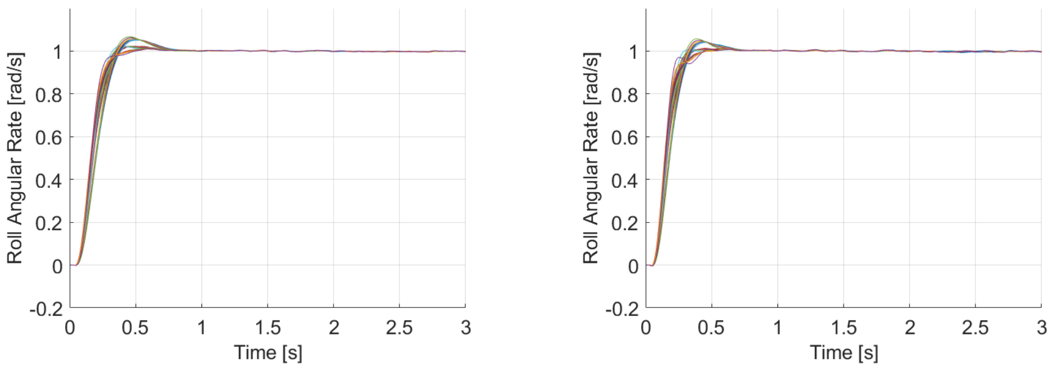
(b) Roll Step Command at Mach=1.8 Altitude=2000m

Figure 5.10 Roll Channel Nonlinear Model Analysis at 2000m Altitude



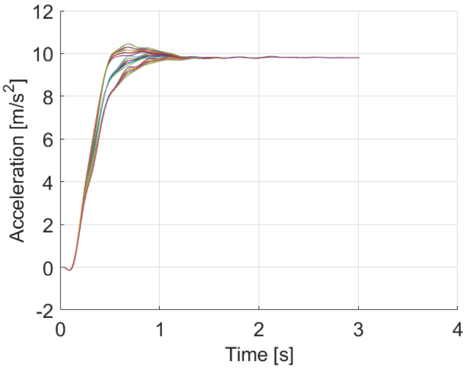
(a) Roll Step Command at Mach=1.2 Altitude=3000m (b) Roll Step Command at Mach=1.4 Altitude=3000m

Figure 5.11 Roll Channel Nonlinear Model Analysis at 3000m Altitude

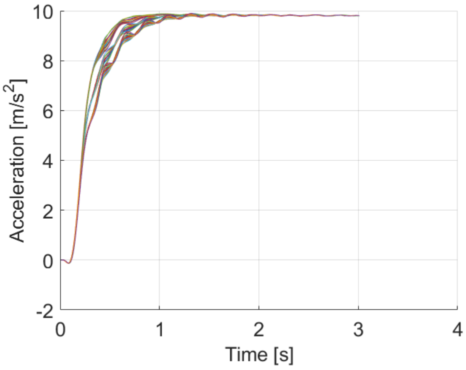


(a) Roll Step Command at Mach=1.6 Altitude=3000m (b) Roll Step Command at Mach=1.8 Altitude=3000m

Figure 5.12 Roll Channel Nonlinear Model Analysis at 3000m Altitude

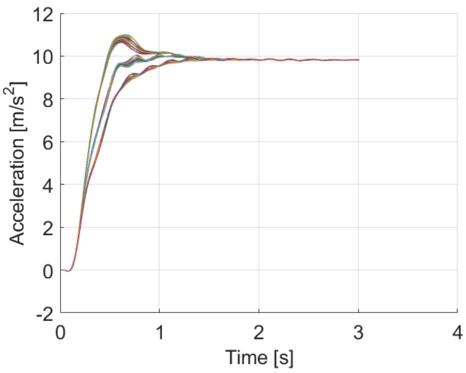


(a) Acceleration Step Command at Mach=1.2

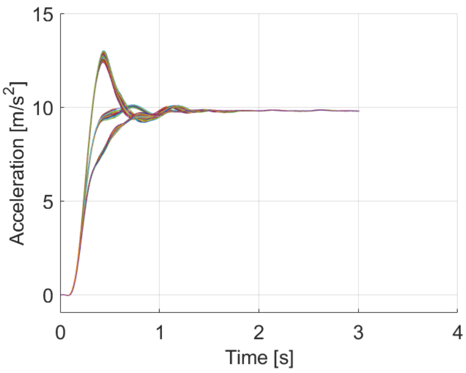


(b) Acceleration Step Command at Mach=1.4

Figure 5.13 Pitch Channel Nonlinear Model Analysis at 1000m Altitude

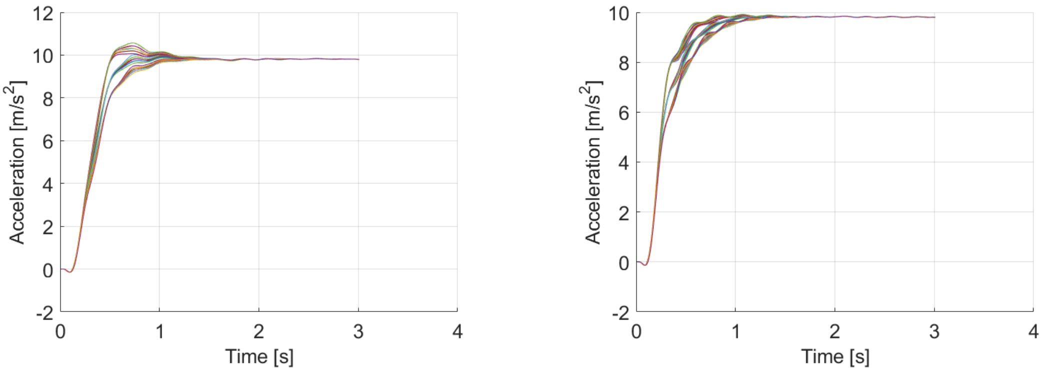


(a) Acceleration Step Command at Mach=1.6



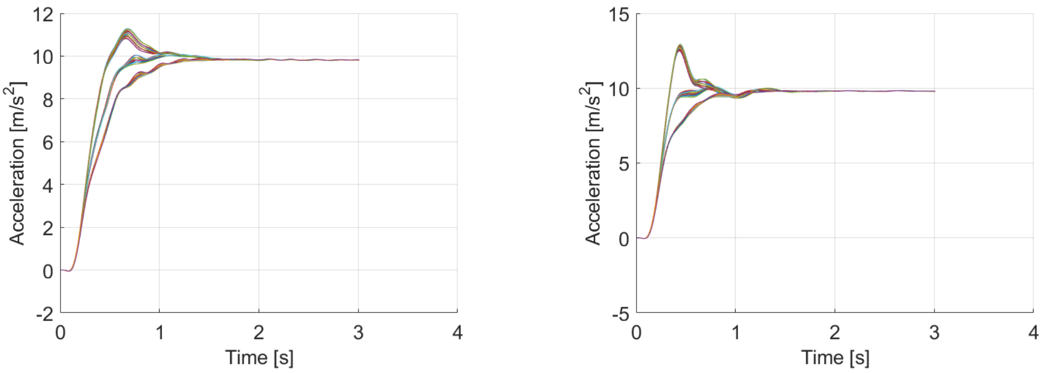
(b) Acceleration Step Command at Mach=1.8

Figure 5.14 Pitch Channel Nonlinear Model Analysis at 1000m Altitude



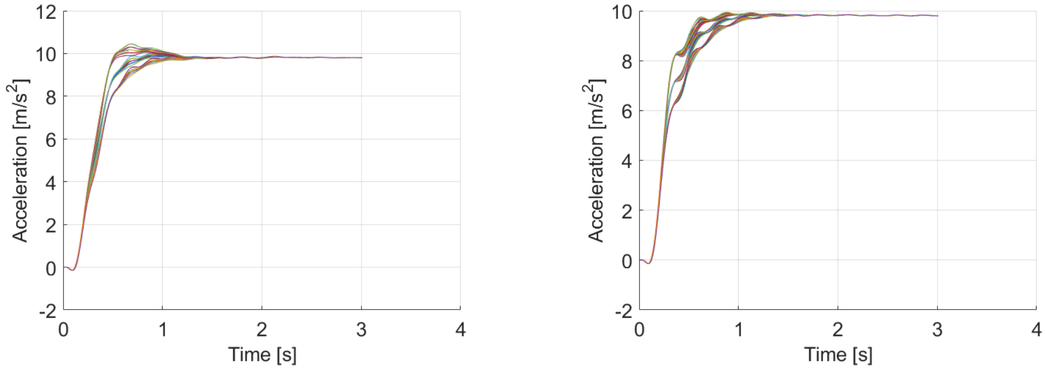
(a) Acceleration Step Command at Mach=1.2 (b) Acceleration Step Command at Mach=1.4

Figure 5.15 Pitch Channel Nonlinear Model Analysis at 2000m Altitude



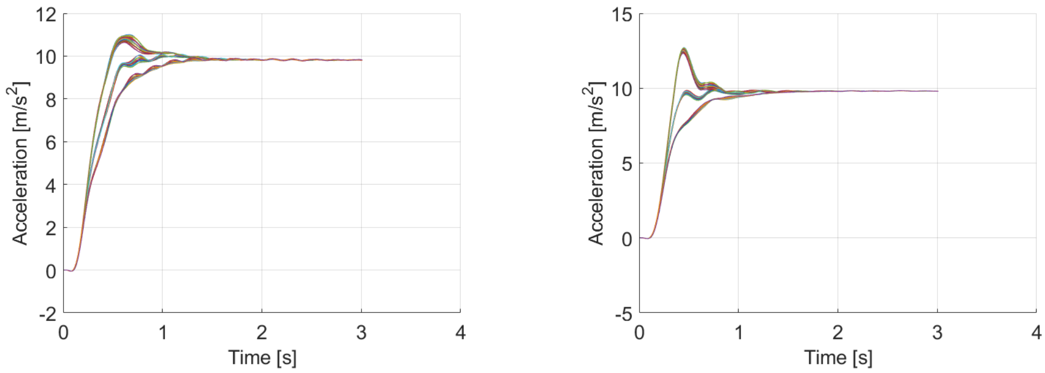
(a) Acceleration Step Command at Mach=1.6 (b) Acceleration Step Command at Mach=1.8

Figure 5.16 Pitch Channel Nonlinear Model Analysis at 2000m Altitude



(a) Acceleration Step Command at Mach=1.2 (b) Acceleration Step Command at Mach=1.4

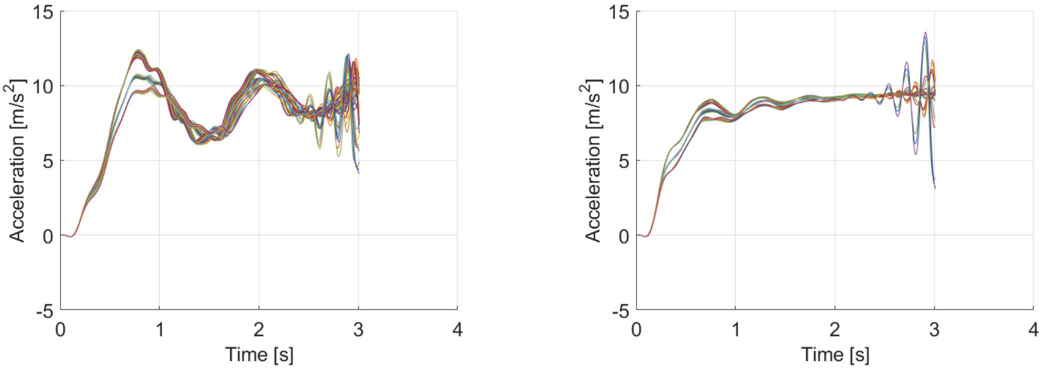
Figure 5.17 Pitch Channel Nonlinear Model Analysis at 3000m Altitude



(a) Acceleration Step Command at Mach=1.6 (b) Acceleration Step Command at Mach=1.8

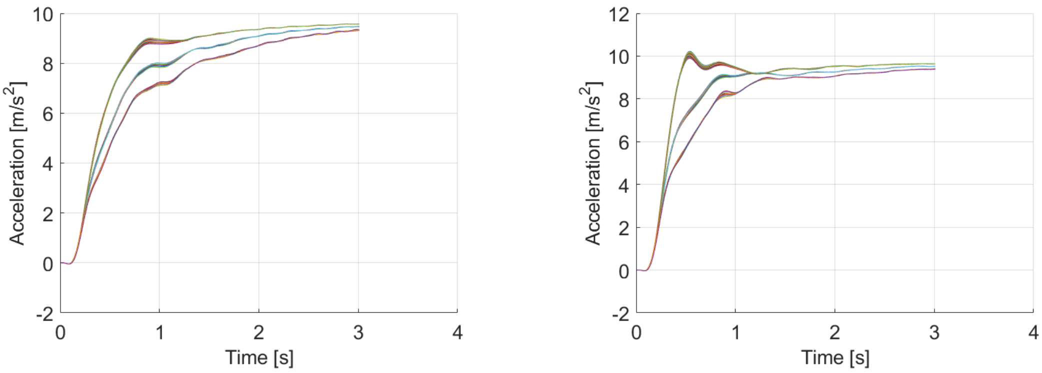
Figure 5.18 Pitch Channel Nonlinear Model Analysis at 3000m Altitude

Figures 5.7-5.18 show that adaptive MPC controller provides a stable acceleration tracking with zero steady-state error in spite of %25 aerodynamic error and 12 milliseconds sensor delays. Center of pressure error affects transient behavior greatly. Without integral state this error would cause a significant steady-state error however due to integral state steady-state error becomes zero however difference in transient phase remains. Sensor delays mostly effect the stability of the closed loop system. When sensor delays reach high values slight oscillations start to appear which indicates being close to marginally stable state.



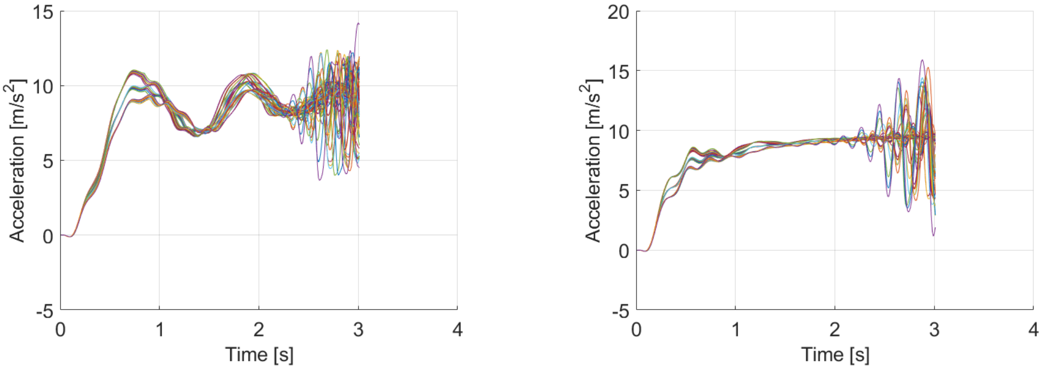
(a) Acceleration Step Command at Mach=1.2 (b) Acceleration Step Command at Mach=1.4

Figure 5.19 Pitch Channel Nonlinear Model Analysis at 1000m Altitude under 1st Actuator Failure



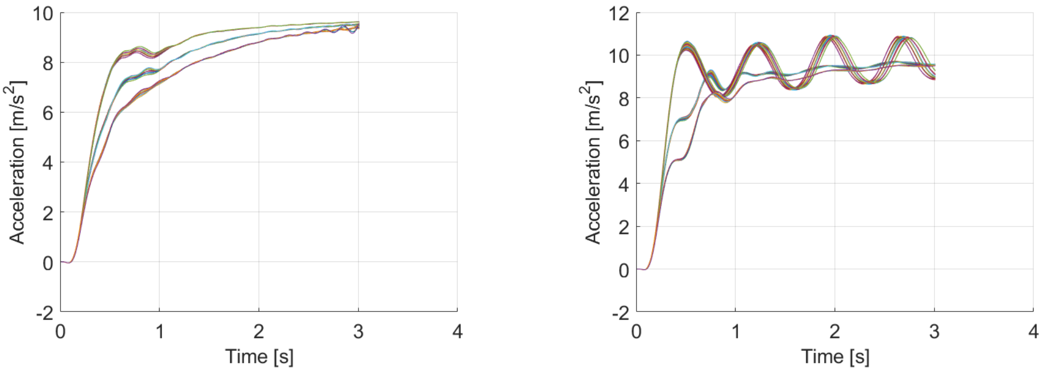
(a) Acceleration Step Command at Mach=1.6 (b) Acceleration Step Command at Mach=1.8

Figure 5.20 Pitch Channel Nonlinear Model Analysis at 1000m Altitude under 1st Actuator Failure



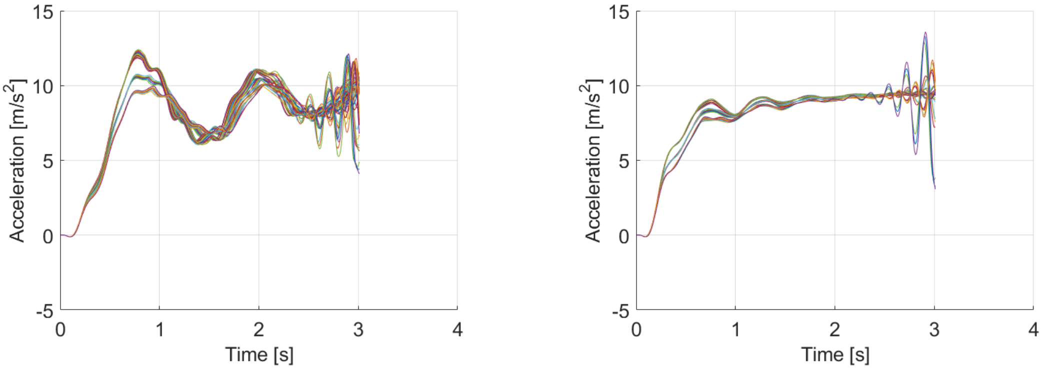
(a) Acceleration Step Command at Mach=1.2 (b) Acceleration Step Command at Mach=1.4

Figure 5.21 Pitch Channel Nonlinear Model Analysis at 2000m Altitude under 2nd Actuator Failure



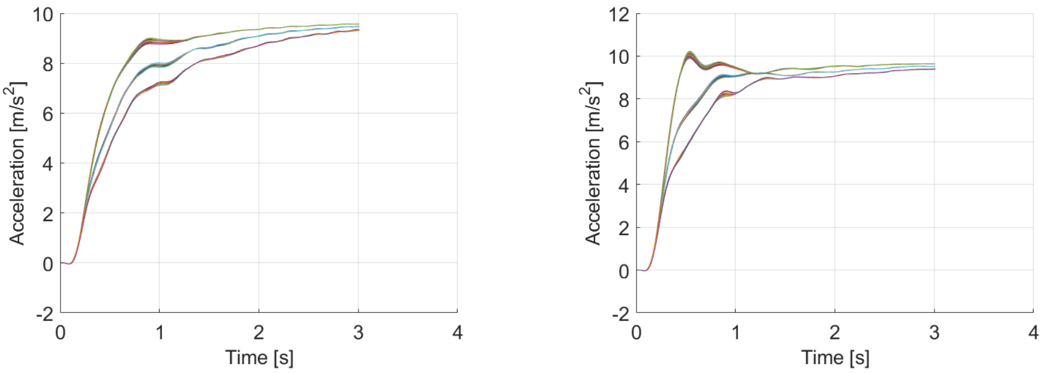
(a) Acceleration Step Command at Mach=1.6 (b) Acceleration Step Command at Mach=1.8

Figure 5.22 Pitch Channel Nonlinear Model Analysis at 2000m Altitude under 2nd Actuator Failure



(a) Acceleration Step Command at Mach=1.2 (b) Acceleration Step Command at Mach=1.4

Figure 5.23 Pitch Channel Nonlinear Model Analysis at 3000m Altitude under 3rd Actuator Failure

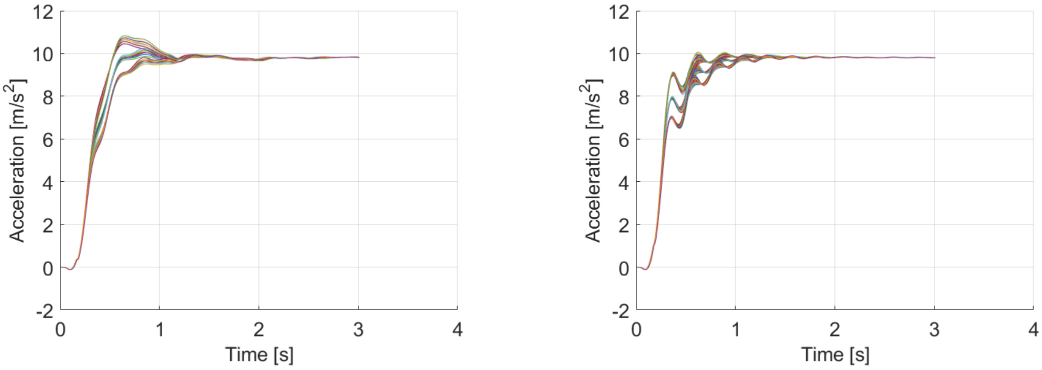


(a) Acceleration Step Command at Mach=1.6 (b) Acceleration Step Command at Mach=1.8

Figure 5.24 Pitch Channel Nonlinear Model Analysis at 3000m Altitude under 3rd Actuator Failure

Figures 5.19-5.24 show the effect of an actuator failure without an active fault tolerant action. In some mach numbers the effect is great increase in settling time. Because commands are distributed to actuators as if there are still four of them, however since one of them is dysfunctional this results with less elevator deflection than desired. Also another effect is due to uneven aileron distribution roll autopilot can not stabilize roll rate. Since one of the linearization assumption for pitch/yaw channels was zero roll rate, this causes inaccuracy of linear model proportional to roll rate. Therefore in some cases like 1.2 Mach and 1.4 Mach system becomes completely unstable.

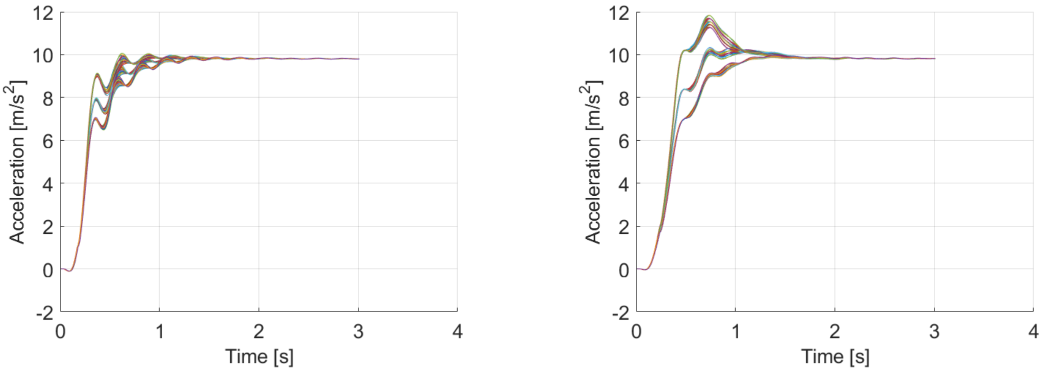
These results also prove that an active FTC method is needed in order to restore system performance upon an actuator failure.



(a) Acceleration Step Command at Mach=1.2

(b) Acceleration Step Command at Mach=1.4

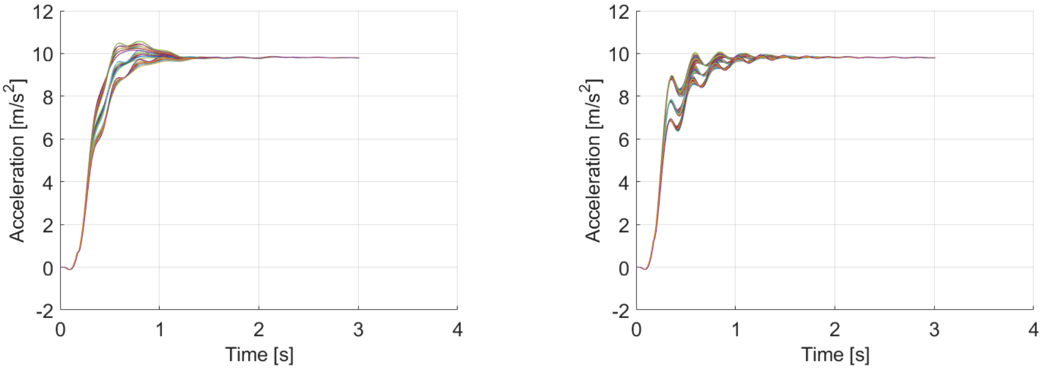
Figure 5.25 Pitch Channel Nonlinear Model Analysis at 1000m Altitude under 1st Actuator Failure with FTC



(a) Acceleration Step Command at Mach=1.6

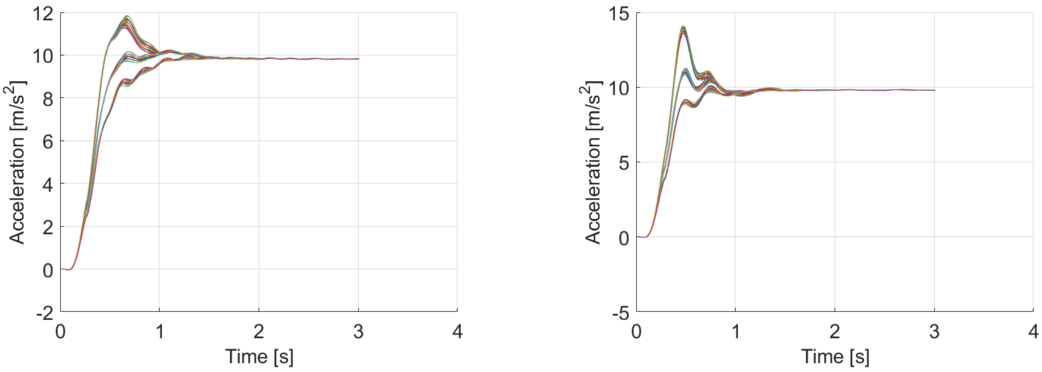
(b) Acceleration Step Command at Mach=1.8

Figure 5.26 Pitch Channel Nonlinear Model Analysis at 1000m Altitude under 1st Actuator Failure with FTC



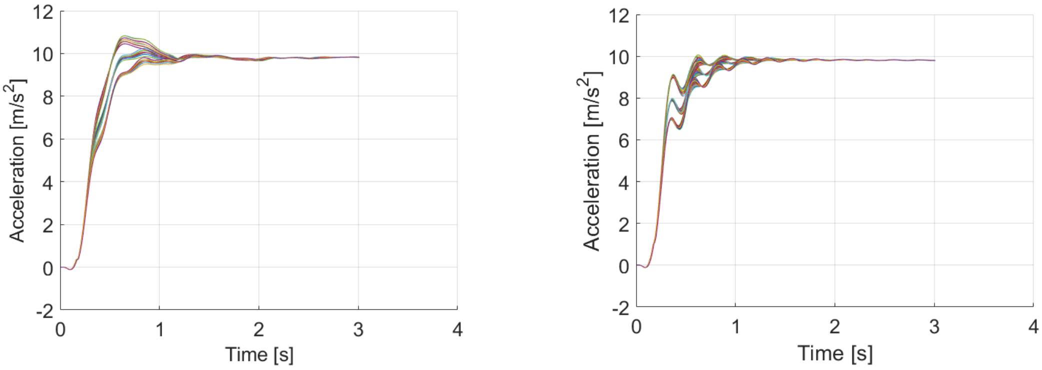
(a) Acceleration Step Command at Mach=1.2 (b) Acceleration Step Command at Mach=1.4

Figure 5.27 Pitch Channel Nonlinear Model Analysis at 2000m Altitude under 2nd Actuator Failure with FTC



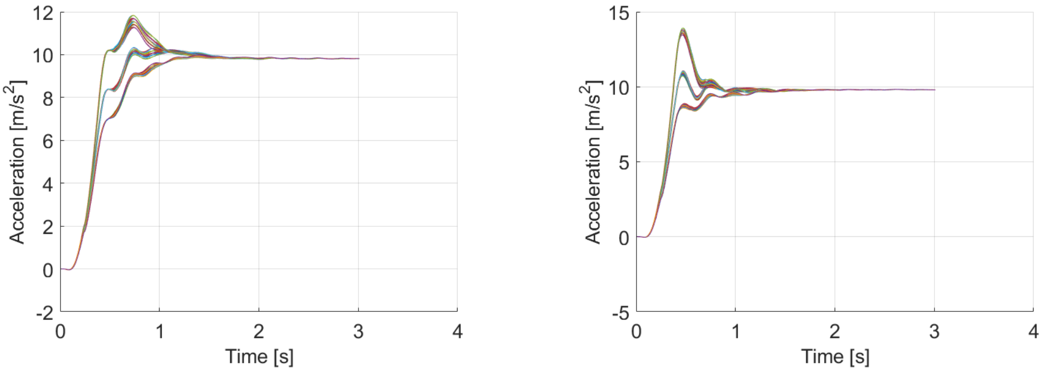
(a) Acceleration Step Command at Mach=1.6 (b) Acceleration Step Command at Mach=1.8

Figure 5.28 Pitch Channel Nonlinear Model Analysis at 2000m Altitude under 2nd Actuator Failure with FTC



(a) Acceleration Step Command at Mach=1.2 (b) Acceleration Step Command at Mach=1.4

Figure 5.29 Pitch Channel Nonlinear Model Analysis at 3000m Altitude under 3rd Actuator Failure with FTC



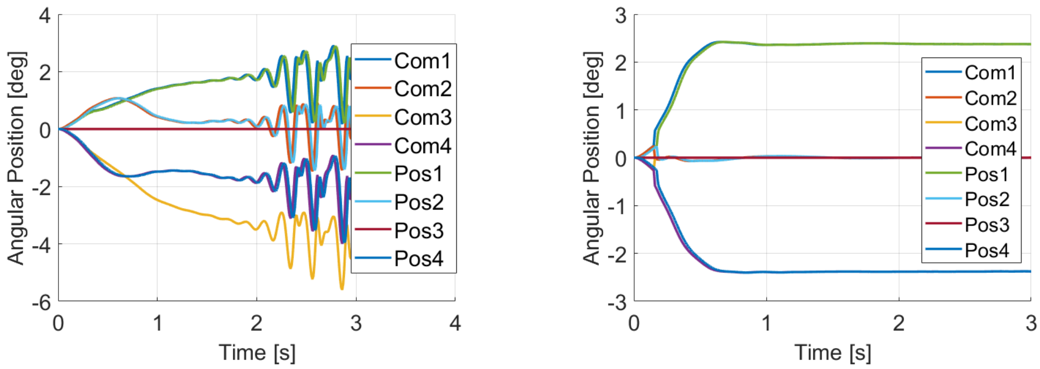
(a) Acceleration Step Command at Mach=1.6 (b) Acceleration Step Command at Mach=1.8

Figure 5.30 Pitch Channel Nonlinear Model Analysis at 3000m Altitude under 3rd Actuator Failure with FTC

In above figures, it can be seen that with command reconfiguration FTC method the catastrophic effects of actuator failure are eliminated. Failure of the system is prevented which is the main purpose of fault tolerant control.

5.3. Single Case Study

In order to observe the comparison in more detail one single run is chosen. In this simulation run mach number is 1.2, altitude is 1000 meters and sensor delays are 12 milliseconds. Fault scenario is third actuator being locked. One g step acceleration reference command is given.

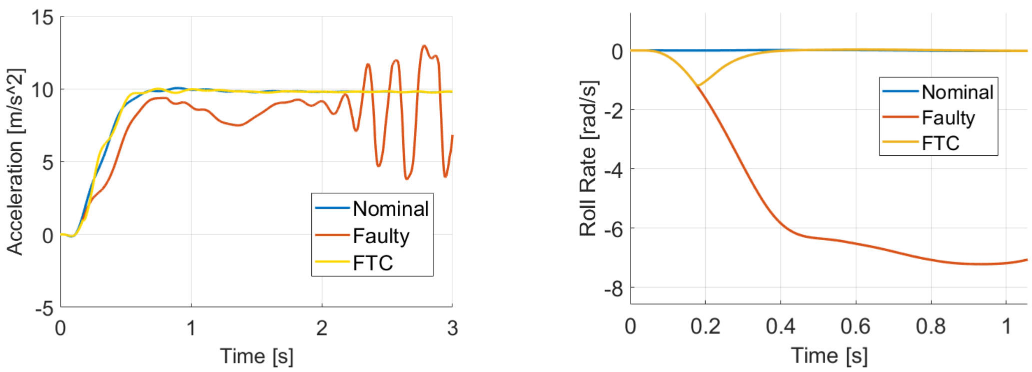


(a) Closed Loop Actuator Responses with Failed 3rd Actuator

(b) Closed Loop Actuator Responses with Failed 3rd Actuator and FTC is on

Figure 5.31 Comparison of Actuators for Faulty and FTC Cases at M=1.2 Altitude=1000m

Figure 5.31 (a) shows the actuator command and responses for faulty case without FTC. It can be clearly noticed that the angular position of third actuator does not track its corresponding command. When FTC is activated in Figure 5.31 (b) the elevator deflection is distributed to other actuators correspondingly.



(a) Acceleration Responses for Nominal, Faulty and FTC Cases

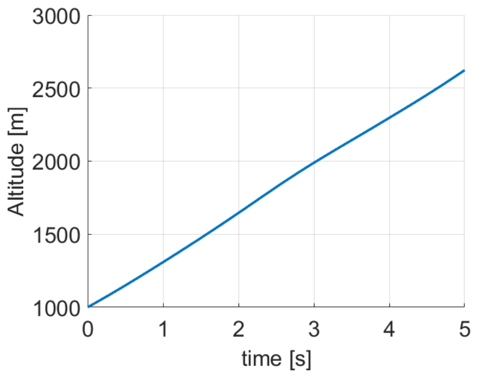
(b) Roll Angular Rates for Nominal, Faulty and FTC Cases

Figure 5.32 Comparison of Results for Nominal, Faulty and FTC Cases at M=1.2 Altitude=1000m

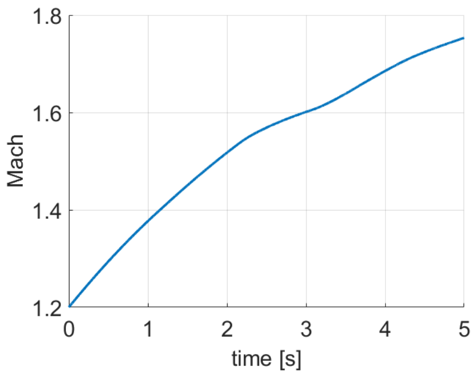
Figure 5.32 (a) shows that without FTC the transient response of faulty scenario is clearly worsened and even becomes unstable. This is because the wrong actuator distribution also causes undesired roll rate (Figure 5.32 (b)). This roll rate causes disturbance and reduces the performance of lateral controllers since the linearization contained zero roll rate assumption. Also the roll autopilot tries to overcome this roll rate but due to the failed actuator it generates more acceleration disturbance on pitch channel. When control mixer is reconfigured by FTC algorithm it can be seen that the roll rate rapidly reduces to zero.

5.4. Nonlinear Dynamic Simulation Scenario

The autopilot effectiveness can also be tested in a nonlinear dynamic simulation scenario[17]. In this section, nonlinear simulation results are presented for a dynamic scenario where mach and altitude change over time. The mach and altitude profiles are shown below.



(a) Altitude Change Profile for Simulation Scenario



(b) Mach Profile for Simulation Scenario

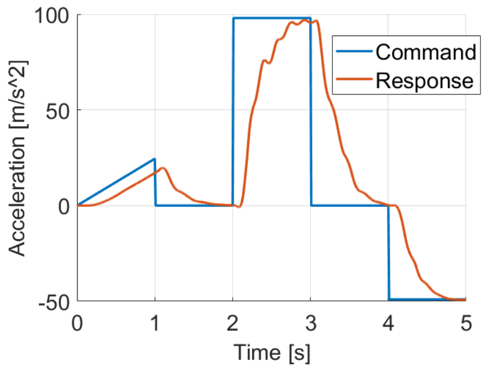
Figure 5.33 Altitude and Mach Profiles for Dynamic Nonlinear Simulation Scenario

While mach and altitude change over time, various ramp and step acceleration commands are given to pitch autopilot for nominal, faulty and active FTC cases. Two different actuator failure scenarios are studied.

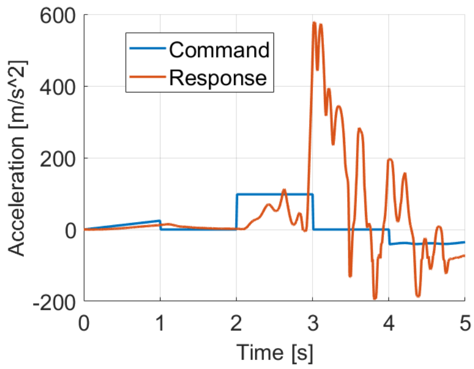
5.4.1. Scenario 1

In first scenario third actuator is stuck at initial zero angle and does not respond to given commands throughout the flight. Then same scenario is repeated with active FTC being on.

The fault diagnosis algorithm detects which actuator has failed and switches to corresponding reconfigured distribution. Acceleration responses for nominal, faulty and FTC cases are shown in figures below.

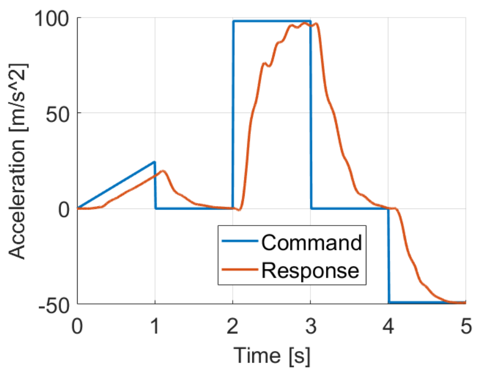


(a) Nominal Case - Acceleration Response for Dynamic Nonlinear Simulation

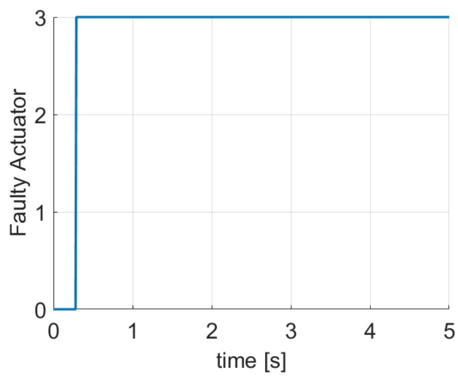


(b) Actuator Failure Case - Acceleration Response for Dynamic Nonlinear Simulation

Figure 5.34 Acceleration Responses for Nominal and Faulty Cases



(a) FTC Case - Acceleration Response for Dynamic Nonlinear Simulation



(b) FTC Case - Detection of Actuator Failure

Figure 5.35 Acceleration Responses for Nominal and Faulty Cases

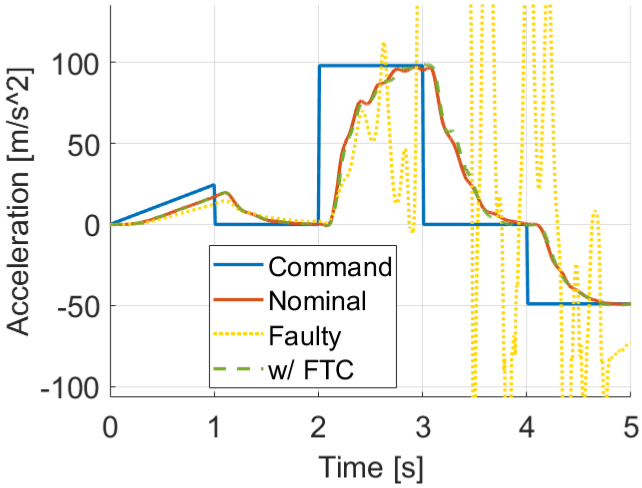
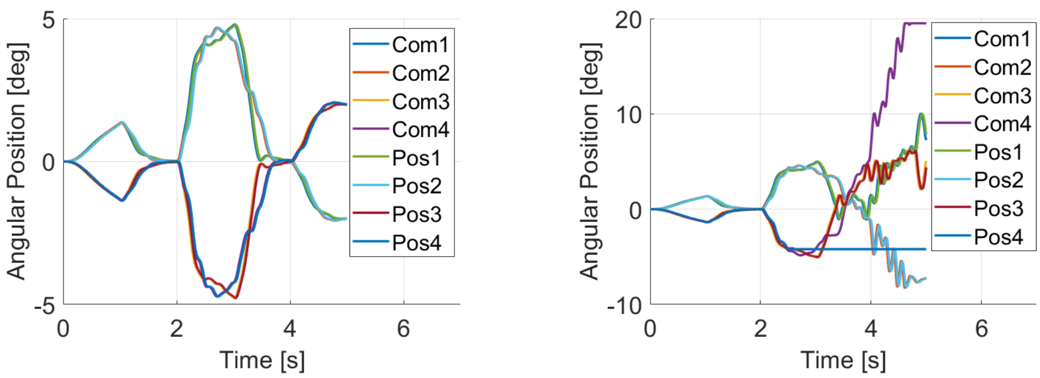


Figure 5.36 Acceleration Responses for Nominal and Faulty Cases

It can be seen that actuator failure can lead to instability of entire system without and active fault tolerance measure. When active FTC method is applied by detecting the fault and reconfiguring control commands the system function is restored.

5.4.2. Scenario 2

In second scenario, fourth actuator fails at t=2.5 seconds and stays stuck at that angle for the rest of the scenario. The figure below shows the actuator responses with and without failure.

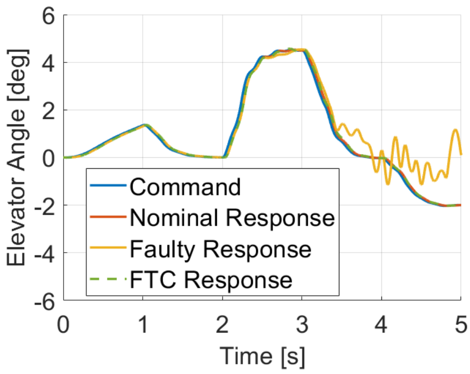


(a) Actuator Responses without Fault

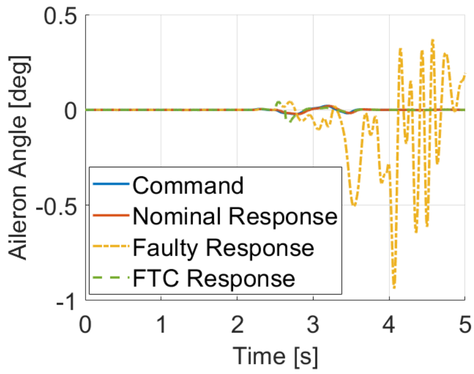
(b) Actuator Responses with Fault

Figure 5.37 Responses of Actuators for Nominal and Faulty Cases

The purpose of actuator movements is to provide desired elevator and aileron deflections. Figures below show that in case of actuator failure desired deflections are not reached.

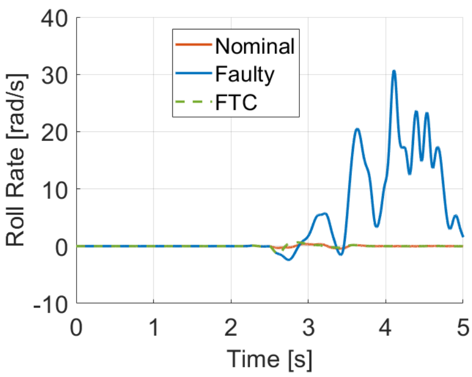


(a) Comparison of Elevator Angles

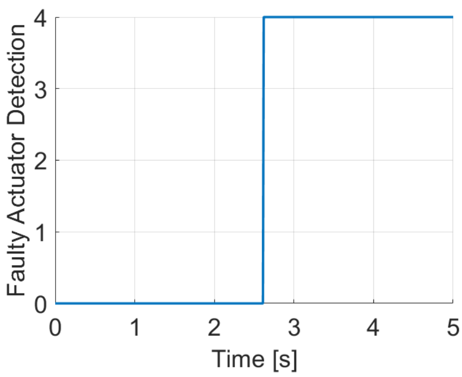


(b) Comparison of Aileron Angles

Figure 5.38 Elevator and Aileron Angles for Nominal, Faulty and FTC Cases



(a) Comparison of Roll Angular Rates



(b) Detection of Actuator Failure with active FTC

Figure 5.39 Roll Angular Rate and Faulty Actuator Detection

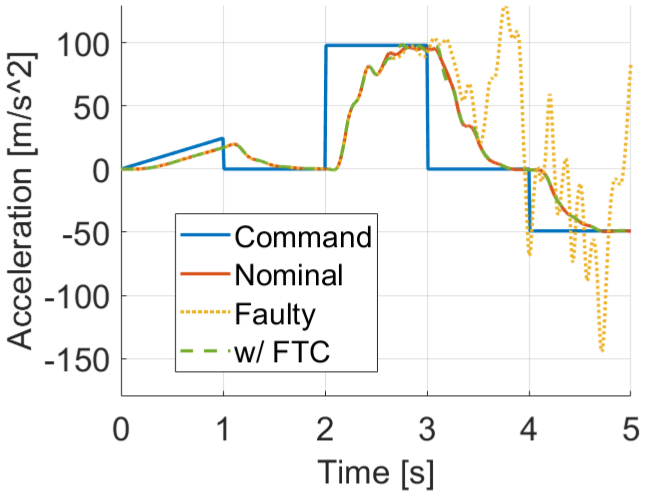


Figure 5.40 Acceleration Responses for Nominal, Faulty and FTC Cases

In above figures, it is shown that an actuator failure causes both roll and pitch autopilots to dysfunction. However the active FTC method in this study resolves this issue by reconfiguring control commands.

6. CONCLUSIONS AND FURTHER WORK

6.1. Conclusion

In this thesis study an autopilot is designed for a missile system with fault tolerant control and model predictive control. An introductory insight has been given on main concepts such as missile autopilot, model predictive control, fault tolerant control in Section 2 - Background. The differences and similarities of this thesis with other papers that investigated similar subjects have been compared in Section 3 - Related Works. Afterwards in Section 4, the methods used in various papers have been referenced while presenting the methods followed in this thesis study.

It has been discussed that the actuators are essential for missiles to make necessary control surface deflections so that the missile can complete its mission by doing required maneuvers. The system is normally over-actuated with four actuators and three motions to control (roll, pitch, yaw). The elevator, rudder and aileron commands are distributed to four actuators shown in section 4.4.2 . However, if one of the actuators fail and stop corresponding to given commands the incorrect transformation from ERA to actuators cause whole system to fail shown in results section 5.2, 5.3 and 5.4 . With remaining three actuators the system is properly-actuated which indicates that control in all three channels should still be possible. However without a proper fault tolerant method whole system is lost because of a single actuator dysfunction. A fault tolerant control is applied in this study that detects which actuator is faulty and reconfigure the control commands to remaining actuators appropriately. This way, advantage of being properly-actuated can be kept alive and still preserve control in roll, pitch and yaw channels.

The main controller of the autopilot system is designed with model predictive control theory. The autopilot is expected to operate at various speeds and altitudes. Depending on altitude, the air density changes and consequently difference in dynamic pressure affects the amount of aerodynamic force and moments generated. Also aerodynamic coefficients vary significantly depending on mach number of the missile. While the missile gains speed its mass and inertia also changes as rocket fuel is expelled. Therefore a linear adaptive MPC controller from MATLAB toolbox is used in order to design a controller that adapts to different system dynamics. Linearization of the system model is shown in section 4.2 . Sample time of the MPC controller is chosen as 0.01 seconds, prediction horizon is chosen as 50 steps and control horizon is 5 steps. The MPC controller output weights are tuned at various mach numbers and turned into lookup tables with mach input. Control input constraints are also defined according to physical limitations of actuators. When one of the actuator failure is detected, the control input constraint is also updated according to capacity of new actuator configuration.

A nonlinear analysis model is assembled using MATLAB/Simulink softwares. The model mainly consists of thrust, aerodynamics, flight dynamics and avionics models. The missile avionica contain IMU, CAS and the autopilot blocks. The autopilot is tested in various mach and altitudes (Table 5.1). At each analysis various errors and sensor delays were given as shown in Table 5.2.

The analysis results under these cases showed the robustness of the designed MPC controller to a range of system uncertainties and sensor delays. Next, the same analysis were repeated under various actuator failure scenarios. At 1000 meters altitude failure scenario of 1st actuator were shown at various flight conditions, at 2000 meters 2nd actuator and at 3000 meters 3rd actuator failure effects were shown. These effects mostly include great reduction in controller performance even including instability. The main reason can be simply pointed out that without the proper command reconfiguration, given actuator commands complete only 3/4 of required deflection. Therefore we see almost two times slower responses. At certain flight conditions system can become completely unstable because incorrect command distribution also disrupts the roll autopilot. When active fault tolerance method was applied by reconfiguring the control commands, the original response of the controller was shown to be recovered in following figures. Finally, two scenarios in a nonlinear simulation with varying mach and altitude were tested. In these scenarios the autopilot were given various ramp and step acceleration commands. In first scenario actuator no. 3 does not respond since t=0. In the second scenario, actuator no. 4 is locked at its last position at t=2.5 seconds. The same scenarios were repeated with fault tolerant control implementation. The results were compared in a single graph and it was noticed that the faulty case without FTC completely diverges while FTC provides recovery of stable acceleration tracking.

6.2. Recommendations for Further Work

The work that has been proposed in this thesis can be further improved in future by extending certain features. The linear model for the adaptive model predictive control can be further enlarged as an additional study by adding the effect coupling due to roll rate into the state-space representation. If the coupling of pitch and yaw channels due to roll rate can be correctly modeled and implemented this can make the lateral autopilots more resistant against roll rates.

The fault tolerance algorithm can also be further improved by covering two simultaneous actuator failures. If two actuators have failed the system becomes under-actuated which means certain loss of performance will be inevitable. In that case ability to make simultaneous pitch-yaw maneuvers while stabilizing roll rate will not be possible due to lack of actuators. In that case an active fault tolerant action that changes the maneuver configuration from skid-to-turn into bank-to-turn can be considered. In bank-to-turn configuration the missile follows a roll angle such that all required maneuver is aligned with a single axis. This way there would be two motions (roll-pitch) and two actuators which makes the system properly-actuated.

REFERENCES

[1] Tong, L., Zhenyu, J. (2019). Reconfigurable fault-tolerant control for supersonic missiles with actuator failures under actuation redundancy. *Chinese Journal of Aeronautics, 33*(1), 324-338. https://doi.org/10.1016/j.cja.2019.05.015

[2] Blanke, M., Staroswiecki, M., Wu Eva, N. (2001). Concepts and methods in fault tolerant control. *American Control Conference*. DOI: 10.1109/ACC.2001.946264

[3] Ryu, S.M., Won, D.Y., Lee, C.H., Tahk, M.J., (2011). Missile autopilot design for agile turn control during boost-phase. *International Journal of Aeronautical and Space Science, 12*(4) 365-370. DOI:10.5139/IJASS.2011.12.4.365.

[4] Blanke, M., Kinnaert, M., Lunze, J., Staroswiecki, M. (2006). *Diagnosis and Fault Tolerant Control* (2nd ed.). Springer.

[5] Liu, X., Liang, Y., Pan, Q., Zhang, H. (2006). On robust fault-tolerant control of missile control system. *IFAC Proceedings Volumes, 39*(13), p354-359.

[6] Louhuan, J., Mingjian, D., Jianjun, M. (2020). Deep learning based active fault-tolerant control for missile actuators. *2020 Chinese Automation Congress*. DOI: 10.1109/CAC51589.2020.9326477

[7] Fleeman, E.L. (2001). *Tactical Missile Design*. AIAA Education Series.

[8] Reichert, R.T. (1992). Dynamic scheduling of modern robust control autopilot designs for missiles. *IEEE Control Systems Magazine, 12*(5).

[9] Urban, T.J. (1989). *Synthesis of missile autopilots robust to the presence of parametric variations* [Master's thesis, Massachusetts Institute of Technology Aeronautics and Astronautics]

[10] Buschek, H. (2001). Design and flight test of a robust autopilot for the IRIS-T air-to-air missile. *IFAC Proceedings Volumes, 34*(15), 449-454.

[11] Gezer, R.B., Kutay, A.T. (2014). Robust model following control design for missile roll autopilot. *UKACC International Conference on Control* DOI: 10.1109/CONTROL.2014.6915107

[12] Mohammadi, M.R., Jegarkandi, M.F., Moarrefianpor, A. (2016). Robust roll autopilot design to reduce couplings of a tactical missile. *Aerospace Science and Technology, 51*, 142-150. https://doi.org/10.1016/j.ast.2016.02.001

[13] Kohli, C., Chandar, T.S. (2016). Design of Robust Roll Autopilot with Actuator Compensation. *IEEE Region 10 Conference*. DOI: 10.1109/TENCON.2016.7848736

[14] Talole, S.E., Godbole A.A., Kolhe J.P., Phadke, S.P. (2012). Robust roll autopilot design for tactical missiles. *Aerospace science and technology, 51*, 142-150. https://doi.org/10.2514/1.50555

[15] Mohamed, E., Yan, L. (2016). Autopilot design based on robust LQR approach. *Chinese Control and Decision Conference IEEE*. DOI: 10.1109/CCDC.2016.7531088

[16] Jackson, P., (1990). Applying μ-synthesis to missile autopilot design. *Proceedings of the 29th conference on decision and control*.

[17] Shen, Y., Yu, J., Luo, G., Mei, Y. (2017). Missile autopilot design based on robust LPV control. *Journal of systems engineering and electronics, 28*(3), 536-545. DOI: 10.21629/JSEE.2017.03.13

[18] Li, K., Yu, Y., Zhao, L. (2012). Three loop topologies scheme of spinning missiles with decoupling control. *International Conference on Automatic Control and Artificial Intelligence*, 2209-2212.

[19] Kırımlıoğlu, F. (2021). *Actuator fault tolerant terminal sliding mode guidance law with impact angle and acceleration constraints*[Master's thesis, Hacettepe University]. https://acikerisim.hacettepe.edu.tr/items/5bda6691-f4d9-4c73-8fa0-661681e6842e/full

[20] Wang, L. (2009). *Model Predictive Control System Design and Implementation Using MATLAB*. Springer.

[21] Etkin, B., Reid, L.D. (1996). *Dynamics of Flight Stability and Control.* (Third ed.). JOHN WILEY & SONS, INC.

[22] Fawzy, M., Aboelela, M.A.S., Abd El Rahman, O., Dorrah, H.T. (2011). Design of missile control system using model predictive control. *The Online Journal on Computer Science and Information Technology, 1*(3).

[23] Asfihani, T., Mufidah, M.K., Subchan, S., Adzkiya, D. (2019). Missile control design for moving target using model predictive control. *Journal of Physics: Conference Series 1490*. doi:10.1088/1742-6596/1490/1/012069

[24] Schwenzer, M., Ay, M., Bergs, T., Abel, D. (2021). Review on model predictive control: an engineering perspective. *The International Journal of Advanced Manufacturing Technology, 117*, 1327–1349. https://doi.org/10.1007/s00170-021-07682-3

[25] Menon, P.K., Ohlmeyer, E.J. (2001). Integrated design of agile missile guidance and autopilot systems. *Control Engineering Practice, 9*(10), 1095-1106. https://doi.org/10.1016/S0967-0661(01)00082-X

[26] Naidu, B.N., Kumar, K.A., Priyanka, K.S., Govardhan, D. (2021). Design of lateral acceleration control autopilots for missile Systems. *International Journal of Engineering Research and Technology, 10*(6). ISSN:2278-0181.

[27] Abhishek, D. (2021). *Adaptive model predictive control for uncertain LTI systems with state and input constraints*[Doctoral dissertation, INDIAN INSTITUTE OF TECHNOLOGY DELHI].

[28] Bajpai, G., Chang, B.C., Lau, A. (2000). Reconfiguration of flight control systems for actuator failures. *19th Digital Avionics Systems Conference*. DOI: 10.1109/DASC.2000.886932

[29] Liu, G., Wang, D., Li, Y. (2004). Active fault tolerant control with actuation reconfiguration. *IEEE Transactions on Aerospace and Electronic Systems, 40*(3). DOI: 10.1109/TAES.2004.1337485

[30] Zong, G., Yang, D., Lam, J., Song, X. (2021). Fault-Tolerant Control of Switched LPV Systems: A Bumpless Transfer Approach. *IEEE transactions on Mechatronics*.

[31] Wang, W., Wen, C. (2010). Adaptive actuator failure compensation control of uncertain nonlinear systems with guaranteed transient performance. *Automatica, 46*(12), 2082-2091. https://doi.org/10.1016/j.automatica.2010.09.006.

[32] Boskovic, C., Mehra, R. (2002). Multiple-Model Adaptive Flight Control Scheme for Accommodation of Actuator Failures. *Journal of Guidance, Control and Dynamics*. DOI: 10.2514/2.4938

[33] Zaid, F. A., et al. (1991). Accommodation of Failures In The F-16 Aircraft Using Adaptive Control. *IEEE Control Systems Magazine, 11*,(1), 73-78. DOI:10.1109/37.103360

[34] Moerder, D. D., Halyo, N., Broussard, J.R., Caglayan, A.K. (1989). Application of precomputed control laws in a reconfigurable aircraftflight control system. *Journal of Guidance, Control and Dynamics, 12* (3). https://doi.org/10.2514/3.20412

[35] Bennani, S., Sluis, R., Schram, G., Mulder, J. (1999). Control law reconfiguration using robust linear parameter varying control. *Guidance, Navigation and Control Conference and Exhibit*. https://doi.org/10.2514/6.1999-4137