İSTANBUL TECHNICAL UNIVERSITY ★ INSTITUTE OF SCIENCE AND TECHNOLOGY

DESIGN OF A MULTIMODAL PLANNING AND CONTROL FRAMEWORK FOR AGILE MANEUVERING UCAVs

M.Sc. Thesis by
Nazım Kemal ÜRE

Department : Defence Technologies

Programme : Defence Technologies

JUNE 2010

İSTANBUL TECHNICAL UNIVERSITY ★ INSTITUTE OF SCIENCE AND TECHNOLOGY

DESIGN OF A MULTIMODAL PLANNING AND CONTROL FRAMEWORK FOR AGILE MANEUVERING UCAVs

M.Sc. Thesis by
Nazım Kemal Üre
(514081007)

Date of submission : **07 May 2010**
Date of defence examination: **09 June 2010**

Supervisor (Chairman) : **Assoc. Prof. Dr. Gökhan İNALHAN (ITU)**
Members of the Examining Committee : **Prof. Dr. Çingiz HACIYEV (ITU)**
Assoc. Prof. Dr. M. Turan SÖYLEMEZ (ITU)

JUNE 2010

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

İNSANSIZ SAVAŞ UÇAKLARI İÇİN AGRESİF MANEVRA PLANLAMA VE KONTROL SİSTEMLERİNİN GELİŞTİRİLMESİ

YÜKSEK LİSANS TEZİ

Nazım Kemal ÜRE
(514081007)

Tezin Enstitüye Verildiği Tarih : 07 Mayıs 2010
Tezin Savunulduğu Tarih : 09 Haziran 2010

Tez Danışmanı : Doç. Dr. Gökhan İNALHAN (İTÜ)
Diğer Jüri Üyeleri : Prof. Dr. Çingiz HACIYEV (İTÜ)
Doç. Dr. M. Turan SÖYLEMEZ (İTÜ)

HAZİRAN 2010

FOREWORD

First of all, I would like to express my gratitude to my advisor Prof. Gökhan İNALHAN for providing me with the research environment and vision to conduct this research. Without his inputs, this work would never be complete. I cannot thank him enough for influencing me to pursue an academic career.

During my life as a research assistant at Controls and Avionics Lab. Emre KOYUNCU had always been a solid colleague and a decent friend, I would like to send him my special thanks for his support. I would also like to thank Oktay ARSLAN, Melahat CİHAN and Melih FİDANOĞLU; their friendship eased the pain of long working hours at the laboratory and made it a joyful place to work.

This part would be incomplete without mentioning my family. My parents, Mehmet Celal ÜRE and Dilek ÜRE made a lot of self sacrifices for me to provide a comfortable life to me as a student. My brother Cihan Deniz ÜRE had also been a big support and a source of joy for me as well as our little sister Queen. Words are simply not enough to give them proper credits.

June 2010

Nazım Kemal Üre

Aeronautical and Astronautical Engineer

TABLE OF CONTENTS

Page

ABBREVIATIONS

ACM	**:** Air Combat Maneuvering;
BFM	**:** Basic Fighter Maneuvers
DOF	**:** Degree of Freedom
FL	**:** Feedback Linearization
HOSM	**:** Higher Order Sliding Mode
MBM	**:** Mode Based Maneuvering
MP	**:** Minimum Phase
NMP	**:** Non-Minimum Phase
SMC	**:** Sliding Mode Control
SEAD	**:** Suppression of Enemy Air Defenses
UCAV	**:** Unmanned Combat Aerial Vehicle
VSS	**:** Variable Structure Systems

LIST OF TABLES

Page

LIST OF FIGURES

Page

DESIGN OF A MULTIMODAL CONTROL FRAMEWORK FOR AGILE MANEUVERING UCAVs

SUMMARY

Generation and control of agile maneuver profiles, began to study heavily especially in the last decade due to fact that most Unmanned Combat Air Vehicles (UCAVs) are in threat for limited maneuvering capability in complex battlefield scenarios. In this sense, agility is referred as being able to execute controllable maneuvers under high “g” forces on complex flight trajectories, very much like piloted fighter aircrafts do. It is of critical importance to develop autonomous UCAV systems, which can plan and execute agile maneuvers in full flight envelope in order to avoid the threats in battlefield. In this study, focus will be on planning and control of agile maneuvers.

The ability of unmanned combat air vehicles (UCAV) to autonomously design and execute agile maneuvers in complex and dynamic environments is a low-level enabling technology for future air scenarios driven by performance and safety goals. In this work, we present a multi-modal framework which spans the full flight envelope dynamics of the aircraft. Maneuver decomposition methodology of the developed framework, reduces complexity of the planning and control problems for UCAVs operating in dense and threatening environments.

Multi Modal control framework basically consists of decomposing the arbitrary maneuvers into set of maneuver modes and associated maneuver parameters. Main advantage of framework is allowance of the feasibility and control problems to be treated for each sub-maneuver individually instead of whole maneuver, thus reducing the complexity of the problem (modal sequence has strictly lower dimension than state space description) and control part was relaxed by designing specific controllers for each mode and switch between them, in order track maneuver mode sequence instead of designing a single controller for maneuver tracking over full flight envelope.

One of the major applications of Multi Modal Control Framework is to integrate it with a path planner within hierarchical control architecture. In general approach, outputs of path planner layer (planned trajectory) is directly fed to the low level control layer, which usually results in improper and unfeasible maneuvers for agile maneuvering aircraft. Methodology of Multi Model Control Framework can be inserted between path planning and control layer as a “Maneuver Planning Layer”, where trajectory is decomposed into maneuver modes and modal inputs are selected within feasible limits. Since there is a stable controller assigned to each mode, low level control is achieved through switching between them. Overall, this extra planning layer re-defines planned trajectory in terms of agile aircraft motion instead of directly forcing aircraft to track the trajectory.

Main contribution of this thesis is a new perspective on maneuver/motion planning algorithms, which doesn't require any pre-build maneuver libraries and relies on the parameterized modal decomposition of arbitrary flight maneuvers. This planning method can be used on any flight trajectory over full flight envelope and due to fact that algorithms take in to account of envelope and continuity constraints on maneuvers; once a maneuver profile is created it is guaranteed that is trackable by the switched control system based on the nonlinear higher order sliding mode control, since every mode is locally controllable by a nonlinear controller. Proposed framework is tested on simulations of full scale aircraft models for various agile maneuvers taken from fighter combat examples and maneuvering in dense environment scenarios.

İNSANSIZ SAVAŞ UÇAKLARI İÇİN AGRESİF MANEVRA PLANLAMA VE KONTROL SİSTEMLERİNİN GELİŞTİRİLMESİ

ÖZET

İnsansız Savaş Uçaklarının (İSU) savaş alanında sürekli tehdit altında çalışmasından dolayı, son yıllarda agresif manevra profillerinin üretilmesi ve kontrol edilmesi ilgi çekici bir konu olmaya başlamıştır. Burada agresiflik, pilotlu savaş uçaklarının yaptığına benzer şekilde yüksek "g" kuvvetine maruz kalan yüksek açısal hızlı manevralar ile tanınmlanmıştır. Böyle bir sistemin otonom olarak geliştirilmesi savaş alanlarındaki İSU kayıplarını hızını olumlu yönde etkileyecektir. Bu tezin başlıca amacı bu otonom sistemin tasarımıdır.

Bu çalışmada, İSU'ların otonom şekilde manevralarını planlayabilmesi ve takip edebilmesi için mod temelli bir kontrol sistemi önerilmiştir. Bu system uçağın bütün uçuş zarfındaki manevraları ele almak, onları parçalara bölmek ve her bir manevra parçasının kendine has kontrol problemini çözerek eldeki orijinal problemin derecesini ve karmaşıklığını düşürmeyi amaçlamaktadır.

Bu çok modlu sistemin başlıca avantajı, karmaşık manevraları, geliştirilen bir algortima yardımı ile parçalara ayırması, her bir manevra parçası için uçağın uçuş zarfı içinde kalan manevra parametrelerini sentezlemesi ve gerekli manevra parçası dizisini oluşturmasıdır. Her bir manevra parçası için o parçaya has bir kontrolcü tasarımı gerçekleştirilebildiği için, bu manevra parçaları için bir kontrolcü ailesi geliştirildiğinde ve bunlar arasında değişimler yapıldığında, bütün manevranın kontrolü elde edilecektir.

Geliştirilen sistemin bir başka avantajı ise rota/yörünge planlayıcılar ile tümleştirilebilme özelliği olmasıdır. Rota planlayıcılar uçağın açısal hareketini göz önüne almadan rota sentezledikleri için, bu sentezlenen rotalar direkt olarak alçak seviye kontrol sistemine girildiklerinde uçağın kararsızlığa gitmesine sebep olabilir. Fakat çok modlu sistem anlayışı ile yerleştirilmiş bir manevra planlama bölümü, rota planlayıcısından aldığı rotalar üzerinden manevraları tanıyarak uçağın açısal durumunuda tanımlayabilir, böylece bir önceki paragraftaki değişimli kontrolcüler kullanılarak kararlı manevra takip etme sağlanabilir.

Bu tezde ilk olarak çok modlu sistemin modlarının listesi ve parametreleri oluşturulmuştur, daha sonra standart uçak manevralarını modlara ayıran bir algortima, ile yörünge planlayıcıdan aldığı rotayı modlara ayıran bir algoritma tasarlanmıştır. Son olarak Yüksek Dereceden Kayma Kipli kontrol ilkelerini göz önüne alarak doğrusal olmayan değişimli bir kontrolcü ailesi tasarlanmış, ve bütün sistem Savaş Uçağı Manevraları, Akrobatik Manevralar ve Dar Geçitlerde Navigasyon problemleri üzerinde test edilmiştir.

1. INTRODUCTION

Today, demands on agile maneuvering from unmanned air vehicles increases due to complexity of the operations and loss of aircrafts resulting from their incapability to handle aggressive evasive maneuvers. UCAVs are expected to operate in dense and often threatening environments, which need collision free trajectories to fly on. Such trajectories in complex environments will need to make use of agile maneuvering capability of aircraft (such as high “g” turns) and its full envelope flight characteristics (such as high angle of attack flight). Therefore, to develop an autonomous system it is critical to synthesize controllers for nonlinear aircraft dynamics, as well as designing motion planning algorithms which takes account of flight envelope and actuator limits of the aircraft.

Main aim of this thesis is to approach this problem in a multi-modal manner (decomposing the dynamics of aircraft into distinct flight modes) to develop a motion planning algorithm and a switching control methodology over flight modes to gain control over full flight envelope maneuvering dynamics of the aircraft.

In this first chapter of the thesis, main background on UCAVs and motivation of the research is presented. This chapter also contains literature survey and research summary on this subject. Final section of the chapter outlines the structure of the thesis as well as objectives and results of each chapter.

1.1 Background and Motivation

1.1.1 Unmanned combat aerial vehicles

Unmanned Combat Aerial vehicles (UCAVs) are natural extension of the classical Unmanned Aerial Vehicle (UAV) concept to military applications. Since unmanned vehicles have superior advantages over the manned ones in the battlefield such as; low cost and not having risk for loss of human life, they are usually one of the fundamental elements of the future command and control models. On the military basis, a UCAV can be described as aircraft which can launch, attack and recover without crew members abroad.

A recent paper on potential benefits of the UCAVs, points out that employment of unmanned vehicles with direct attack munitions could radically reduce the cost per target kill [1]. For example an Air Force B-52 shoots 80-90 conventional air launched cruise missiles (CALCM) which weighs around 2000 lbs and costs 1,160,000 $ per unit. On the other hand a UCAV can deliver the same lethality with far less costs by launching joint direct attack munitions (JDAM) guided by Global Positioning System (GPS). Same paper also asserts advantages of UCAVs on aircraft design due to reduced risk factor of human life leads to reduced margins in safety factors which relaxes the design process.

One of the first successful military applications of an unmanned system was achieved by MQ-1 Predator UAS in 2002 [2]. However this system was not fully automated; aircraft still received remote inputs from pilot during operation.

One of the most actively developing projects on UCAVs is the J-UCAS (Joint Unmanned Combat Air Systems) conducted by DARPA and Air Force Navy [3]. This project aims to develop a system which fully exhibits the advantages of UCAV fleets, possessing interoperability, autonomy and platform flexibility spanning military applications such as Suppression of Enemy Air Defenses (SEAD) and Electronic Attack (EA). A 21st Century Battlefield envisioned by this program is displayed in Figure 1.1.

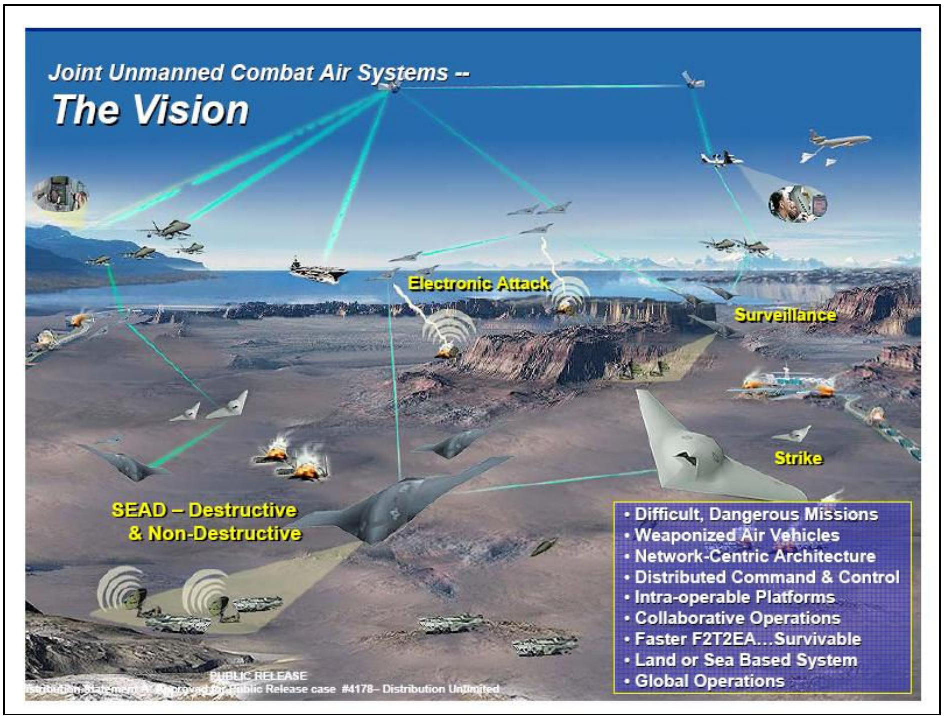


Figure 1.1 : Vision of a modern UCAV mission [3].

1.1.2 Threats on UCAVs on battlefield

A UCAV operates usually in and adversary environment which are full of enemy vehicles aiming to eliminate it. For example, UCAV can be sensed by radars in the field and can be brought down by a Surface to Air Missile (SAM) and Anti Aircraft Artillery (AAA). It is also very probable for it to be gunned down by an enemy manned aircraft [4]. Since autonomous vehicles does not possess the same agility and decision methodology of a manned vehicle, so it cannot perform necessary maneuvers to evade these attacks, which poses the most serious problem in implementation of UCAVs on realistic battlefield scenarios.

A study conducted by Department Science Board asserts that this issue rises from the fact that current UCAV technology does not support stealth and active countermeasure technology [5]. In the same study, a table which compares the mishap rates of UCAVs and manned aircraft per 100,000 flight hours is also provided. This data is displayed on Table 1.1.

Table 1.1 : Mishap rates per 100,000 flight hours [5].

UAV Mishaps	Aircraft Mishaps
Predator - 32*	F-16 -3
Pioneer - 334*	General Aviation - 1
Hunter - 55*	Regional Commuter - 0.1
* much less than 100,000 flight hours	Large Airliners - 0.01

1.1.3 Problem statement

Under the light of the discussion on the preceding subsection, our study aims to develop a planning and control framework in which provide required autonomy for aircraft to perform agile maneuvers to operate safely in adversary environments thus reducing the mishap rates.

On technical context, this problems in treated as a navigation and control problem in a dense 3D environment as displayed in Figure 1.2. Note that obstacles in Figure 1.2 is not “hard” obstacles like mountains, they are representing “forbidden zones” such as sensing area of a radar, or range of an AAA system. Thus a trajectory which is collision free in this environment corresponds to a safe trajectory in the battlefield. Since the environment is very dense, aircraft must execute agile maneuvers to evade these obstacles/to track the prescribed safe trajectory.

On the planning side, trajectory or path planning in a dense environment is complicated problem on its own and is out of scope of thesis. Reader may consult path planning strategies developed by Koyuncu, specifically in this framework in [6].

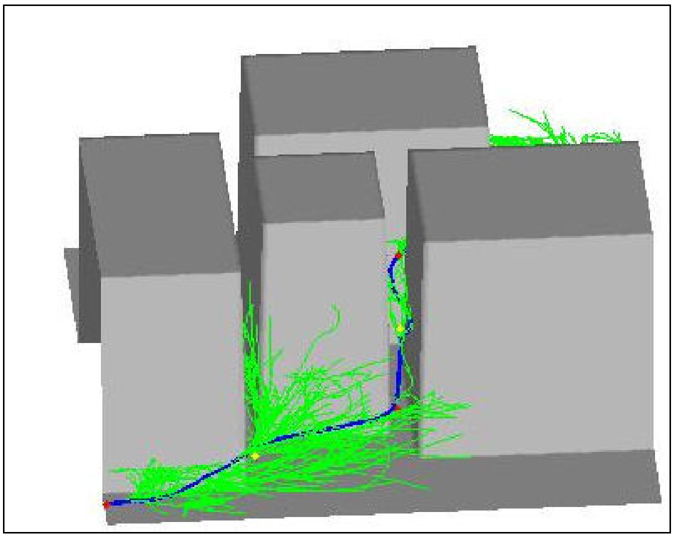


Figure 1.2 : Navigation in a dense 3D environment.

However path planning alone is insufficient to cover this problem, due to fact that aircraft maneuvers possess complex attitude dynamics along the path. Therefore one of the main aims of this thesis is to develop an appropriate motion language for expressing and planning agile maneuvers in this framework.

On the controller design part, aim of the thesis is to create low level feedback controllers capable of tracking the maneuver profiles generated by the planning layer while taking into account of nonlinear dynamics of the aircraft, rejecting the disturbances from the environment and being robust to uncertainty in the aircraft dynamics. A typical “Agile Maneuver” is displayed in Figure 1.3, to display the complexity of the problem and the fact that control strategy of agile maneuvering must strictly differ from conventional control problem of trimmed maneuvers, such as discussed in books on flight stability and control.

The problem stated above is approached by developing modal decomposition of aircraft dynamics and nonlinear control methods in the thesis. In the next section literature on this subject is reviewed to gather knowledge on current state of the art.

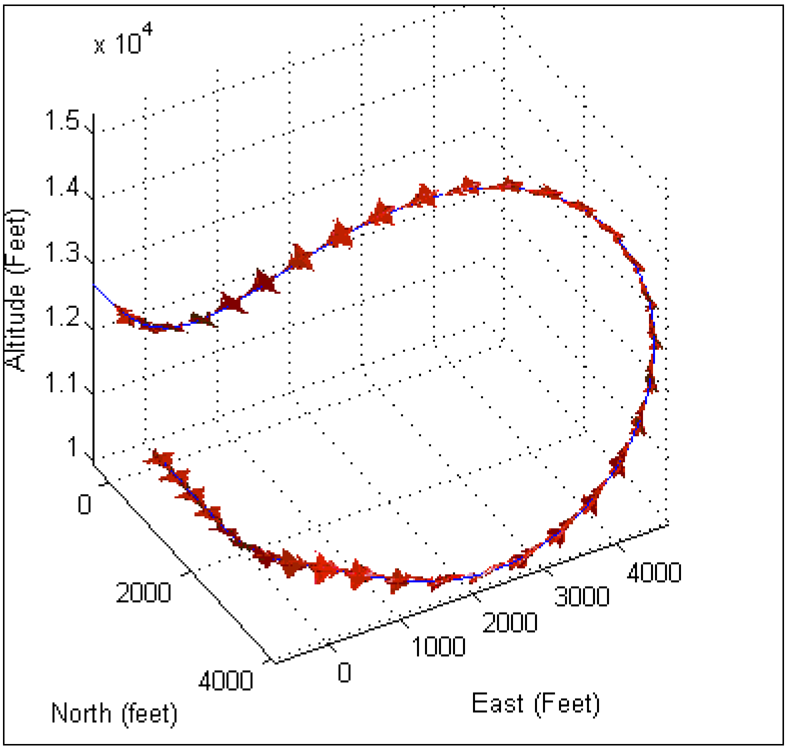


Figure 1.3 : Example agile maneuver.

1.2 Literature Survey

As it is asserted in the problem statement, approach of thesis involves decomposing the full flight envelop dynamics into distinct flight modes with individual dynamics and state constraints. Since nonlinear dynamics of agile aircraft strongly suggests that this problem must inspected under the area of nonlinear control, a separate survey is conducted on this subject on the second part of the survey.

1.2.1 Modelling the aircraft dynamics as a hybrid system

Motion planning problem for aerospace vehicles are complicated by the fact that, planners based on optimal performance begin to fail in means of computation, when one takes into account of constraints related with dynamical equations of aircraft.

Due to fact that, aircrafts state space is at least 12 dimensional, input-state search becomes too complicated; therefore such planners are only successful for vehicles with small state space dimensions [7]. To reduce the complexity of this problem, motion description languages and quantized control concept have been adopted into motion planning [8].

Motion description languages, makes use of classified combination of simplified control laws to track generalized outputs. Most of these languages are strongly connected with the concept of hybrid systems, which in general, classifies the motion by using discrete states which switches in between according to input and state information and each discrete state having its own continuous dynamics. A subclass of such languages which is based on classification of behaviour (or reaction) of the dynamic systems, has been successfully adapted for non-holonomic robotic systems [8,9]. More recently, closed loop hybrid control systems were developed based on linear temporal logic for the same purpose by [10]. For aerospace vehicles, a hybrid model for aircraft traffic management was developed in [11]. Study showed that, hybrid system representation gives opportunity to calculate reachable sets of the system and design hybrid control laws to drive the system to safe states [12].

Frazzoli suggested a maneuver automaton, which uses a number of feasible system trajectories to represent the building blocks of the motion plan of the aircraft, and a trajectory based (based on maneuver regulation principle) control system which asymptotically regulates the actual trajectory to the trajectory generated by maneuver automaton [7]. However, motion plans and controllable trajectories are restricted to the library of the maneuver automaton. Such libraries can be built by using interpolation between feasible trajectories [13]. Another study extended this system for online planning of feasible trajectories in partially unknown environments by using receding horizon iterations [14].

Description of aircraft dynamics from hybrid system point of view has been studied previously in [15-17] .These works have been successful in using the advantages of hybrid system methodology in control of both single and multiple aircrafts. However, these approaches did not include the full flight envelope dynamics of the aircraft. Specifically, both mode selection and controller design is strictly based on selected maneuvers; therefore controllability is limited [15-17] to these predefined trajectories. In our work, we make use of parameterized sub maneuvers which builds up complex maneuver sequences. We show that it is possible to cover almost any arbitrary maneuver and the entire flight envelope by this approach.

1.2.2 Nonlinear flight control

Two of the applicable nonlinear control techniques are feedback linearization, and sliding mode control [18-20]. Unfortunately, these techniques cannot be applied directly to the trajectory control of aircraft due to non-minimum phase nature of the controlled outputs resulting in unstable internal dynamics.

Early works at this area neglected the non-minimum phase (NMP) effect on the control system design, but included it in simulations to show that its effect was negligible [21,22]. However, during aggressive maneuvering NMP effect becomes significant due to high dynamic pressure and high angles of attack. An alternative way to overcome NMP effect is to perform a stable inversion technique. This is discussed in [23-25] for both conventional and vertical take-off and landing aircraft models. However, the stable inversion technique limits the full control bandwidth of the aircraft and diminishes the maneuvering capability to attain stability.

In our approach, we have followed an outer loop sliding mode control design to overcome non-minimum phase effect on the system. Sliding mode controllers offers insensitivity (robustness) to matched disturbances with known upper bounds, and they can be integrated with a dynamic compensator to overcome NMP effect as shown on [26]. Main drawback of the sliding control methods is the resulting chattering in the actuator signals due to discontinuous terms in the control laws. This can be problematic for the operation of the actuators specifically when the aircraft is exposed to high gain noises.

Boundary layer solution offers a low pass filter in the neighbourhood of sliding manifold to soften the effect of chattering [27]. A more elegant solution to the problem is through higher order sliding modes, which moves the discontinuous term into higher derivatives of the control [28]. For this purpose, several algorithms were developed by A. Levant, and these algorithms were applied to a real world aircraft pitch control problem [29]. We make use of the higher order sliding mode control laws at inner loop to provide continuous signals to controllers and increase accuracy of the controller.

1.3 Objectives, Challenges and Solutions

In these section main objectives of the thesis as wells challenges and contributions on solutions is briefly presented. Results are more carefully underlined in conclusion chapter.

First Objective is development of a mode based motion language for agile flight. Challenges are listed below;

- Arbitrariness of description in three dimensional aircraft trajectories

- Coverage of full flight envelope aircraft dynamics

- Expressing domains of modes and mode switching conditions

Our solution is; a mode based maneuvering automaton (displayed in Figure 1.4) driven by maneuver parameters (modal inputs) decomposing large and complex maneuvers into smaller segments characterized by state constraints and control objectives. This framework was initially published by authors in [30, 31].

Second objective is to develop A Manuever Generation Algorithm for navigation in dense environments. List of challenges are:

- Mode Transition Constraints on maneuver mode sequencing

- Maneuver parameter selection in flight envelope limits

- Degrees of freedom in maneuver sequence

Our solution is, a maneuver generation algorithm working on input trajectory sent by path planner, which incorporates the mode transition conditions and flight envelope limits drawn by agility metrics to compute the feasible mode sequence (displayed in Figure 1.5). This algorithm was first published by authors in [32]. Integration with a path planner was more deeply studied in [33].

Third objective is the design of a low level controller for tracking agile maneuver profiles. Challenges are;

- Nonlinear Aircraft Dynamics

- Uncertainties in Aerodynamic Coefficients and External Disturbances

- Compability with modular structure

Our solution is the, set of tow-loop nonlinear sliding mode controllers distributed through modes and switched in maneuver mode sequence to gain control over full maneuver profile. Proposed architecture is displayed in Figure 1.6.This part of the research was published in [34].

Structure of the thesis is as follows; second chapter demonstrated the maneuver decomposition methodology and lists of modes and modal inputs of the framework. Next chapter explains the maneuver identification and a generation algorithm. A nonlinear F-16 fighter aircraft model used in simulations and control design is provided in chapter 4. Fifth chapter is on the switching nonlinear controller design and final chapter ends the thesis with discussion of results and future research.

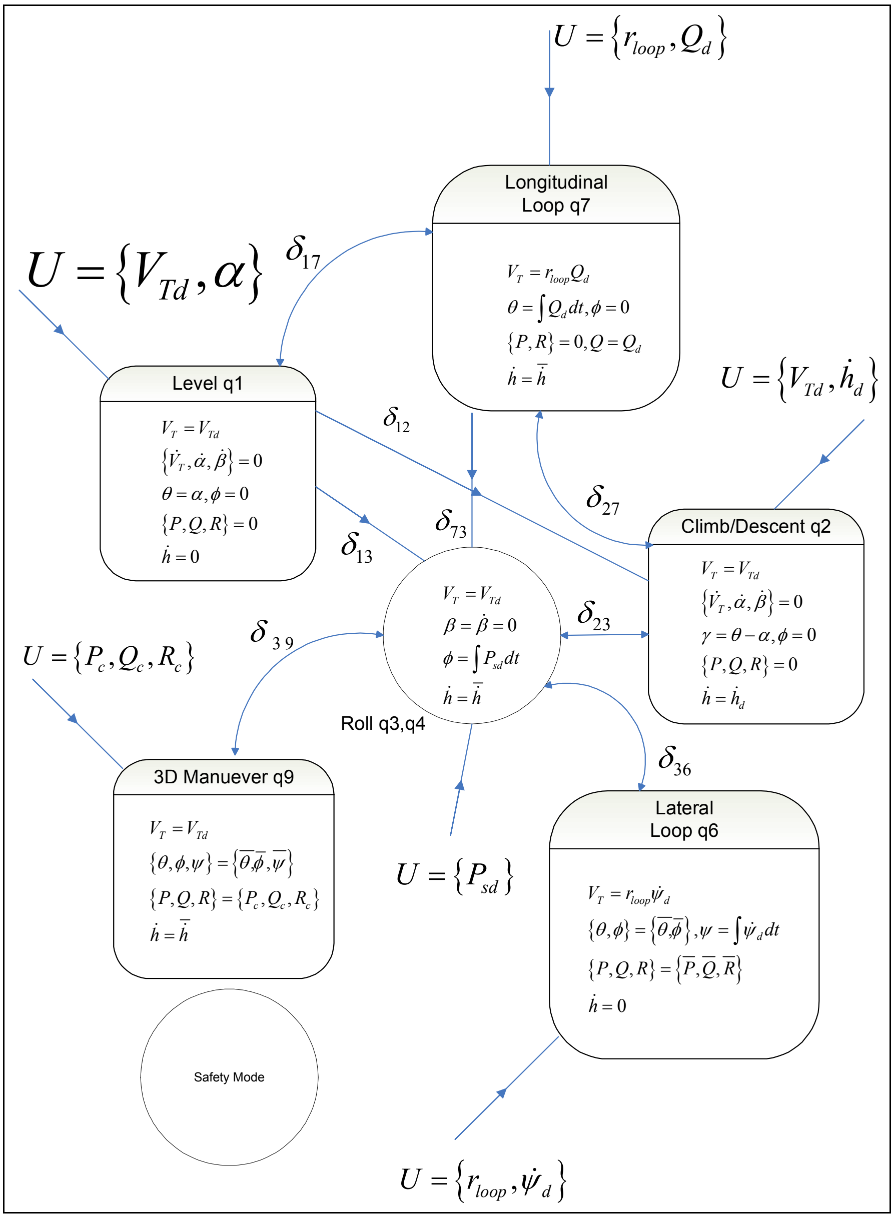


Figure 1.4 : Mode based maneuvering automaton.

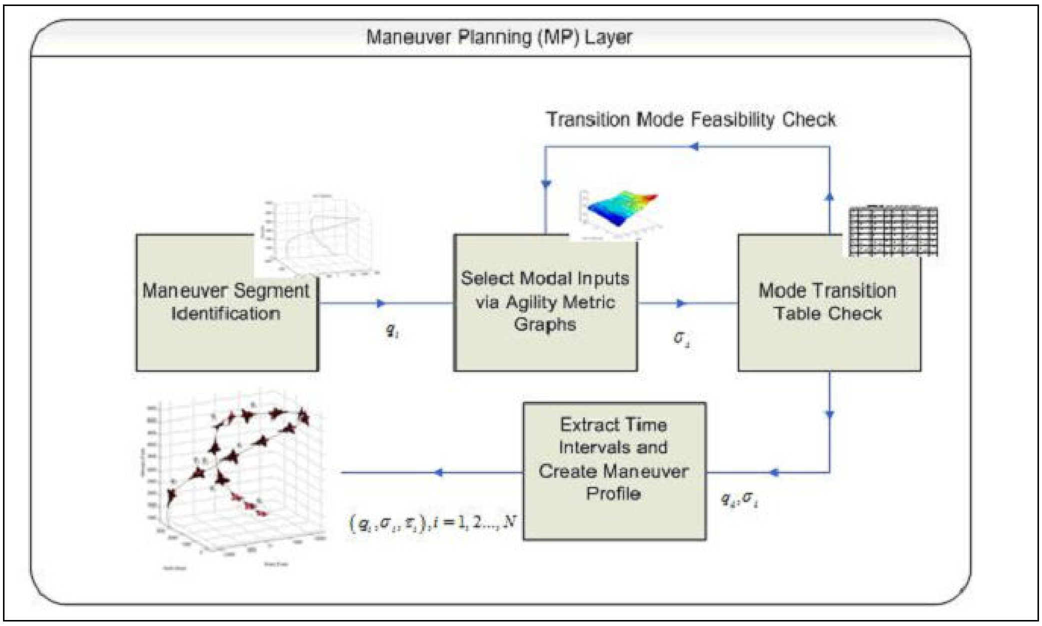


Figure 1.5 : Overview of maneuver generation algorithm.

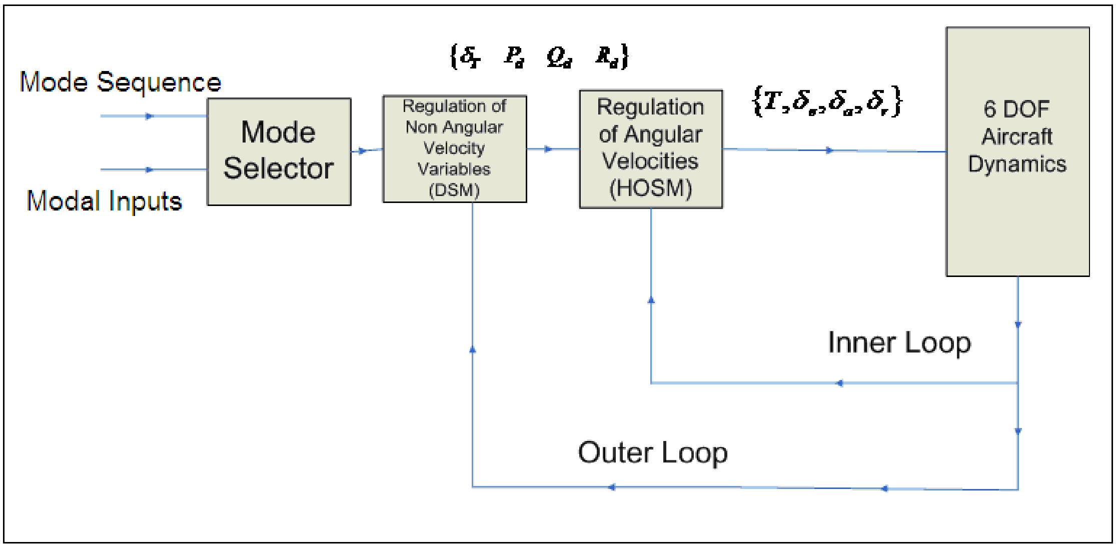


Figure 1.6 : Switched control system diagram.

2. MANEUVER DECOMPOSITION METHODOLOGY

This chapter is devoted to develop a high level motion description language for agile maneuvering aerial vehicles. As it is already stated in introduction, main approach of this thesis to this problem is to develop a maneuver decomposition methodology, which replaces the conventional state space description of maneuver with sequences of maneuver modes and associated parameters.

Chapter starts with inspection of classes of aerobatics and combat maneuvers to gain insight the physics of agile flight and mode decomposition. In the next section complete list of maneuver modes, associated maneuver parameters (modal inputs). On the final section of the chapter shows how the proposed framework can be modelled in terms of hybrid dynamical systems.

2.1 Aerobatics and Combat Maneuvers

In this section aerobatics and combat maneuvers are briefly described and inspected to aid our quest to develop a flexible motion language for agile flight. Both types of maneuvers are examined from modular point of view; that is if they can be represented as a combination of elementary maneuvers. Then human perspectives for flying these maneuvers are also given, such as how to enter a specific maneuver and how to make a smooth transition between two maneuvers. These examinations will shape the ideas for building parameterized maneuver library in the next section.

2.1.1 Aerobatics

Aerobatics (which is simply combination of the word “acrobatics” and “aeronautics”) is a very popular form of entertainment, where the aircraft is flown through unusual attitudes through combination of circular trajectories (loops) and lines. More information on aerobatics can be found on [35-37].

Although aerobatics is usually referred as performance stunts of an airplane, most of the maneuvers are directly influenced by evasion maneuvers performed by World War II fighter aircraft pilots. Since aerobatic flight spans a wide selection of the possible aircraft maneuvers and usually performed at the edge of the flight envelope, it is of great interest to inspect the nature and variety of aerobatic maneuvers for developing a maneuver library.

Some of the popular aerobatics maneuvers (which are frequently encountered in both airshows and competitions) are shown in Figure 2.1, Figure 2.2 and Figure 2.3, in terms of trajectory-attitude plots. The first maneuver "Slow Roll" is an elementary aerobatics maneuver where the aircraft is rolled through 360 degrees around longitudinal axis while keeping the path of the flight as straight as possible, the second maneuver "Split S" combines a half loop in the longitudinal plane (where the aircraft is inverted as a results of the loop) with a slow roll to regulate aircraft back into straight wing level flight, thus as a result of the maneuver aircraft gains altitude and its direction is reversed. Final maneuver in the figure is called "Half Cuban Eight", which combines a ¾ loop with a slow roll and descent to reverse the direction of the aircraft without loss in the altitude.

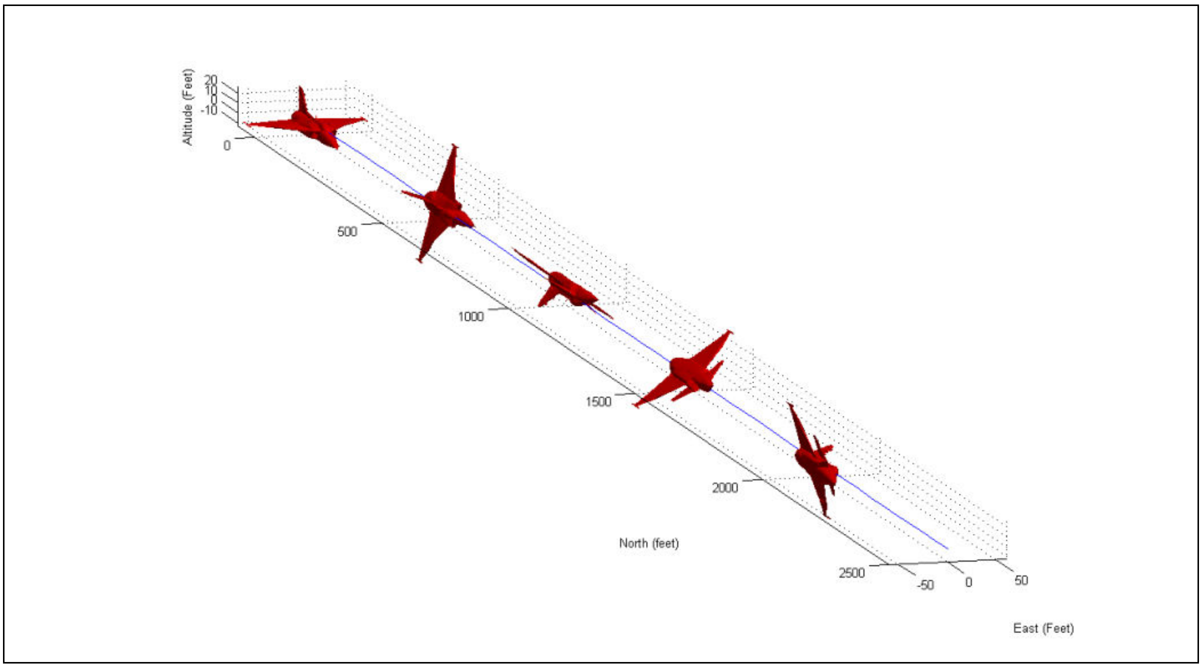


Figure 2.1 : Slow roll.

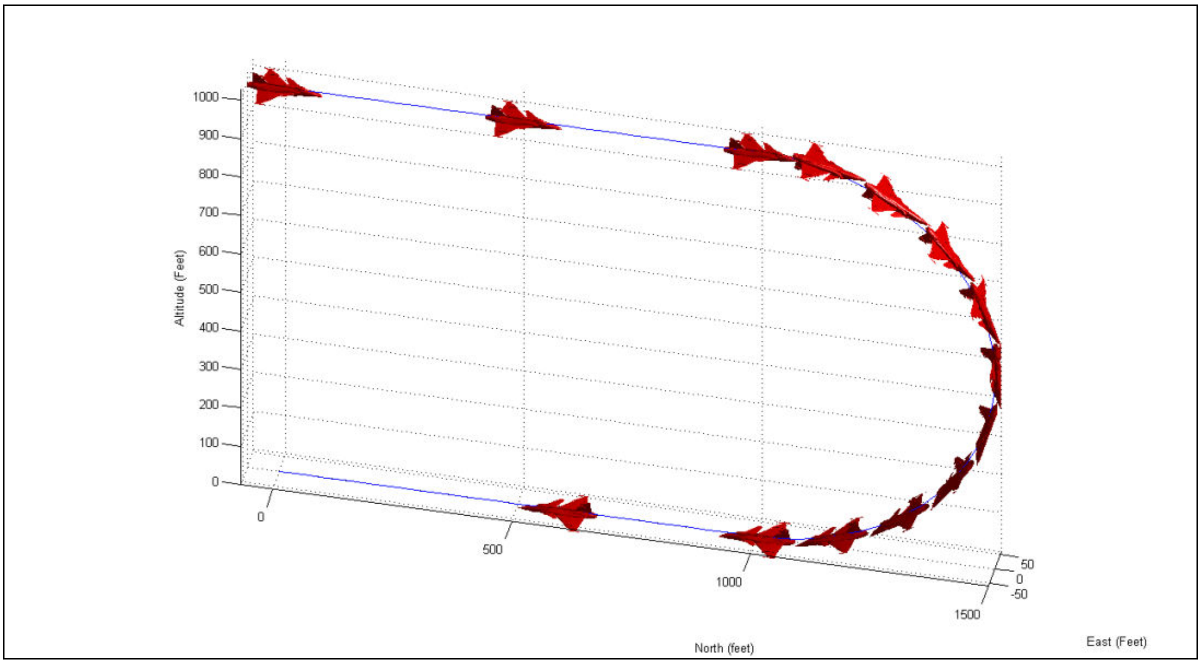


Figure 2.2 : Split s.

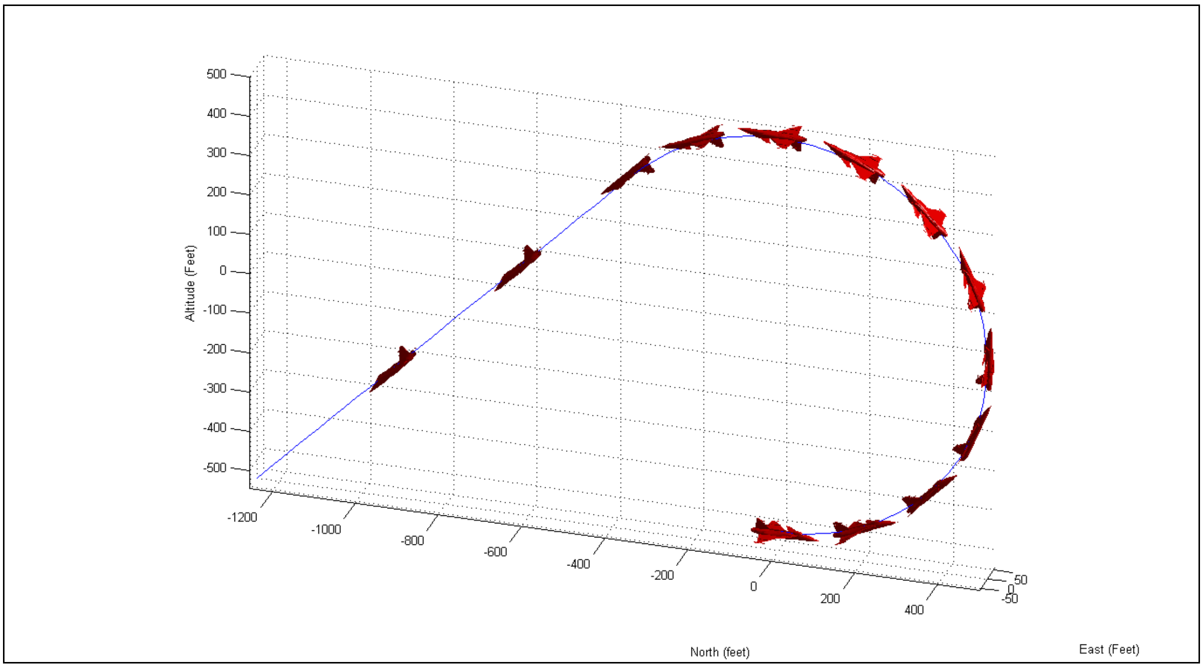


Figure 2.3 : Half cuban eight.

Now let's take a look at the modular structure of the aerobatics; a quick inspection of the Figure 2.1 through Figure 2.3 obviously tells us almost all aerobatics can be classified in to two categories; elementary maneuvers and maneuvers which can be represented as combination of elementary maneuvers.

For example, slow roll can be considered as an elementary maneuver so can the loop and Split S is simply a well ordered sequence of these elementary maneuvers. As a fact, roll, loop, spin and hammer-head is usually accepted as building blocks of arbitrary aerobatic maneuvers. This modular structure gives the pilot flexibility of learning the basic maneuvers to fly more challenging maneuvers. However, mastering elementary maneuvers does not indicate pilot will fly combined maneuvers easily, passing from one maneuver to another is another subject of its own and strongly connected to limits of the aircraft as well as transition logic.

Various books on aerobatics [35, 37] give a pretty good idea of how to fly elementary and combined maneuvers. Basically there are two important aspects of flying aerobatic maneuvers, entering it with appropriate speed and orientation and keeping the aircraft in limits of the flight envelope (from stalling and spinning) while keeping track of the maneuver parameters (such as altitude and roll rate). It is also possible to say that these aspects play an important role while switching between maneuvers. From modular point of view in the previous paragraph, each mode possesses an ideal initial condition and flying objective to fly it correct and safely. Thus, once the transition logic between each mode is implemented in the structure above, it is possible to derive a switched control system which operates on elementary flight modes to fly arbitrary complex aerobatics maneuvers.

2.1.2 Air combat maneuvering

Air combat Maneuvering (ACM) is the class of maneuvers which is performed by a fighter aircraft in either defensive or aggressive manner to gain tactical advantage over enemy aircraft during air combat. These maneuver are usually more complex than Aerobatics and evolves in three dimensions, due to fact that maneuvers are not motivated for artistic quality but to evade or attack the target aircraft. One of the main aims of air combat is to get in the targets behind (referred as cone of vulnerability) to fire its guns to bring the target down. More on information air combat maneuvering and tactics reader are directed to [38, 39].

A large portion of the Aerobatics shares maneuvers with ACM due to fact that most of the Aerobatics maneuvers are derived from maneuvers invented by fighter aircraft pilots during World War II. Two combat maneuvers for one on one dogfight are provided in Figure 2.4 and Figure 2.5 for example, taken from a website focusing on ACM [40]. From top to bottom these maneuvers are usually referenced as “High Yo-Yo” and “Split-S”. First maneuver of this example shows how defender aircraft can turn the situation into its own advantage by executing an out of plane maneuver; at the initial position attacker is behind the defenders cone of vulnerability, then defender executes a high speed climbing turning maneuver to get of the critical area while attacker aircraft continues to turn in the defenders direction; at the final stage defender dives sharply to get behind the attacker which results in interchange of their roles in combat. Second maneuver “Split-S” (which is an example of direct influence of ACM on aerobatics) is a defensive maneuver, where the defender rolls the aircraft inverted and performs a half loop to get outside firing range of the fighter aircraft.

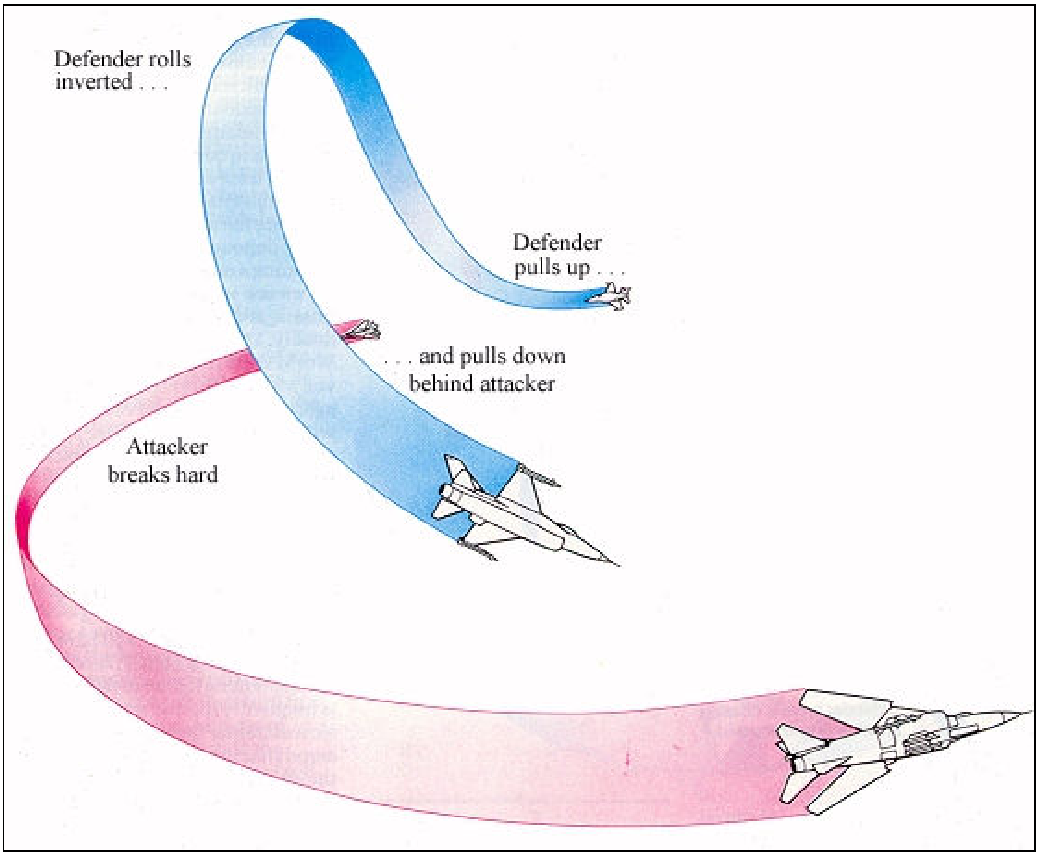


Figure 2.4 : High yo-yo [40].

Derivation of a modular structure is far more complicated in ACM than in aerobatics, due to fact that variation of maneuvers in three dimensions. Although it is possible to decompose maneuvers into “roll” or “turn” in most the situations, maneuvers like High Yo-Yo are far too arbitrary to break into meaningful parts. One possible methodology is to accept these maneuvers as modes without breaking them up and build a library based on these modes coupled with a decision strategy to model the air combat. However this results in major disadvantage, which is already discussed in the introduction part of the thesis; this maneuver library will likely to fail the span of full flight envelope characteristics of the aircraft.

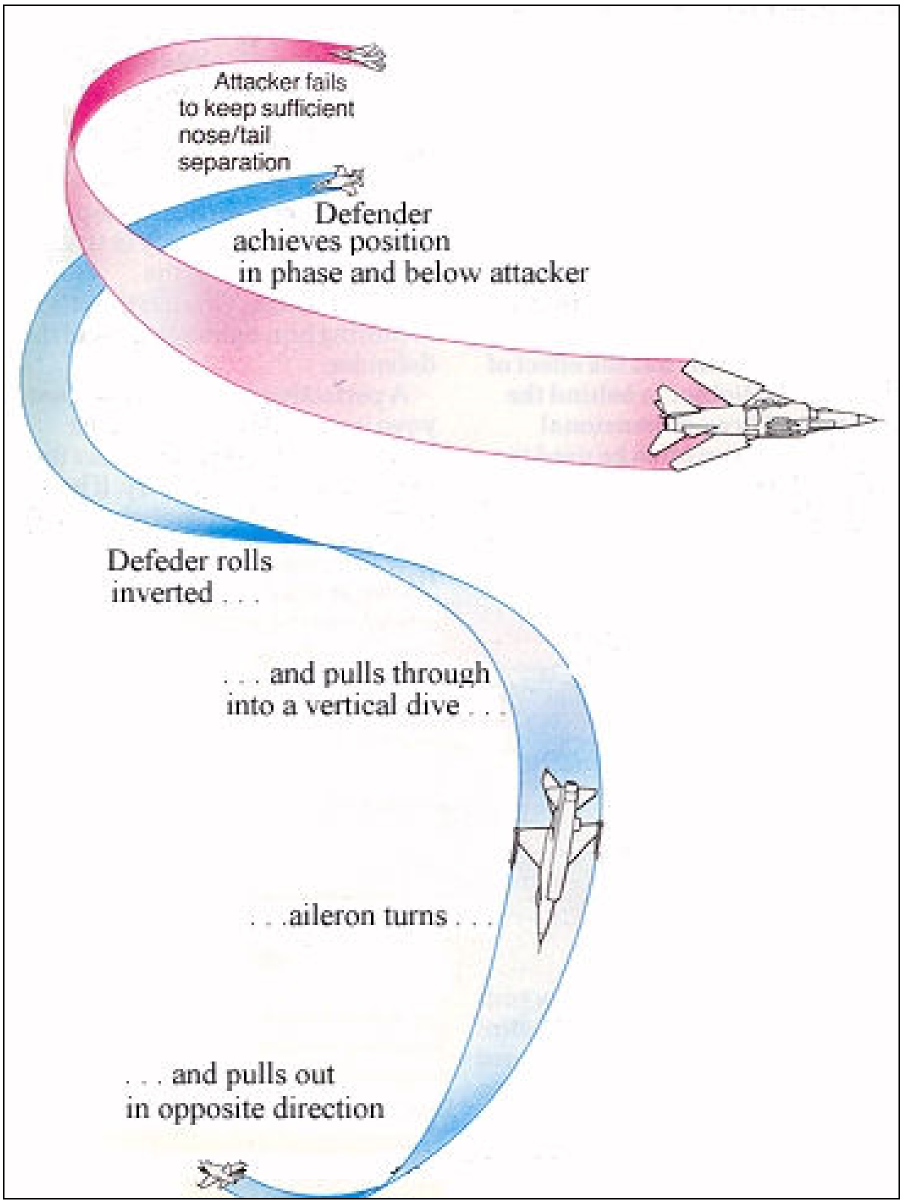


Figure 2.5 : Combat split s [40].

From control point of view, ACM also represents a series of difficult challenges. As discussed in the aerobatics section performing agile maneuvers is a complicated job on its own due to flight envelope limits and precise tracking of angular variables of aircraft. ACM is further complicated by the fact that maneuvering aircraft must also observe the state of the defender/attacker aircraft. Based on the attacker aircrafts situation; defender could be required to rapidly change its strategy/maneuver sequence which poses a series of difficult problems, like staying in envelope limits (in both flight and actuator envelopes) quick and precise transition between modes (when defender executes transition later than its need to be, attacker usually predicts its routine and answers with a counter maneuver which usually results in death of defender pilot). Therefore; a modular switching control strategy as suggested in aerobatics maneuver must be coupled with a decision strategy based on the enemy aircrafts maneuvers.

2.2 Concepts of Maneuver Modes and Modal Inputs

Inspection of agile maneuvers in terms of Aerobatics and ACM in the previous section has helped us to gain insight for modal decomposition of these maneuvers. In this section we are going to develop mode based agile maneuvering motion language, which will be used through entire thesis. This section also introduces the concepts of maneuver building blocks (referred as maneuver modes) and corresponding parameters (modal inputs) as well as controller design requirements for each mode.

2.2.1 On selection of modelling parameters and modal inputs

Before starting to define the maneuver modes let us remind the notation we will use throughout thesis to describe aircraft states and maneuvers (more on dynamic model of the aircraft can be found on Chapter 4). Selection of kinematic and dynamic variables to model the maneuvers are of great importance since one choice of coordinates may have significant advantage on other.

Since most of the maneuvers differ from each other from their geometry in inertial frame, switching between Cartesian, Spherical and Cylindrical coordinates may yield advantage for modelling in particular mode. Due to this reason, Cartesian coordinates positional kinematic variables $\rho_c = \begin{bmatrix} n_p & e_p & h \end{bmatrix}^T$ (north-position, east-position, altitude) are employed along with longitudinal and lateral cylindrical coordinates (more on the definition of these coordinates are presented in 3D modes subsection).

Perhaps one of the most critical parts of modelling is to decide on parameterization of attitude of A/C. While most of the works on agile maneuvering prefers Quaternions or non singular parameterizations to Euler Angles, we mostly prefer Euler angles for description since they are the most intuitive and natural method to describe the orientation of the aircraft. To avoid singularities some modifications will be made on the usual statement of Euler Angles in maneuver generation algorithm (see associated section) and in numerical simulations quaternions will be used to integrate the differential equations. So Euler Angles $\Phi = [\phi \quad \theta \quad \psi]^T$ (in the sequence roll-pitch-yaw) will be used in most of the modes.

For velocity variables, the trio $V = [V_T \quad \alpha \quad \beta]^T$ (Total velocity in body axes, angle of attack and sideslip angle) is preferred, due to fact that, total velocity will be treated as an output in most of the modes and aerodynamic angles are key parameters in determining the feasible motion due to fact that force and moment limits are strongly connected to amount of angle of attack (sideslip angle will be considered zero or will be regulated to zero in most of the cases). For angular velocities usual $\omega = [P \quad Q \quad R]^T$ vector is considered (roll, pitch and yaw rate in body axes).

True inputs of the aircraft are considered as $\delta = [\delta_T \quad \delta_e \quad \delta_a \quad \delta_r]^T$ (throttle, elevator, aileron and rudder deflections). Although it would be very interesting to consider thrust vectoring or direct force control actuators to expand the maneuver set, it is wise to keep it at conventional aircraft inputs at this point.

As indicated in the introduction chapter, one of the main aims of the thesis is to present flexible, parameterized sets of maneuver modes to cover full flight envelope. From now on, strict definitions of “mode” and “modal inputs” have to be made to build up the motion language for agile maneuvering.

To gain insight to what we mean by parameterized maneuvers, Figure 2.6 shows the basic difference between a set of “well defined maneuvers” and “parameterized” maneuver. In the first setting we have three distinct level flights with different velocity profiles which is the approach used extensively in the literature so far for the advantage in planning.

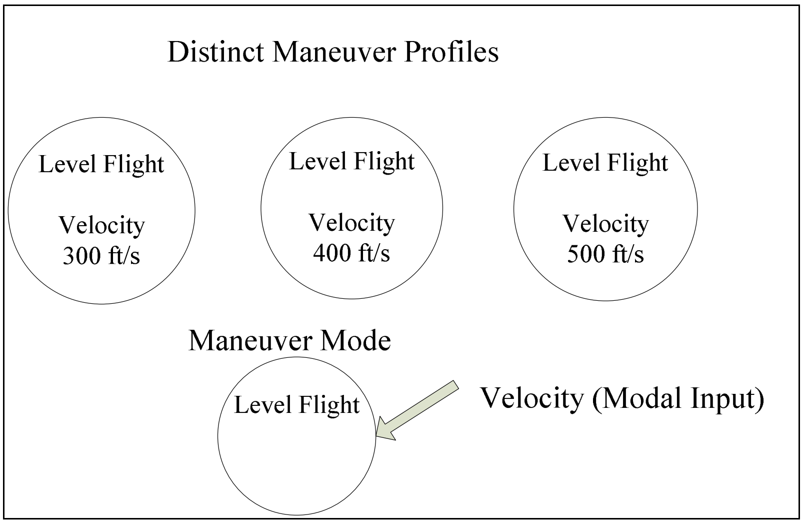


Figure 2.6 : Comparison between defined and parameterized maneuvers.

This approach, as shown in Figure 2.6 has the advantage of creating a parameterized maneuver family spanning full flight envelope dynamics of the aircraft. So in selection of the maneuver modes this mode-parameter connection will be taken into account.

Before starting to investigate the properties of these modes, let us note an important relation between the modal inputs and initial conditions. Modal inputs are maneuver parameters which defines the evolution of the dynamics of the mode for given initial conditions. In this sense, altitude (or any other position variable) cannot be a modal input to level flight mode, since due to state constraints on level flight altitude cannot change during execution of the maneuver therefore once the initial condition for altitude is given it stays at this value during the execution of the mode. This argument also applies to heading (Euler Yaw angle) in level flight. Velocity however is a modal input candidate to level flight since velocity can be subject to change during execution of the maneuver. This discussion represents the basis for finding a modal input to the maneuver modes in the next subsection.

2.2.2 Gathering possible maneuver modes

One way to derive the motion alphabet of the aircraft is to think on the set of minimal states which define a sub-maneuver (or a letter of the motion alphabet). Such list of modes will give a preliminary idea of constructing the final version of mode library.

One suitable approach is to divide the specifications of the sub-maneuver to three categories; altitude, rolling and looping; so three main specifications which define a sub-maneuver are; constant (A1) or time varying altitude (A2), constant (B1) or time varying roll angle (B2); drawing a loop in longitudinal (C2) or lateral plane (C3) Drawing a loop in longitudinal (C2) or lateral plane (C3).

Combining results of these three specifications gives the total number of flight modes which can be characterized by the combination of these properties. In this case it is twelve distinct modes as it is shown on Table 2.1. First three columns show the combinations of above listed properties and the rightmost column shows the resulting sub-maneuver. Name of the sub-maneuvers are taken from either aeronautics literature or aerobatics books.

Table 2.1 : List of all possible flight modes.

A1	B1	C1	Level Flight
A1	B1	C2	Pitch Regulation
A1	B1	C3	Coordinated Turn
A1	B2	C1	Rolling Level Flight
A1	B2	C2	Spin Roll
A1	B2	C3	Rolling Circle
A2	B1	C1	Climbing/Diving Flight
A2	B1	C2	Pitch Up/Down
A2	B1	C3	Climbing/Diving Turn
A2	B2	C1	Rolling Climbing/Diving
A2	B2	C2	Barrel Roll
A2	B2	C3	Rolling Scissors

A quick review of the Table 2.1 shows that, this approach indeed results in logical representation for a large scope of maneuvers. For example, common maneuvers like level flight and climb/descent are characterized by non-rolling and straight trajectories, while out of plane maneuvers are usually represented by non constant altitude and loops.

It must be noted that Table 2.1 is not exactly complete; it can expanded by adding coordinated / uncoordinated flight option (zero or non-zero sideslip angle). Since zero sideslip angle is desired most of the time, this option is omitted from now on.

Sub-maneuvers such as “rolling circle” or “rolling climbing/diving” (which indicates continuous rolling motion while flying) are also rarely encountered (except in some advanced aerobatics competitions) thus they are omitted in our analysis. For rolling motion only one mode will be considered called “roll mode” which acts as a transition mode between other modes, as it will be more evident in maneuver generation constraints.

Although Table 2.1 gives a good idea for building motion alphabet, it is somewhat redundant, and does not sort maneuvers from piloting and planning point of view. In the next subsection we categorize and sort the sub-maneuvers listed in table 2.1 to complete list we will use throughout the thesis.

Table 2.2 outlines the complete list that will be used in throughout the thesis;

Table 2.2 : Maneuver modes and modal inputs.

	Mode	State Constraints	Modal Inputs
q_0	Level Flight	$\dot{h}=0,(\dot{\phi},\dot{\theta},\dot{\psi})=0$	V_T,α
q_1	Climb/Descent	$(\dot{\phi},\dot{\theta},\dot{\psi})=0$	$V_T,(\dot{h},\theta_w)$
q_2	Roll	$(\dot{\theta},\dot{\psi})=0,\beta=0$	$V_T,\int P_w dt$
q_3	Longitudinal Loop	$(\dot{\phi},\dot{\psi})=0,$	$(V_T,r_{loop}),\dot{\theta}$
q_4	Lateral Loop	$\dot{h}=0,(\dot{\phi},\dot{\theta})=0$	$(V_T,r_{loop}),\dot{\psi}$
q_5	3D Mode	$\{\ \}$	V_T,P,Q,R $V_T,\phi_w,\theta_w,\psi_w$
q_6	Safety	$\{\ \}$	$\{0,1\}$

Now let us analyze the elements of the table; rightmost column carries the symbol associated with each mode q_i, where as the next columns is the formal name of the mode, third column represents the state constraints associated with each mode and the last column represents the modal inputs associated with each mode. Subscript w refers to wind axes and r_{loop} denotes the radius of the loop.

Selection of the modal inputs is based on the discussion in previous subsection. For example it is convenient to define both velocity and angle of attack history in level flight to represent the acceleration/deceleration maneuvers. Whereas 3D mode which represents out of plane maneuvers is best described by wind axes Euler axes or angular velocity profile itself depending on the situation.

It can be concluded that a multi modal maneuver description is an alternative to state space trajectory time history of aircraft $X(t)$ as a dynamical system. It gets rid of the redundancy of states while describing certain motions, by benefiting from the fact that some states are constant or zero during propagation of certain maneuver. So instead of giving the total state space time history, maneuver is described by the trio; flight mode q_i, modal input σ_i, and time interval Δt_i; the string $(q_i \quad \sigma_i \quad \Delta t_i), i = 1,2,...n$ is referred as maneuver string, and it is of lower dimension compared to $X(t)$. This property is further exploited in "Maneuver Identification Algorithm" in the next chapter.

2.3 Mode Based Maneuvering as an Hybrid Dynamical System

Up to this point nothing had been said about hybrid system presentation of the framework, but all of the results indicate that multi modal control framework is naturally a hybrid system, where discrete and continuous dynamics are found together. Each flight mode represents an element of discrete dynamics and each flight modes has continuous dynamics (represented by state vector). Each mode has its domain defined by flight envelope limits. Execution of the flight modes is constrained by mode transition table (which is introduced in the next chapter) and execution of continuous states is constrained by 6 degrees-of-freedom differential equations. Also, discrete modes and continuous dynamics have different input set (mode transition commands and modal inputs).

Multi modal control framework is converted into an automaton called Mode Based Maneuvering Automaton (MBMA), elements of MBMA are,

$$MBMA = \{Q, X, U, \Sigma, f, \delta, Dom, Init, \Omega\} \tag{2.1}$$

$$\begin{aligned} Q &= \{q_0, q_1, q_2, q_3, q_4, q_5, q_6\} = \\ &\{\text{Level}, \text{Climb}, \text{Roll}, \text{Long. Loop}, \text{Lat. Loop}, \text{3D}, \text{ Safety}\} \end{aligned} \tag{2.2}$$

$$X = \begin{bmatrix} V_T & \alpha & \beta & \varphi & \theta & \psi & P & Q & R & n_p & e_p & h \end{bmatrix}^T \tag{2.3}$$

$$U = \left\{V_T, \dot{h}, \int P_w dt, r_{loop}, \dot{\theta}, \dot{\psi}, P, Q, R, \phi_w, \theta_w, \psi_w \right\} \tag{2.4}$$

$$\Sigma = \bigcup_{i=1,j=1}^{i=1,j=7} \sigma_{ij} \tag{2.5}$$

$$\dot{X} = f_i(X, U, Q) \tag{2.6}$$

Where Q is the set of discrete flight modes in (2.2), X is the set of continuous states (flight dynamics) in (2.3) , U is the set of continuous inputs (parameters that generate modal inputs) in (2.4) and Σ is the set of discrete inputs (transition commands between modes) in (2.5). Discrete transitions are governed with transition relation, where δ symbolizes the elements of mode transition table. Dom is the domain of continuous states and inputs, which is defined by flight envelope limits. $Init$ is the set of initial states and flight modes. Mode transitions have to obey trajectory acceptance condition;

$$\begin{aligned} \Omega &= \left(q_1 \in Init\right) \wedge \left(q_{i+1} = \delta_{i,i+1}(q_i, \sigma_{i,i+1})\right) \\ &\wedge \left(q_i, x_i \in Dom\right) \end{aligned} \tag{2.7}$$

A diagram of multi modal control framework as a hybrid system is shown on Figure 2.7.

Once the system is converted to hybrid form, algorithms and methods developed for the system can be applied to automaton; this has two powerful advantages. First one is the reachability concept [41]. By using the system verification tools one can prove that every mode is reachable from given initial conditions which imply that there always exists a hybrid system trajectory (trajectories which obey equation 2.6) , that will connect an initial and final mode. Second advantage is for proving switched system stability; by constructing Lyapunov functions and using hybrid system stability one can show that stability in each discrete mode also implies switched system is stable [42]. These subjects are left to future research.

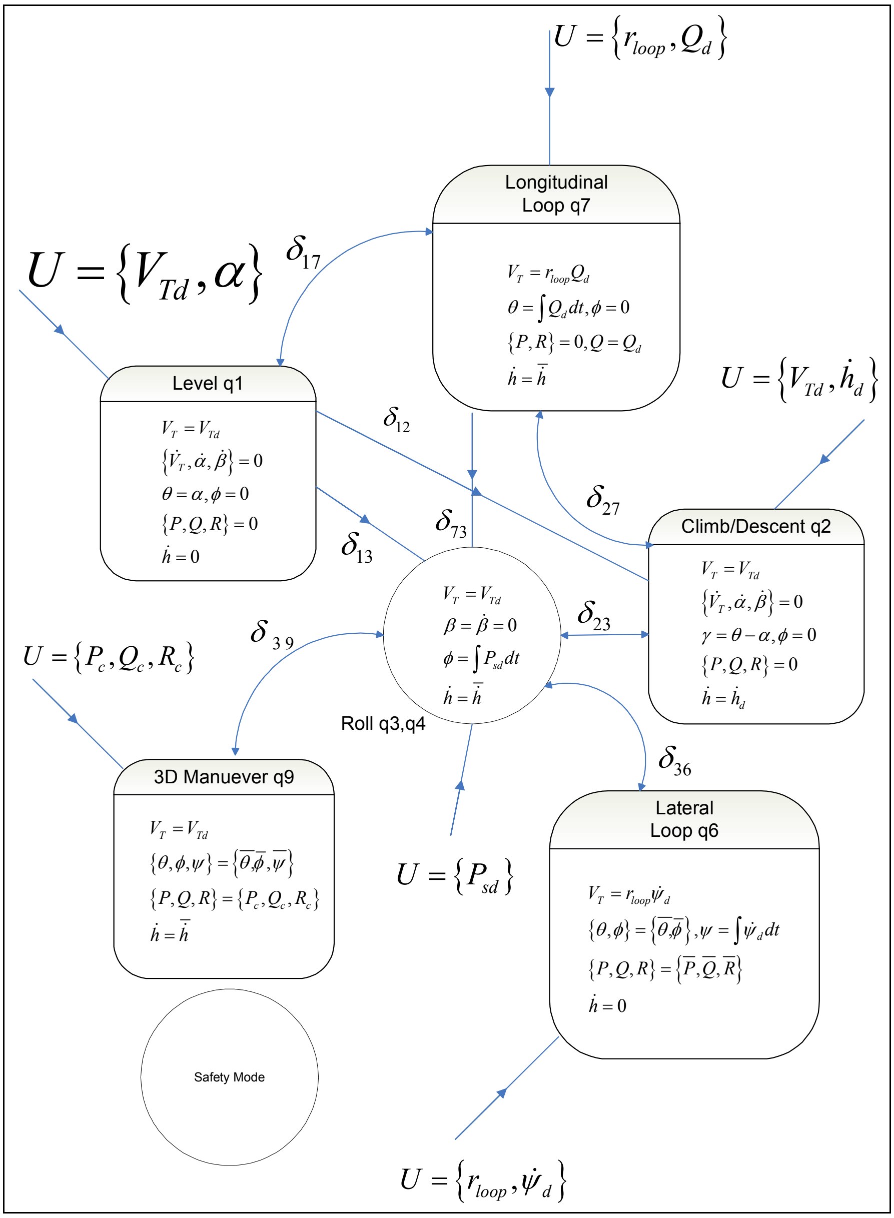


Figure 2.7 : Automaton diagram.

3. MANEUVER GENERATION ALGORITHM

This chapter focuses on development and structure of two algorithms which makes use of the modal structure developed in the previous chapter. The first one, maneuver identification algorithm converts the standard state space description of the aircraft dynamics and decomposes it to a string of maneuver modes modal inputs and time intervals. The second algorithm which is called maneuver generation algorithm accepts the flight trajectory as input and determines the mode sequence and modal inputs by taking mode Compability and flight envelope limits into account.

3.1 Maneuver Identification Algorithm

To strengthen the ideas provided in Chapter 2, an algorithm which converts the given state space trajectory of the aircraft into a modal sequence is discussed in this section. Easiest way to describe a maneuver in state space is to give the flight trajectory plus attitude on flight trajectory. This kind of description is not often used by path planners, which can give the flight trajectory in a discrete form with constant time step, but not the attitude. So this algorithm will convert a well described maneuver into a modal sequence without checking the feasibility of the trajectory.

This algorithm assumes that a feasible kinematic state space trajectory of the maneuver is given (feasible in the sense that maneuver is executable by the aircraft, so all of the saturation and flight envelope constraints are met), and by using the state constraints on each mode in Table 2.2, identifies the flight segments and extracts modal input from each of them.

In this section, firstly trajectory variables are transformed into wind axes, since they are more appropriate for identifying the modes, then how each mode can be identified via its state constraints is described. Then a general algorithm for converting the state trajectory is constructed and explained. Section ends with an aerobatic maneuver example.

3.1.1 Transformation to wind axes

In general, state space trajectory consists of 12 states that are given in(2.3). As it is explained in the introduction of this section, we will assume that flight trajectory$\left(n_p(t), e_p(t), h(t)\right)$, and attitude of the aircraft$\left(\theta(t), \psi(t)\right)$ is given via a path planner algorithm or extracted from flight data. The Euler roll angle is usually not given, because it is strongly connected to other equations and giving all six kinematic variables often results in infeasible flight maneuvers. Therefore we have the find roll angle time history from the given variables.

First, velocity variables are extracted from trajectory data in wind axes via(3.1).

$$\begin{bmatrix} \dot{n}_p \\ \dot{e}_p \\ \dot{h} \end{bmatrix} = V_T \begin{bmatrix} \cos\theta_w \cos\psi_w \\ \cos\theta_w \sin\psi_w \\ \sin\theta_W \end{bmatrix} \Rightarrow \begin{aligned} \psi_w(t) &= \tan^{-1}\left(\frac{\dot{e}_p}{\dot{n}_p}\right) \\ \theta_w(t) &= \tan^{-1}\left(\frac{\dot{h}}{\dot{e}_p}\sin\psi_w\right) \\ V_T(t) &= \sqrt{\dot{n}_p^2 + \dot{e}_p^2 + \dot{h}^2} \end{aligned} \tag{3.1}$$

After solving the wind axes velocity variables, we can easily find the angle of attack and roll Euler angle from kinematic equations. We follow the work of Kato here [43]. For this purpose let the orthogonal transformation matrix be presented by $R(.) \in SO(3)$, by using the transformation equations from inertial, wind and body axes we write;

$$R(-\beta, \alpha, 0).R(\psi_w, \theta_w, \phi_w) = R(\psi, \theta, \phi) \tag{3.2}$$

By setting sideslip angle zero, and comparing two sides of these equations which do not contain the variable ϕ_w, we write the equations in closed form;

$$\begin{aligned} \alpha &= f_a(\psi_w, \theta_w, \psi, \theta) \\ \phi &= f_a(\psi_w, \theta_w, \psi, \theta) \end{aligned} \tag{3.3}$$

Now that we have angle of attack and all of the attitude angles, we can start to identify maneuver segments.

3.1.2 Identifying maneuver segments

Simplest way to identify the maneuver segments is to discretize the flight trajectory check the state constraints between each adjacent waypoints. Once the particular set of waypoints is identified as a maneuver mode, set of modal inputs associated with maneuver mode can be found using the data obtained from flight trajectory. In this section, identification technique for each mode is presented.

Since the wind axes Euler angles are obtained from(3.1), it is possible to label each mode based on the time history of these angles. This is due to fact that, flight path is bended and twisted according to these angles independent of the velocity on the flight path. Since most of the maneuver modes are identified due to shape of the flight path, this is a very natural way for identifying them. Table 3.1 shows how wind axes Euler angles evolve during each maneuver mode.

Table 3.1 : Wind axes Euler angles identification.

Mode / Angle	θ_w	$\dot{\theta}_w$	ψ_w	$\dot{\psi}_w$
Level Flight	0	0	C	0
Climb / Descent	C	0	C	0
Roll	C	0	C	0
Lon. Loop	T	T	C	0
Lat. Loop	C	0	T	T
3D Mode	T	T	T	T

On the Table 3.1 “C” means constant and “T” means time varying. Table is self-explanatory; for example in level flight, flight path angle (or wind axis pitch angle) must be zero so that aircrafts altitude doesn’t change, while heading can take any value as long as it doesn’t vary with time (i.e. zero derivatives). This straightforward logic is applicable to every mode on the table. Only exception is the roll mode, which also requires to check the Euler Roll Angle history (if Wind axes Euler angles are constant and the Euler Roll angle is time varying, mode is identified as “roll mode”). Therefore one can decompose the maneuver into waypoints and check the obtained wind axes Euler angles to identify each mode via Table 3.1. Next step is to obtain modal inputs for each mode.

After decomposing the trajectory and labelling each mode from Table 3.1, we have to extract the modal inputs associated with each mode. Total velocity is a modal input for every mode and it is available from(3.1). For level flight and Climb/Descent mode, everything needed is already available on table 3.1. (Climbing and diving rate). In roll mode, only desired roll angle displacement (value of the integral associated with roll mode on Table 2.2 is needed, which can be obtained from Euler roll angle time history. Loop modes require the body Euler angle rates as modal input, which can be obtained from $\theta(t)$ and $\psi(t)$ for Longitudinal Loop and Lateral Loop respectively.

Things are a bit more complicated in 3D mode, because it is required to extract the angular body rates from a given 3D trajectory and attitude data. Since all the body Euler angles are available, kinematical equation for Euler angles can be solved inversely to acquire the angular rates, but this is not convenient since equations have singular points and requires manipulating trigonometric equations. More elegant approach would be converting the Euler angles to Quaternions, and solve the algebraic, singularity free Quaternion kinematical equation to obtain body angular rates. (3.4) Gives the well known formula for converting Euler angles to Quaternions and (3.5) shows the kinematical Quaternion equation which has to be solved inversely in order to obtain the angular rates.

$$Q=\begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix}=\begin{bmatrix} \pm\left(C\frac{\phi}{2}C\frac{\theta}{2}C\frac{\psi}{2}+S\frac{\phi}{2}S\frac{\theta}{2}S\frac{\psi}{2}\right) \\ \pm\left(S\frac{\phi}{2}C\frac{\theta}{2}C\frac{\psi}{2}-C\frac{\phi}{2}S\frac{\theta}{2}S\frac{\psi}{2}\right) \\ \pm\left(C\frac{\phi}{2}S\frac{\theta}{2}C\frac{\psi}{2}+S\frac{\phi}{2}C\frac{\theta}{2}S\frac{\psi}{2}\right) \\ \pm\left(C\frac{\phi}{2}C\frac{\theta}{2}S\frac{\psi}{2}-S\frac{\phi}{2}S\frac{\theta}{2}C\frac{\psi}{2}\right) \end{bmatrix} S:Sin, C:Cos \qquad \textbf{(3.4)}$$

$$\dot{q}=\begin{bmatrix} \dot{q}_0 \\ \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{bmatrix}=\begin{bmatrix} 0 & -P & -Q & R \\ P & 0 & R & -Q \\ Q & -R & 0 & P \\ R & Q & -P & 0 \end{bmatrix}\begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix} \qquad \textbf{(3.5)}$$

3.1.3 Identifying maneuver segments

Now we are ready to write down the algorithm for, decomposing the given feasible flight trajectory in maneuver mode sequence, thus identifying the maneuver in the proposed motion language. Algorithm 1 displayed in Table 3.2 (Maneuver Identification /Decomposition algorithm), takes the flight trajectory and Euler pitch and yaw angle time history as input. In the first and second step, algorithm computes the rest of the state space variables by the equations given in this section. In the third step flight path is divided in to waypoints with constant time step (so flight path is "sampled" in a sense). This allows us to check each flight segment step by step and compare the wind axes Euler angles of each waypoint. This task is done at fifth step in the algorithm, and the mode is labelled after finding its name on the Table 2.2. In sixth step modal input set is obtained from state space trajectories or kinematic equations given in sub subsection "Obtaining Modal Inputs". After labelling and finding modal input tasks, total time duration is calculated from summing the constant time step intervals. Then these tasks are repeated for next maneuver mode, until the flight trajectory is complete. Output of the algorithm is the modal sequence which has total N pieces.

Output of the algorithm is shown below on an example aerobatics maneuver Cuban Eight in Figure 3.1.

Cuban eight is a popular aerobatics maneuver, which consists of connected loops and rolls in order to draw an "eight" figure in the 2D plane. Maneuver can be broken down into maneuver mode and modal input sequence as shown by Figure 3.1 and Table 3.3.

Advantage of multimodal state framework is easily seen; instead of giving 12 time-state curve for these 14 seconds lasting maneuver, all the information is contained in table 3 with only 4 maneuver modes and modal inputs.

In this section maneuver identification algorithm was developed and tested on an agile flight maneuver. Only direct use of the algorithm is possible if the path/motion planner guarantees that the generated flight trajectory is feasible. Then this algorithm neglects checking feasibility and concentrates on decomposing the given maneuver into language of maneuver modes and modal inputs. After this decomposition multi-modal control approach can be used on the system to track the maneuver.

Table 3.2 : Maneuver identification algorithm.

Algorithm I: Maneuver Identification / Decomposition
Input : Flight Trajectory $\left(n_p(t), e_p(t), h(t)\right)$ and attitude angles $\left(\theta(t), \psi(t)\right)$
Output: Modal Sequence $\left(q_i, \sigma_i, \tau_i\right), i = 1, 2..., N$
1: Solve the velocity variables form eq. 2
2: Solve the angle of attack and roll angle history from eq. 4
3: Discretize the flight trajectory, with constant time step Δt
4: Repeat
Repeat
5: Check $\left(\theta_{w_{j+1}} - \theta_{w_j}\right)$ and $\left(\psi_{w_{j+1}} - \psi_{w_j}\right)$, Label The mode according to table 2.
6: Switch $q_i \leftarrow$ "Mode Label"
Case " Level Flight"
$\sigma_i \leftarrow$ {Obtain From $V_T(t)$ }
Case " Climb / Descent"
$\sigma_i \leftarrow$ {Obtain From $V_T(t), \theta_w(t)$ }
Case "Roll Mode"
$\sigma_i \leftarrow$ {Obtain From $V_T(t), \phi(t)$ }
Case "Lon. Loop"
$\sigma_i \leftarrow$ {Obtain From $V_T(t), \theta(t)$ }
Case "Lat. Loop"
$\sigma_i \leftarrow$ {Obtain From $V_T(t), \psi(t)$ }
Case "3D Mode"
$\sigma_i \leftarrow$ {Obtain From $V_T(t)$,eqs. 5 and 6}
Until Next Mode
7: Calculate duration of the mode $\tau_i \leftarrow \sum \Delta t_i$
Until Trajectory is complete $i = N$

However in general case, path planner do not focus on feasibility but only on finding a path that connects initial and final waypoints while avoiding forbidden areas (such as obstacles) in the environment. Again in general they will not give the attitude of the aircraft but only the flight trajectory in 3D plane possibly without any time constraint. Therefore the algorithm must find the feasible sequence of maneuver modes and modal inputs for this trajectory as well as the time duration (therefore velocity) for each mode.

Next section is dedicated to design of such an algorithm called "Feasible Maneuver Generation Algorithm", which approaches the feasibility problem via sequential and envelope constraints, and generates the feasible maneuver sequence by the help of searching tables and graphs based on the modal approach of the proposed language.

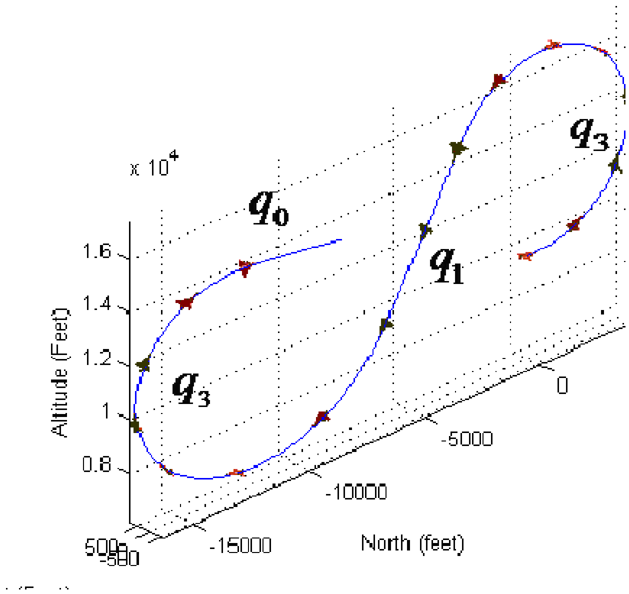


Figure 3.1 : Cuban Eight.

Table 3.3 : Modal Decomposition of Cuban Eight Maneuver.

Maneuver Mode	Modal Inputs	Time Intervals (sec)
q_3	$r_{loop} = 3000\,ft, \dot{\theta} = 225/5 \deg .s^{-1}$	$[0,5]$
q_1	$V_T = 400\,ft/s, \theta_W = 45\deg$	$[5,7]$
q_3	$r_{loop} = 3000\,ft, \dot{\theta} = 225/5 \deg .s^{-1}$	$[7,12]$
q_0	$V_T = 400\,ft/s, \alpha = 2\deg$	$[12,14]$

3.2 Maneuver Generation Algorithm

One of the main problems of the hierarchical control system design is to select appropriate layers and corresponding algorithms in the architecture. Usually path planning layer is positioned directly on the feedback control layer; that is trajectory generated by path planning is directly controlled via a low level control system. Although this approach could give successful results for a wide class of autonomous systems (such as mobile robots) it is not likely to work efficiently for agile maneuvering aircraft due to fact that attitude of the aircraft is equally important in maneuvering as seen in the examples of Chapter 2. Methodology of Multi Model Control Framework can be inserted between path planning and control layer as a "Maneuver Planning Layer", where trajectory is decomposed into maneuver modes and modal inputs are selected within feasible limits. This new architecture is displayed in Figure 3.2 .

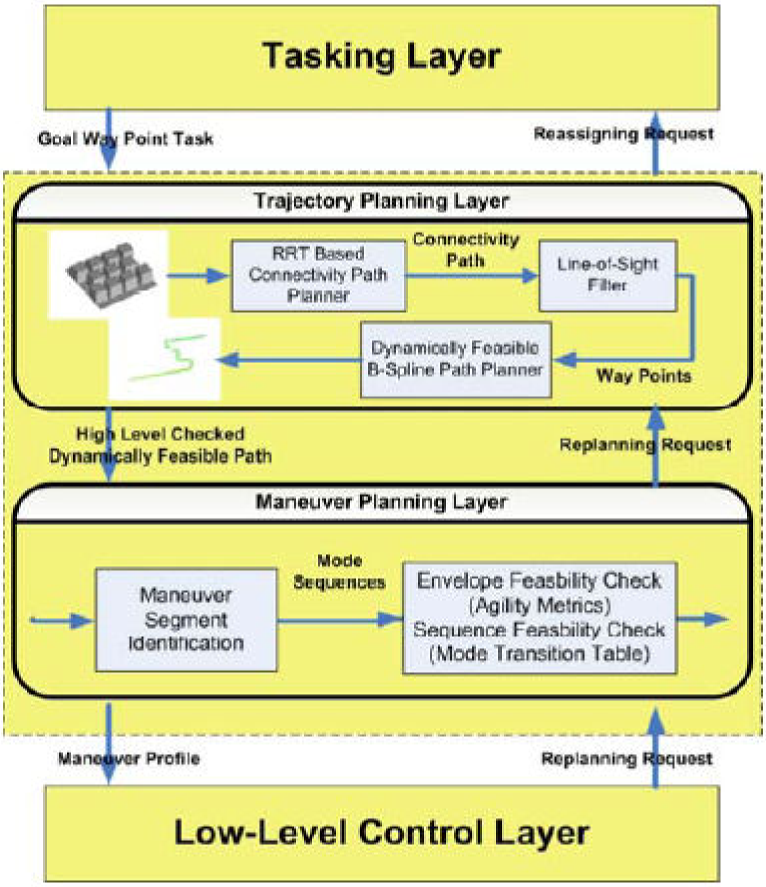


Figure 3.2 : Integrated path/maneuver planning.

This section focuses on creating a feasible maneuver sequence from a given flight trajectory. So, when integrated with a path planner, this algorithm will solve the motion/planning problem in multi modal control framework. Feasibility in this section is divided into two classes; feasibility of the mode sequence (sequential constraints) and feasibility of modal inputs (envelope constraints). If the modal sequence satisfies these constrains it would be accepted as a feasible maneuver. Thus, algorithm created in this section consists of the expansion maneuver identification algorithm form previous section with feasibility checks of the constraints above.

Firstly, physical nature of these two types of constraints along with the proposed solution methods for dealing with these constraints will be presented. Next, general structure of the algorithm is discussed. Finally, an example agile trajectory taken from a path planner will be converted to a feasible maneuver sequence.

3.2.1 Sequential constraints and mode transition table

Multi modal control system is based on the successful execution of one mode after another, but this doesn't mean that mode sequence is arbitrary. We can simply cast the mode compatibility condition as, final states of the first maneuver mode must intersect with the second modes initial conditions [44]. This is valid for kinematical variables, since most of the dynamic variables (i.e. modal inputs) are subject to envelope constrains rather than sequential which will be discussed in the next section.

For the trajectory variables n_p, e_p, h there is no problem, since the trajectory given by path planner is already connected. Challenging part is the attitude compatibility problem, because attitude is mostly unnatural to flight path. Therefore, we have to check if the attitude of each mode is compatible with each other. This problem mostly rises on body roll angle and pitch angle. For example for a lateral loop, aircraft is assumed to have an initial roll angle required for that turn, however in level flight wings are kept at zero roll angle. So transition for level flight to lateral loop needs a roll mode in between to connect each modes attitude. Analogously, longitudinal loop mode can be used for same purpose to set to correct pitch angle when two modes pitch attitude is not compatible with each other. Based on this discussion, a mode transition table was prepared on Table 3.4 , which explicitly shows which mode are directly compatible with each other and which needs a change of attitude between them.

Table 3.4 : Mode transition table.

δ_{ij}	q_0	q_1	q_2	q_3	q_4	q_5	q_6
q_0	1	θ^*	1	1	ϕ^*	1	1
q_1	θ^*	θ^*	1	1	θ^*, ϕ^*	ϕ^*	1
q_2	1	θ^*	1	θ^*	1	θ^*	1
q_3	1	1	1	1	θ^*, ϕ^*	ϕ^*	1
q_4	ϕ^*	θ^*, ϕ^*	1	θ^*, ϕ^*	ϕ^*	θ^*, ϕ^*	1
q_5	θ^*, ϕ^*	θ^*, ϕ^*	1	θ^*, ϕ^*	θ^*, ϕ^*	θ^*, ϕ^*	1
q_6	1	0	0	0	0	0	0

Inspection of the Table 3.4 shows that roll mode and longitudinal loop mode acts as a transition mode between the others because they can create the desired shift in roll and pitch angles. Thus one can construct maneuvers by using level and climbing flights along with loops and connect them using rolling and pitching. It is also noted that safety mode is accessible all the time, once it is activated it automatically regulates back to level flight, so all other transitions from this mode is blocked.

3.2.2 Envelope constraints and agility metrics

Second types of constraints are risen form physical limitations of the aircraft, i.e. envelope constraints. Due to capability of engine and control surfaces of the aircraft, as well as the aerodynamic limits of the aircraft, there is a certain limit on the max. aircraft velocity, max. climb rate etc. However these boundaries are not constant, they vary continuously with the aircrafts angle of attack, altitude etc.

These constraints can be embedded into multimodal control framework as limits on modal inputs. Then feasible maneuver generation algorithm can select the modal inputs while ensuring they are in operation limits.

This method however, does not reflect the true agility characteristics of the fighter aircraft. Limits imposed on modal inputs (as well as envelope limits) shows only potential agility of the aircraft and often limited to trimmed maneuvers, they do not show the transient capabilities of the aircraft. Therefore, a more elegant solution would be searching the modal input parameters not in their original envelope, but in another norm, which could show the agile capability of the aircraft.

A similar problem, in fighter aircraft technology development was studied during 70s. Aircraft designers wanted the design and compare fighter aircrafts in new metrics, because existing metrics (such as wing loading to thrust to weight ratio) was incapable of showing aircrafts transient performance, and often shadowed the maneuver capability of aircraft. Such metrics were found, tested on the simulations on various NASA reports [45-48].

In this subsection, we take the advantage of multimodal control framework and develop separate agility metrics for each mode. Most of these metrics were taken from the NASA reports above. All of the metrics were evaluated on a high fidelity F-16 aircraft simulation which is also represented in the Chapter 4 [49]. Detailed description of the metrics and simulation results are contained below.

Some of the metrics were chosen and inspected in previous work but as the research progressed, these metrics were replaced by more convenient metrics [45]. Metric graphs were produced by nonlinear 6 DOF F-16 simulations via feedback control loops for various angle of attack and Mach numbers for various altitudes. Example graphs given at the end of the subsection are for sea level.

For level and climbing/diving flight total speed and acceleration capability is the most important parameter. Maximum and minimum achievable speed depends heavily on available power and thrust. Selected agility metric is power onset/loss parameter which can be written as:

$$\dot{P}_s = \frac{d}{dt}\left(\frac{V_T\left(T-D\right)}{W}\right) \tag{3.6}$$

Where T is thrust, D is drag and W is weight. This agility metric quantifies maximum thrust and drag of the aircraft, which determines the total speed and acceleration capability. This metric is plotted against Mach number and angle of attack, in a 3D region for F-16 in .

For rolling motion either average roll rate or time to go through certain roll displacement can be used. Second one is more convenient as it gives transient performance more clearly. For specific angle 90 degrees can be used, because most of the rolling maneuvers consists of rolling aircraft to side (knife edge), or inverting it (180 degrees). Therefore selected agility metric is

$$TR_{90} = \text{ Time To Go Through 90 degrees roll angle} \quad \textbf{(3.7)}$$

3D plot of this metric is shown in Figure 3.4.

For pitch up/down motion, very simple and convenient metric is average pitch rate which can be written as

$$Q_{avg} = \frac{\int_{t_1}^{t_2} Q dt}{t_2 - t_1} \quad \textbf{(3.8)}$$

Plot of this metric is shown in Figure 3.5.

For turning performance load factor and turning radius is chosen as a predominant factor, by using the simple kinematic equation [50]:

$$r_{loop} = \frac{V_T^{\ 2}}{g(n^2 - 1)} \quad \textbf{(3.9)}$$

In 3D mode both rolling and pitching motion becomes dominant; therefore we seek a metric which can combine these two properties. An appropriate metric is loaded roll which is given by the formula

$$PN = p_w N_{z,w} \quad \textbf{(3.10)}$$

This is simply the product of roll rate in wind axes and normal acceleration in wind axes. This metric belongs to torsional agility and combines the rolling motion of the aircraft with bending of the flight path.

3.2.3 Structure of the algorithm

General structure of the algorithm (outlined in Table 3.5) is explained step by step below;

Initially, path planning algorithm provides the flight trajectory history, $n_p(t), e_p(t), h(t)$. From this data it is easy to recover the velocity variables from equation 3.1. Since mode labeling action only depends on wind Axes Euler Angles (via table 3.1), it is possible to recover the mode sequence and time durations for each mode. However complete modal sequence is incomplete because modal input sequence cannot be recovered without attitude history. In the next step each mode will be analyzed via agility metric graphs, and modal inputs will be recovered from flight equations or graphs.

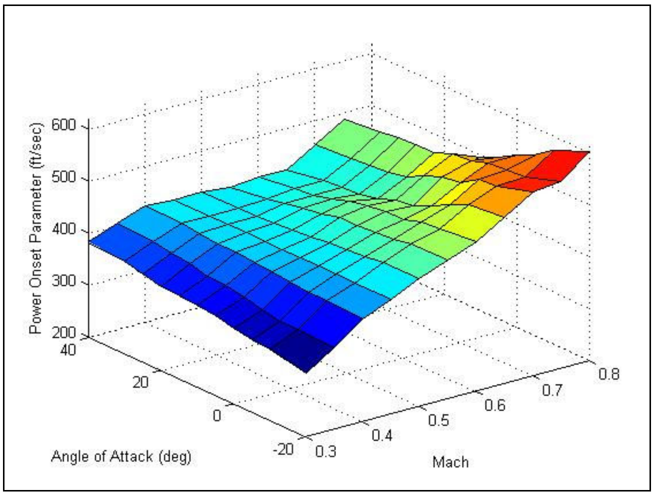


Figure 3.3 : Power onset parameter metic for sea level.

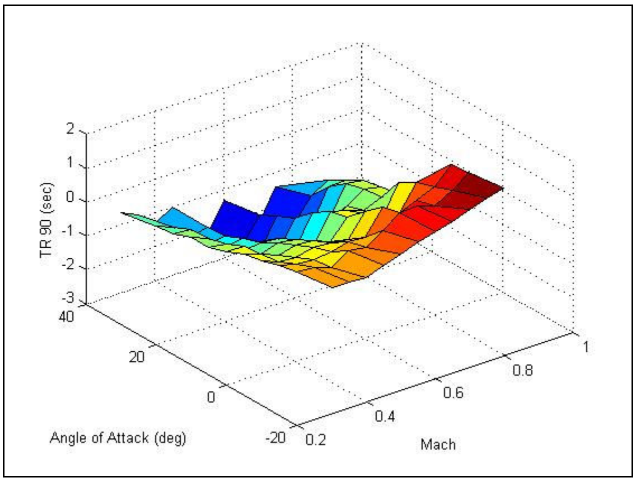


Figure 3.4 : Time to capture 90 deg roll angle metric for sea level.

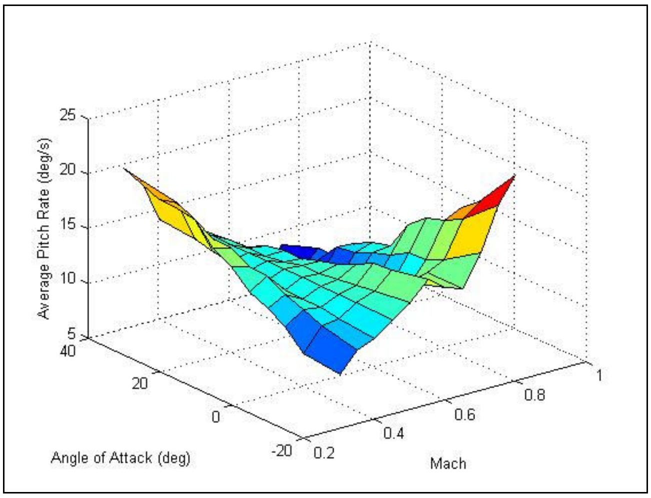


Figure 3.5 : Average pitch rate metric for sea level.

Since we have decomposed the flight trajectory into mode sequence, we can determine the feasibility of each mode and recover its modal input parameters (and quantify its agility). Very similar to Maneuver identification algorithm, this process is different for each mode. For example in level flight since the velocity is known (so is the Mach number), only free variable on 3D agility metric plot is angle of attack. Since searching for a certain parameter or optimization process could take time and damage the real time implementability of the framework, free parameter angle of attack is chosen randomly in the feasible interval, but with more weight on a certain subinterval (this method is very similar to kinodynamic path planner appeared on previous paper, where modal inputs are directly selected on this methodology [30]). If the corresponding velocity is not in the feasible interval in the graph, or results in angle of attack with unacceptable values, flight paths time can be scaled to change the velocity variable (for detailed discussion on time scaling of flight paths see[6]).

For other modes, process is similar; once a desired velocity and angle of attack couple is found from agility metric graph, modal inputs are recovered from flight equations 3.3-3.5 and agility metric equations (3.6) - (3.10). Also when modal inputs of the i^{th} mode is recovered, if the $(i+1)^{th}$ mode is sharing the same modal inputs, σ_{i+1} initial values of the are chosen same as σ_i to maintain continuity.

When both total velocity and angle of attack is recovered, it is very easy to recover rest of the attitude history via (3.2) and (3.3) . Recovering attitude history is important since it will help us to connect the disconnected maneuver modes via mode transition table in the next step.

Now the flight path is decomposed into modes and modal inputs (which are feasible in the sense that they fall into feasible region of agility metric graphs) but modes must satisfy the mode compatibility condition which was discussed on section 3.2.1. Flight path is already given connected via path planner but attitude must be checked via Table 3.4. At this step, mode transitions which satisfy the table are neglected and the modes which require attitude tweaking (in either roll or pitch angle) is checked to be compatible. If they are not, additional translational modes (roll mode and longitudinal loop) are inserted between these modes to connect the attitudes of each mode. This attitude changes are made quickly as possible to not to change shape of the flight trajectory, if these transitions make dramatic changes on the flight trajectory, re-planning of the trajectory may be required to make sure that path avoids the obstacles. This interaction between the multimodal framework and path planning can be found on [33].

After checking the mode transition table, and finding the appropriate attitude changes for sequential feasibility, these transition modes are added to original mode sequence and final modal sequence is recovered. This modal sequence is feasible in the sense that it satisfies the envelope and sequential constraints. An example for an agile combat maneuver is provided to show the capability of the algorithm.

3.2.4 Application to agile combat maneuver

The maneuver is inspired from the book which displays the agile maneuvering capability of an aircraft, through connecting loops and rolls which is very common in fighter combat [50]. Maneuver is similar to High Yo-Yo maneuver where aircraft gains altitude in order to make a steady fast on the target get behind it to enter into firing range.

Trajectory generated by the path planner, shown on Figure 3.6.This trajectory is entered as input the Algorithm II. By going through the steps of the algorithm, feasible modal sequence is created, which can be found on Table 3.6. It is revealed that, maneuver (on Figure 3.7) starts with a level flight, then continues with a loop in the longitudinal plane, then moves onto on arc in 3D plane, it is seen that start of the arc and end the loop had different roll angles, so this is compensated by adding a roll mode between these two. After 3D mode aircraft begins to dive, and finally enters into another arc. Another roll mode followed by a mini longitudinal loop is inserted here in order to satisfy sequential constraints.

Time history of fey modal inputs selected on agility metric criteria and kinematic parameters are shown in Figure 3.8 (red square highlight the modal inputs for each mode). This feasible modal sequence profile, actually acts as a reference profile for the switched control system (multimodal control framework) which will eb introduced introduced in Chapter 0.

Table 3.5 : Maneuver generation algorithm.

Algorithm II: Feasible Maneuver Generation
Input : Flight Trajectory $\left(n_p(t), e_p(t), h(t)\right)$ Output: Feasible Modal Sequence $\left(q_i, \sigma_i, \tau_i\right), i = 1, 2..., N$ 1: Run Maneuver Identification Algorithm (Algorithm I) up to line 5 and recover q_i 2: Repeat Switch "Mode Label" Case "Mode Label" Check Agility Metric Graph, recover α Recover Modal inputs σ_i through flight And agility metric equations. Adjust σ_i such that that it is compatible with σ_{i-1} Until Trajectory is complete 3: Repeat Check Mode Transition between q_{i-1} And q_i, through table 4. Case 1 Do Nothing Case θ^* Insert "Longitudinal Loop" for pitch regulation Case ϕ^* Insert "Roll Mode" for Roll pitch regulation Case θ^*, ϕ^* Insert both "Longitudinal Loop" And "Roll Mode" Until Trajectory is complete 4: Gather feasible modal sequence by combining results of 2 and 3

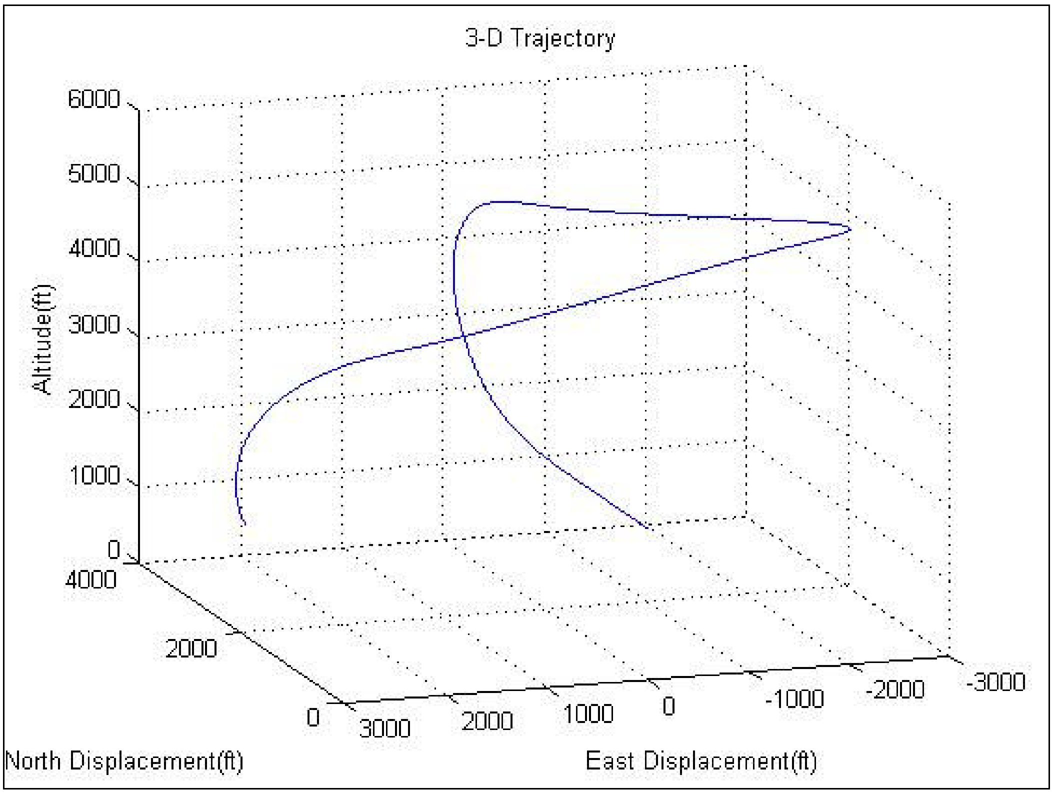


Figure 3.6 : Trajectory generated by the path planner.

Table 3.6 : Modal description of agile combat maneuver.

Mode	Label	Time Intervals (sec)
q_0	Level Flight	$[0,3]$
q_3	Longitudinal Loop	$[3,9.7]$
q_2	Roll Mode (Transition)	$[9.7,11.7]$
q_5	3D Mode	$[11.7,22.8]$
q_1	Dive	$[22.8,25.4]$
q_2	Roll Mode (Transition)	$[25.4,26.6]$
q_3	Longitudinal Loop (Transition)	$[26.6,27.8]$
q_5	3D Mode	$[27.8,32]$

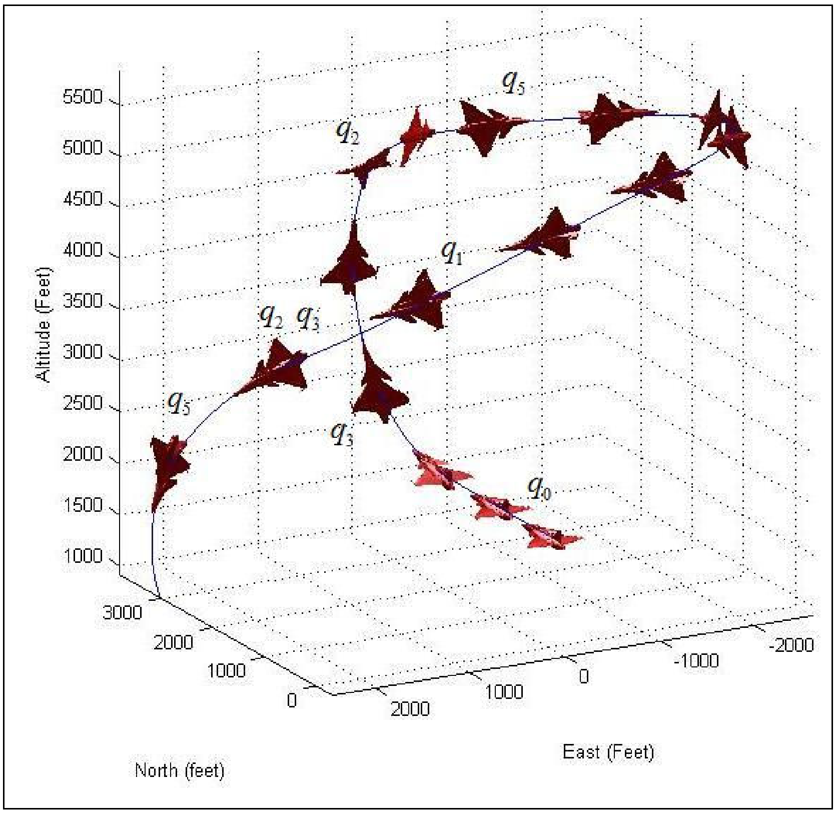


Figure 3.7 : Modally decomposed agile combat maneuver.

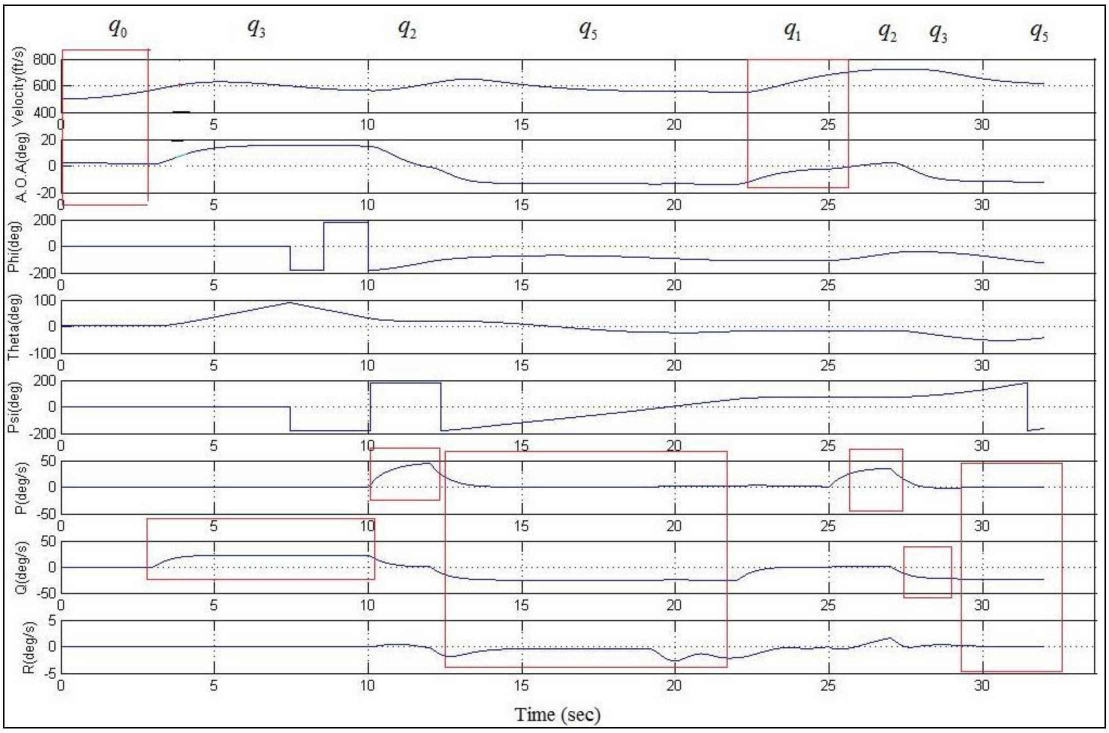


Figure 3.8 : State and modal input history for agile combat maneuver.

4. NONLINEAR 6 DOF F-16 MODEL

Before starting to develop control system design strategies, it is reasonable to spend time on building a reasonable model. By "reasonable" we mean an aircraft model, sophisticated enough to represent real life flight dynamics, but simple enough to apply mathematical tools on it.

4.1 Dynamic Model

In this part we will quickly derive necessary equation for rigid body 6-DOF dynamics. It is quite easy to see that these equations are valid for any rigid body translates and rotates in space (aircraft, spacecraft,...etc). What makes these equations specific to aircraft is the aerodynamic forces (and engine thrust) exerted on the aircraft and alteration of these forces through translation in atmosphere and deflection of control surfaces.

4.1.1 Axes definitions

In the development of the control equations, reference axes are taken an important place. The form of equations of motion depend on axis systems, which equations are defined in. Axis systems are classified as to their origins. The axis system, whose origins are at the center of gravity of aircraft, are called body, stability and wind axis systems. The other axis system, whose origin is at the center of earth, is called earth axis system.

4.1.1.1 Body axes

The body axis is fixed to the aircraft. Its origin is at the aircraft center of gravity. It moves with aircraft. The X_B axis passes through the nose of the vehicle, the Y_B axis points toward the right wing, and the Z_B axis passes through the bottom of the vehicle.

4.1.1.2 Stability axes

The stability axis is defined by a rotation with an angle of attack, α , from the body X axis while the body Y and Z axis is fixed. The angle of attack from body X axis to stability X axis has the positive sign.

4.1.1.3 Earth axes

The earth axis is fixed to the earth as an inertial reference axis. It does not rotates and translates. Its origin of coordinates is at the earth's center. The Z_E axis points northward and the X_E axis intersects the zero degree longitude line (Greenwich) at "zero time".

4.1.1.4 Local axes

The local axis is fixed to the aircraft center of gravity and it follows the motion of the aircraft. Its X_L axis points northward, Y_L axis points eastward; both are parallel to the earth's surface. The Z_L axis points towards the earth's center. It has a constant distance from the center of the earth R .

4.1.1.5 Relationships between axes

The body to stability axis relationship is defined by a rotation about the body X axis with angle of attack. The relationship is shown below:

The Earth to body frame relationship is defined by specifying the 3-2-1 Euler angles. First, a rotation about the Earth frame Z axis with the heading angle ψ , next a rotation about the new Y axis with the pitch angle θ, finally a rotation about the new X axis with the roll angle ϕ . Rotations are shown in Figure 4.1. The defined axis rotations are called as conversion matrices and shown below:

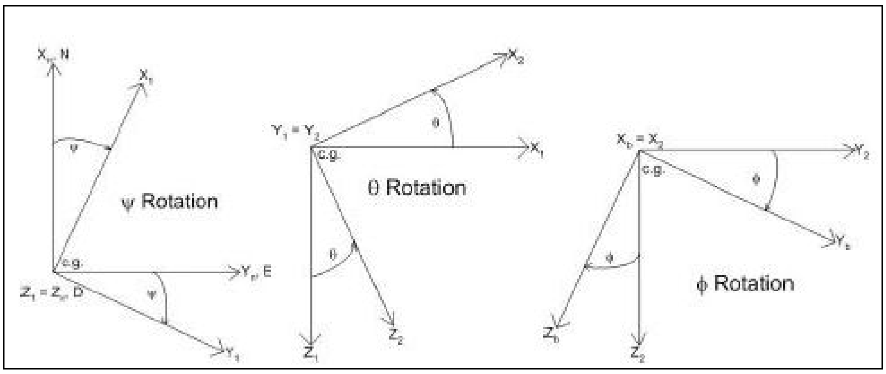


Figure 4.1 : Transformation from earth to body axes.

$$C_\psi = \begin{bmatrix} \cos\psi & -\sin\psi & 0 \\ \sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$C_\theta = \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix} \qquad \textbf{(4.1)}$$

$$C_\phi = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi & \cos\phi \end{bmatrix}$$

The combination of these conversion matrices provides the transformation matrix of earth to body frame:

$$C_B^E = C_\psi \cdot C_\theta \cdot C_\phi$$

c:cos,s:sin **(4.2)**

$$C_B^E = \begin{bmatrix} c\theta c\psi & s\phi s\theta c\psi - c\phi s\psi & c\phi s\theta c\psi + s\phi s\psi \\ c\theta s\psi & s\phi s\theta s\psi + c\phi c\psi & c\phi s\theta s\psi - s\phi c\psi \\ -s\theta & s\phi c\theta & c\phi c\theta \end{bmatrix}$$

The relationship between the Local and body frame is defined by the following two rotations. First a rotation about the local frame Z axis with the sideslip angle β, then a rotation about the new Y axis with angle of attack α. Conversion matrices of rotations are shown below:

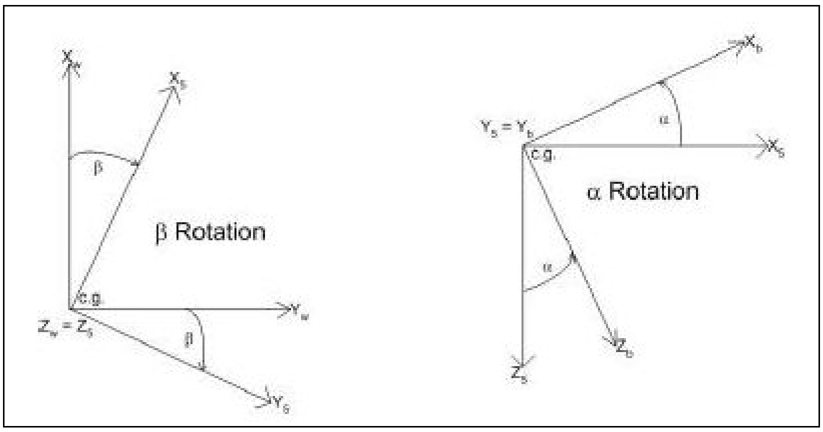


Figure 4.2 : Local to body axes transformation.

$$C_\beta = \begin{bmatrix} \cos\beta & -\sin\beta & 0 \\ \sin\beta & \cos\beta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$C_\alpha = \begin{bmatrix} \cos\alpha & 0 & \sin\alpha \\ 0 & 1 & 0 \\ -\sin\alpha & 0 & \cos\alpha \end{bmatrix} \tag{4.3}$$

The combination of these conversion matrices provides the transformation matrix of local to body frame:

$$C_B^L = C_\beta \cdot C_\alpha$$

$$C_B^L = \begin{bmatrix} \cos\alpha\cos\beta & -\sin\beta & \sin\alpha\cos\beta \\ \cos\alpha\sin\beta & \cos\beta & \sin\alpha\sin\beta \\ -\sin\alpha & 0 & \cos\alpha \end{bmatrix} \tag{4.4}$$

The earth to local frame relationship is also defined by a set of 3-2-1 Euler angles. First, a rotation about the earth frame Z axis with the course angle H, next a rotation about the new Y axis with the flight path angle γ, finally a rotation about the new X axis with velocity bank angle ϕ_v. Rotations are shown in . The conversion matrices are:

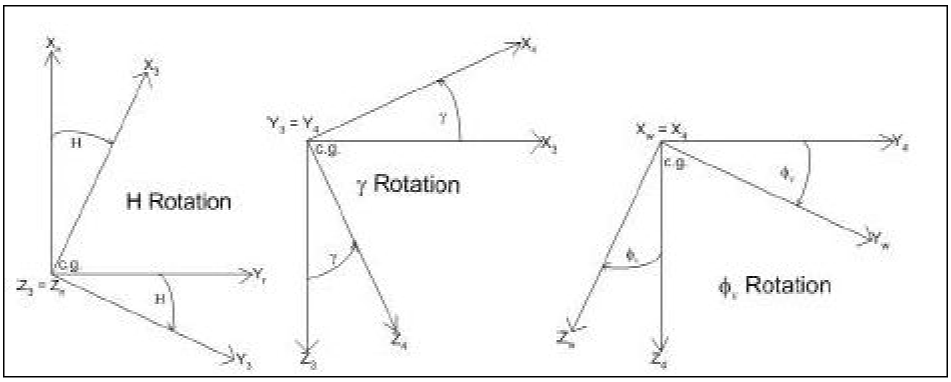


Figure 4.3 : Earth to local axes transformation.

$$C_H = \begin{bmatrix} \cos H & -\sin H & 0 \\ \sin H & \cos H & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$C\gamma = \begin{bmatrix} \cos\gamma & 0 & \sin\gamma \\ 0 & 1 & 0 \\ -\sin\gamma & 0 & \cos\gamma \end{bmatrix} \tag{4.5}$$

$$C_{\phi_V} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi_v & -\sin\phi_v \\ 0 & s in\phi_v & \cos\phi_v \end{bmatrix}$$

The combination of these conversion matrices provides the transformation matrix of earth to local frame:

$$C_L^E = C_H \cdot C_\gamma \cdot C_{\phi_v}$$

c:cos,s:sin **(4.6)**

$$C_L^E = \begin{bmatrix} c\gamma cH & s\phi_v s\gamma cH - c\phi_v sH & c\phi_v s\gamma cH + s\phi_v sH \\ c\gamma sH & s\phi_v s\gamma sH + c\phi_v cH & c\phi_v s\gamma sH - s\phi_v cH \\ -s\gamma & s\phi_v c\gamma & c\phi_v c\gamma \end{bmatrix}$$

The combination of axes rotations give the combined relationship of the tree axes as shown below, where the "s" and "c" are symbolized trigonometric functions "sine" and "cosine"

$$C_B^E = C_L^E \cdot C_B^L \tag{4.7}$$

$$\begin{bmatrix} c\theta c\psi & s\phi s\theta c\psi - c\phi s\psi & c\phi s\theta c\psi + s\phi s\psi \\ c\theta s\psi & s\phi s\theta s\psi + c\phi c\psi & c\phi s\theta s\psi - s\phi c\psi \\ -s\theta & s\phi c\theta & c\phi c\theta \end{bmatrix} =$$

$$\begin{bmatrix} c\gamma cH & s\phi_v s\gamma cH - c\phi_v sH & c\phi_v s\gamma cH + s\phi_v sH \\ c\gamma sH & s\phi_v s\gamma sH + c\phi_v cH & c\phi_v s\gamma sH - s\phi_v cH \\ -s\gamma & s\phi_v c\gamma & c\phi_v c\gamma \end{bmatrix} \bullet \tag{4.8}$$

$$\begin{bmatrix} c\alpha c\beta & -s\beta & s\alpha c\beta \\ c\alpha s\beta & c\beta & s\alpha s\beta \\ -s\alpha & 0 & c\alpha \end{bmatrix} = \begin{bmatrix} c_{11} & c_{12} & c_{13} \\ c_{21} & c_{22} & c_{23} \\ c_{31} & c_{32} & c_{33} \end{bmatrix}$$

Where;

$$c_{11} = c\gamma cHc\alpha c\beta + (s\phi_v s\gamma cH - c\phi_v sH)c\alpha s\beta - (c\phi_v s\gamma cH + s\phi_v sH)s\alpha$$
$$c_{12} = -c\gamma cHs\beta + (s\phi_v s\gamma cH - c\phi_v sH)c\beta$$
$$c_{13} = c\gamma cHs\alpha c\beta + (s\phi_v s\gamma cH - c\phi_v sH)s\alpha s\beta + (c\phi_v s\gamma cH + s\phi_v sH)c\alpha$$
$$c_{21} = c\gamma sHc\alpha c\beta + (s\phi_v s\gamma sH + c\phi_v cH)c\alpha s\beta - (c\phi_v s\gamma sH - s\phi_v cH)s\alpha$$
$$c_{22} = -c\gamma sHs\beta + (s\phi_v s\gamma sH + c\phi_v cH)c\beta \quad \textbf{(4.9)}$$
$$c_{23} = c\gamma sHs\alpha c\beta + (s\phi_v s\gamma sH + c\phi_v cH)s\alpha s\beta + (c\phi_v s\gamma sH - s\phi_v cH)c\alpha$$
$$c_{31} = -s\gamma c\alpha c\beta + s\phi_v c\gamma c\alpha s\beta - c\phi_v c\gamma s\alpha$$
$$c_{32} = s\gamma s\beta + s\phi_v c\gamma c\beta$$
$$c_{33} = -s\gamma s\alpha c\beta + s\phi_v c\gamma s\alpha s\beta + c\phi_v c\gamma c\alpha$$

4.1.2 Kinematic equations of aircraft

Aircraft differential equations can be divided into two main groups; kinematic equations related with the translation rotation of the aircraft related with linear and angular rates exerted on the body, and dynamic equations, which are related with derivatives of rates expressed in terms of forces and moments exerted on the body. In this part we will examine the kinematic relationships.

After definition of frames and coordinates, we can express the Cartesian coordinates of center of mass of aircraft as:

$$R_e = \begin{bmatrix} x_e \\ y_e \\ z_e \end{bmatrix} = \begin{bmatrix} n_p \\ e_p \\ -h \end{bmatrix} \quad \textbf{(4.10)}$$

Where the far right side shows; north position, east position and altitude (NED system). We want to relate time derivatives of position rates to kinematic variables in body frame;

$$V_b = \begin{bmatrix} U \\ V \\ W \end{bmatrix} \quad \textbf{(4.11)}$$

$$O = \begin{bmatrix} \phi \\ \theta \\ \psi \end{bmatrix} \quad \textbf{(4.12)}$$

Here (4.11) refers to velocity expressed in body frame and O is the set of Euler angles used to define body orientation.

One can use alternatively total velocity and aerodynamic angles to represent the velocity variables:

$$V_b = \begin{bmatrix} V_T \\ \alpha \\ \beta \end{bmatrix} \tag{4.13}$$

Transformation between variables is defined as:

$$\begin{aligned} U &= V_T \cos\beta \cos\alpha \\ V &= V_T \sin\beta \\ W &= V_T \cos\beta \sin\alpha \end{aligned} \tag{4.14}$$

$$\begin{aligned} \alpha &= \tan^{-1}\left(\frac{W}{U}\right) \\ \beta &= \sin^{-1}\left(\frac{V}{V_T}\right) \\ V_T &= \sqrt{U^2 + V^2 + W^2} \end{aligned} \tag{4.15}$$

(4.13) is sometimes useful for simulation purposes, because aerodynamic angles are needed all the time for aerodynamic force calculations.

To derive translational kinematics, we simply transform body variables to inertial coordinates by carrying on the matrix multiplication defined in (4.2)

$$\dot{R}_e = \begin{bmatrix} \dot{n}_p \\ \dot{e}_p \\ \dot{h} \end{bmatrix} = \mathrm{C}_B^E \mathrm{V}_B = \begin{bmatrix} \mathrm{c}\theta\mathrm{c}\psi & \mathrm{s}\phi\mathrm{s}\theta\mathrm{c}\psi - \mathrm{c}\phi\mathrm{s}\psi & \mathrm{c}\phi\mathrm{s}\theta\mathrm{c}\psi + \mathrm{s}\phi\mathrm{s}\psi \\ \mathrm{c}\theta\mathrm{s}\psi & \mathrm{s}\phi\mathrm{s}\theta\mathrm{s}\psi + \mathrm{c}\phi\mathrm{c}\psi & \mathrm{c}\phi\mathrm{s}\theta\mathrm{s}\psi - \mathrm{s}\phi\mathrm{c}\psi \\ -\mathrm{s}\theta & \mathrm{s}\phi\mathrm{c}\theta & \mathrm{c}\phi\mathrm{c}\theta \end{bmatrix} \begin{bmatrix} U \\ V \\ W \end{bmatrix} \tag{4.16}$$

(4.16) is the required differential set of equations for translational kinematics.

If the state space variables of interest are (4.13); then one can use transformation (4.15) then integrate (4.16)

Body angular rates are usually expressed as:

$$\omega_b = \begin{bmatrix} P \\ Q \\ R \end{bmatrix} \tag{4.17}$$

To find derivatives of Euler angles in terms of angular rates, we carry out sequential matrix multiplication:

$$\omega_b = \begin{bmatrix} P \\ Q \\ R \end{bmatrix} = \begin{bmatrix} \dot{\phi} \\ 0 \\ 0 \end{bmatrix} + \mathrm{C}_{\phi}\left(\begin{bmatrix} 0 \\ \dot{\theta} \\ 0 \end{bmatrix} + C_{\psi}\begin{bmatrix} 0 \\ 0 \\ \dot{\psi} \end{bmatrix}\right) \tag{4.18}$$

$$\omega_b = \begin{bmatrix} P \\ Q \\ R \end{bmatrix} = \begin{bmatrix} 1 & 0 & -s\theta \\ 0 & c\phi & s\phi c\theta \\ 0 & -s\phi & c\phi c\theta \end{bmatrix}\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \tag{4.19}$$

$$\dot{O} = \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & t\theta s\phi & t\theta c\phi \\ 0 & c\phi & -s\phi \\ 0 & s\phi / c\theta & c\phi / c\theta \end{bmatrix}\begin{bmatrix} P \\ Q \\ R \end{bmatrix} \tag{4.20}$$

Here (4.20) is the required differential equation for Euler angle set. One can alternatively use quaternion form of attitude representation. Quaternion set is a four dimensional vector which describe the orientation via Euler's principle axis theorem, three elements define vector of rotation and the last one defines rotation angle. Derivation of quaternion equations can be found in [20]. Here we only refer to kinematic differential equation:

$$\dot{q} = \begin{bmatrix} \dot{q}_0 \\ \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{bmatrix} = \begin{bmatrix} 0 & -P & -Q & R \\ P & 0 & R & -Q \\ Q & -R & 0 & P \\ R & Q & -P & 0 \end{bmatrix}\begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix} \tag{4.21}$$

Comparing (4.20) and (4.21) we see the major advantage quaternion description. Resulting differential equation is linear and has no singularities whereas (4.20) has singularities when pitch angle approaches 90 degrees. Therefore Euler angles are not suitable when simulating large attitude maneuvers which frequently occur in aggressive maneuvering. But Euler angles are very elegant to describe attitude dynamics and they are easily interpretable. In our simulations we have used (4.20) to integrate quaternions and avoid singularities then we have used transformation (4.21) to convert them to Euler angles which are used in other equations, by the help of direction cosine matrix:

$$\begin{pmatrix} \phi \\ \theta \\ \psi \end{pmatrix} = \begin{matrix} \tan 2^{-1}\left(C^E_{B\,2,3}, C^E_{B\,3,3}\right) \\ -\sin^{-1}\left(C^E_{B\,1,3}\right) \\ \tan 2^{-1}\left(C^E_{B\,1,2}, C^E_{B\,1,1}\right) \end{matrix} \tag{4.22}$$

This completes the treatment of kinematic equations.

4.1.3 Dynamic equations of aircraft

Now that we have related the velocity variables to position variables, we will now relate the derivatives of velocity variables to forces exerted on aircraft by the help of Newton' Laws. Rotational dynamics are similarly described by Euler's equations. First we will examine moments and forces acting on the aircraft then we will apply the equations.

Before we derive the equations we must make some assumptions for the aircraft model.

Aircraft behaves like a rigid body that is we neglect the aeroelastic effects. These effects are not significant up to extreme points in flight envelope, also in ongoing chapters we will show the design methods to deal with these un-modeled dynamics.

Aircrafts mass doesn't change with time so we have neglected the fuel consumption effect. This assumption is also valid for maneuver simulation which usually takes 10 seconds to 3 minutes. Aircrafts body is symmetric with respect to X and Z body frame axes. This assumption simplifies the inertia matrix and rotational dynamic calculations.

Forces acting on the aircraft can be divided into three main categories; aerodynamic forces, thrust and weight. We can express aerodynamic forces in body axes as:

$$\begin{bmatrix} A_x \\ A_y \\ A_z \end{bmatrix} = \begin{bmatrix} qSc_x \\ qSc_y \\ qSc_z \end{bmatrix} \tag{4.23}$$

In (4.23) q is the dynamic pressure, S is the wing area and c_i are the dimensionless aerodynamic coefficients.

Each aerodynamic coefficient is a function of angle of attack, sideslip angle, angular rates in body frame and control surface deflections. For F-16 aircraft specific decomposition is:

$$c_x = c_{x0}\left(\alpha,\beta,\delta_e\right)+\frac{Q\overline{c}}{2V_T}c_{xq}\left(\alpha\right)$$
$$c_y = c_{y0}\left(\alpha,\beta\right)+c_{ya}\delta_a+c_{yr}\delta_r+\frac{Pb}{2V_T}c_{yp}\left(\alpha\right)+\frac{Rb}{2V_T}c_{yr}\left(\alpha\right) \qquad \textbf{(4.24)}$$
$$c_z = c_{z0}\left(\alpha,\beta,\delta_e\right)+\frac{Q\overline{c}}{2V_T}c_{zq}\left(\alpha\right)$$

In (4.24) δ_i are the control surface deflections, $\overline{c}$ is the average chord length and b is the wing span.

When modeling an aircraft, usually it is very hard to find exact expressions for the functions shown in(4.24). Usually these values are measured form wind tunnel experiments and stored in tabular form [50]. One can always interpolate through these tables to get the coefficients. If it is necessary to express these relations in analytical form, one can always fit partial polynomials to tables, which is sometimes essential for control system design.

Weight vector is simply directed towards center of Earth, it can be converted to body frame by:

$$W = C_\theta C_\phi \begin{bmatrix} 0 \\ 0 \\ -mg \end{bmatrix} \qquad \textbf{(4.25)}$$

In (4.25) m is the mass of aircraft and g is the gravitational acceleration.

Finally thrust can expressed as simply a force term in X-Axis body frame (If aircraft has capability of thrust vectoring it can also have thrust terms in Y and Z axes).

Now we can apply Newton’s second law to gather translational dynamic equations:

$$\sum F = \frac{d}{dt}(mV) \qquad \textbf{(4.26)}$$

To take the vector derivative in (4.26) we make use of transportation theorem [20]:

$$\frac{d}{dt}(V)_I = \frac{d}{dt}(V)_B + \omega_{I/B} \otimes \mathrm{V} \qquad \textbf{(4.27)}$$

This means that, to take derivatives in inertial frame, we must take their rate of change in body frame and translate it into inertial frame by carrying vector multiplication with angular velocity vector, which is same in both frames. By inserting (4.23), (4.25) and thrust force to the left side of (4.26) and carrying out the vector algebra at right side and making use of (4.2), we reach the translational dynamic equations.

$$\begin{aligned} T + A_x - mg\sin\theta &= m(\dot{U} - RV + QW) \\ A_y + mg\sin\phi\cos\theta &= m(\dot{V} - PW + RU) \\ A_z + mg\cos\phi\cos\theta &= m(\dot{W} - QU + PV) \end{aligned} \quad \textbf{(4.28)}$$

To gather rotational dynamics we list the moments acting on the aircraft, we assume that only aerodynamic moments act on the aircraft and neglect any other effects.

$$\begin{bmatrix} M_x \\ M_y \\ M_z \end{bmatrix} = \begin{bmatrix} qSbc_l \\ qS\bar{c}c_m \\ qSbc_n \end{bmatrix} \quad \textbf{(4.29)}$$

Aerodynamic moment coefficients are expressed similar to (4.24) and stored in tabular form.

While Newton's Law relates time derivative of linear momentum to sum of forces exerted on the body, Euler's equations relates time derivative of angular momentum to sum of moments exerted on the body:

$$M = \frac{d}{dt}(I\omega) \quad \textbf{(4.30)}$$

In (4.30) I stands for 3x3 symmetric inertia matrix, along with use of (4.3) it can be expressed as:

$$\text{I} = \begin{bmatrix} I_{xx} & 0 & -I_{xz} \\ 0 & I_{yy} & 0 \\ -I_{zx} & 0 & I_{zz} \end{bmatrix} \quad \textbf{(4.31)}$$

Carrying out the vector algebra and calculus in (4.30), inserting (4.29) and (4.31) we get the rotational dynamic equations:

$$\begin{bmatrix} M_x \\ M_y \\ M_z \end{bmatrix} = \begin{bmatrix} I_{xx} & 0 & -I_{xz} \\ 0 & I_{yy} & 0 \\ -I_{zx} & 0 & I_{zz} \end{bmatrix} \begin{bmatrix} \dot{P} \\ \dot{Q} \\ \dot{R} \end{bmatrix} + \begin{bmatrix} P \\ Q \\ R \end{bmatrix} \times \left(\begin{bmatrix} I_{xx} & 0 & -I_{xz} \\ 0 & I_{yy} & 0 \\ -I_{zx} & 0 & I_{zz} \end{bmatrix} \begin{bmatrix} P \\ Q \\ R \end{bmatrix} \right) \tag{4.32}$$

4.1.4 Aircraft equations in state space form

Although we have developed complete set of equations of aircraft, they are not in the perfect condition for simulation purposes. Modern control theory has been developed under formulation of “state space” form**,** in which, whole set of nonlinear differential equations with arbitrary order, is cast as a set of first order differential equations in the form [32]:

$$\dot{X} = f\left(X, U\right) \tag{4.33}$$

This formulation has several advantages, from geometric viewpoint; it is a space in which coordinates of dynamical system are expressed in terms of its states. Therefore, for nonlinear systems one can use tools of vector calculus and algebra to interpret behavior of this system (such as input/output behavior, controllability, observability …etc). Analysis is much more easily and deeply studied, when (4.33) is in the linear form:

$$\dot{X} = AX + BU \tag{4.34}$$

Where A and B are matrices with appropriate dimensions, in this thesis we show how to deal with both linear and nonlinear systems.

A second advantage of state space formulation is simulation purposes. Inspecting the equations developed in previous sections, it is seen that they are highly coupled nonlinear differential equations, which is almost impossible to find an analytical solution. Therefore we have to rely on numerical techniques to solve them, and it is apparent that almost all differential equation solvers are developed for the set of first order equations, (such as Runge-Kutta methods) [35]. So we have to convert equations to state space form for simulations.

First we have to choose a state vector, which will specify the state of the aircraft in any given time. From previous section it is apparent to choose the vector:

$$X = \left[U \ \mathrm{V} \ \mathrm{W} \ \phi \ \theta \ \psi \ \mathrm{P} \ \mathrm{Q} \ \mathrm{R} \ \mathrm{n}_p \ \mathrm{e}_p \ \mathrm{h}\right]^T \tag{4.35}$$

Although selection of coordinates for such state space is redundant, this is the mostly used and conventional way for aircraft simulation (one can also use total velocity and aerodynamic angles (4.13) instead of body frame velocities). This selection is redundant because states heading, north position and east position (and altitude, if we neglect the change of air density with height) are just integrals of the other states, and they don't appear on right side of the equations. This is physically meaningful, aircrafts dynamics are independent of where the aircraft is heading or its position in configuration space (Cartesian coordinates). But we have to evaluate them in order to display aircrafts position and heading on a simulator. If they can be measured, they can also serve as outputs for a feedback system.

By carrying out algebra derivative of (4.35) is gathered from (4.16), (4.21), (4.28) and (4.32). Complete set of state space equations of the aircraft is presented in Table 4.1.

Examining equations of the aircraft first thing we notice is the strong coupling of equations. For general case all of the equations are related to each other somehow, for example angular accelerations integrated to find angular velocities, angular velocities are used to find Euler angle rates and these rates are integrated to find position of the aircraft, which determines velocity, and velocity directly affects the aerodynamic forces, which in turn changes the angular accelerations. Another form of coupling can be seen in moment equations, which is named inertial coupling, any angular rate in two axes produces an angular rate around the third axis (which is orthogonal to other two), which in turn effects the moment around that axis. Strength of inertial coupling is based on the inertial difference terms.

We also see that control surfaces of the aircraft (elevator, aileron and rudder) don't enter directly to equations of motion. Effects of these controls are hidden inside dimensionless coefficients. Deflection of the each surface changes the aerodynamic coefficients; effect of these controls can be seen at aerodynamic tables.

4.1.5 Atmosphere model

Since aerodynamic forces depend directly on the dynamic pressure, we have to evaluate the air density in order to calculate them. Atmosphere model takes aircraft total speed and altitude as inputs and gives dynamic pressure and Mach number as outputs. Atmosphere model is based on Standard Atmosphere 1974.

4.1.6 Engine model

In this thesis, two separate engine models were used for simulation. First model is a realistic one which is an afterburning turbofan engine modeled with first order lag. Engine is controlled via throttle setting which varies between 0-1, and total thrust is a tabular function of power level (delayed state of throttle setting), Mach number and altitude. This model is used for open loop simulations and linear control design. However it is not used in nonlinear system analysis; because it increases the order of the system (it adds engine power level as an extra state).

To simplify the model, it is assumed that a separate engine control system was designed (which is a complicated study on its own) and we can directly command thrust that can be exerted on aircraft with a first order lag. This assumption gives us chance to concentrate more on dynamic equations and neglect the engine dynamics.

4.1.7 Actuator models

Actuator models are important in aircraft simulation, because of limits for deflection and rate of control surfaces. If the control design neglects these limits, actuators may saturate which can result in failure of aircraft. All control surface deflections are modeled with 0.495 seconds of lag; limits are displayed on Table 4.2

Table 4.1 : State space represantation of aircraft equations.

Force Equations

$$\dot{U} = RV - QW - g\sin\theta + \left(A_X + T\right)/m$$
$$\dot{V} = -RU + PW + g\sin\phi\cos\theta + A_y/m$$
$$\dot{W} = QU - PV + g\cos\phi\cos\theta + A_z/m$$

Navigation Equations

$$\dot{n}_p = U\,\mathrm{c}\theta c\psi + V\left(-c\theta s\psi + s\phi s\theta c\psi\right) + W\left(s\phi s\psi + c\phi s\theta c\psi\right)$$
$$\dot{e}_p = Uc\theta s\psi + V\left(c\phi c\psi + s\phi s\theta s\psi\right) + W\left(-s\phi c\theta + c\phi s\theta s\psi\right)$$
$$\dot{h} = Us\theta - Vs\phi c\theta - Wc\phi c\theta$$

Kinematic Equations

$$\dot{\phi} = P + \tan\theta\left(Q\sin\phi + R\cos\phi\right)$$
$$\dot{\theta} = Q\cos\phi - R\sin\phi$$
$$\dot{\psi} = \left(Q\sin\phi + R\cos\phi\right)/\cos\theta$$

Moment Equations

$$\dot{P} = \left[I_{xz}\left(I_{xx} - I_{yy} + I_{zz}\right)PQ - \left(I_{zz}\left(I_{zz} - I_{yy}\right) + I_{xz}{}^2\right)QR + I_{zz}M_x + I_{xz}M_z\right]/\Gamma$$
$$\dot{Q} = \left[\left(I_{zz} - I_{xx}\right)PR - I_{xz}\left(P^2 - R^2\right) + M_y\right]/I_{yy}$$
$$\dot{R} = \left[\left(\left(I_{xx} - I_{yy}\right)I_{xx} + I_{xz}{}^2\right)PQ - I_{xz}\left(I_{xx} - I_{yy} + I_{zz}\right)QR + I_{xz}M_x + I_{xx}M_z\right]/\Gamma$$
$$\Gamma = I_{xx}I_{zz} - I_{xz}{}^2$$

Table 4.2 : Actuator limits.

	Deflection Limit (deg)	Rate Limit (deg/s)
Elevator	25	60
Ailerons	21.5	80
Rudder	30	120

4.2 Trimming algorithm

Purpose of the simulation is to examine the dynamic behavior of the aircraft at trim conditions and its response to disturbances. To do this first we must find the control inputs which puts the aircraft to trim flight for given conditions. By given conditions we mean altitude and velocity. So another program which accepts altitude and velocity as inputs, must calculate trim inputs and give outputs as throttle position, and control surface deflections. In real flight pilots continuously receives visual and intuitional clues to put the plane in a steady state. But in the case of simulation we must calculate the magnitude of inputs which will put the aircraft in steady state. This is equivalent to find control inputs which make the derivatives of the states zero. This can't be done analytically due to complex nature of nonlinear equations. So a numerical trim algorithm which minimizes the cost function (function which has the squares of the derivatives we want to equal to zero) must be developed. After writing and running the trim algorithm we can finally inspect the steady state dynamics of the aircraft.

This algorithm has been developed in MATLAB environment, due to its capability of numerical optimization routines. Subroutine "fminsearch" was used to find derivative minimizing (trimming) input and state combinations.

By using state constraints, one can also develop trimming algorithms for climbing flight, turning flight, etc…, these constraints can be found in textbooks on aircraft dynamics. Although the written trimming algorithm also calculates trim states under these conditions, we didn't need any of them other than level flight.

4.3 Simulation

Integration of state equation can be performed with well known algorithms, such as Runge-Kutta. However, SIMULINK provides more comfortable environment which is easy to interpret input-output relations, and its library provides a lot of useful tools for simulation and control.

We have two different options to construct the model in SIMULINK. We can use Fcn blocks and mathematical operations blocks to write to equations of the system and use integrator blocks to integrate the states and feedback the result to equation block as an input. So what we do really is to write the system in xdot=f(x,u) state space form. This style is used frequently in SIMULINK modeling. But to make a more compact model, instead of writing equations in blocks, we can use our .m files which were described in previous sections.

SIMULNK has a MATLAB Fcn block which uses the written .m file and accepts inputs from SIMULINK workspace. Our inputs are the states and control and output will be derivatives. With integrating the state derivatives (using integrator block) and feeding them back to function block as an input, we write the state space form in xdot=f(x,u).Finally we got the results plots in Scope blocks.

One problem faced during the construction of the model was, using MATLAB Fcn block had slowed down the simulation process drastically. That's because SIMULNK calls MATLAB interpreter at each integration step. To avoid this, an embedded block was used, which integrates the state equation for initial trim conditions.

SIMULINK diagrams are given below in Figure 4.4 and Figure 4.5:

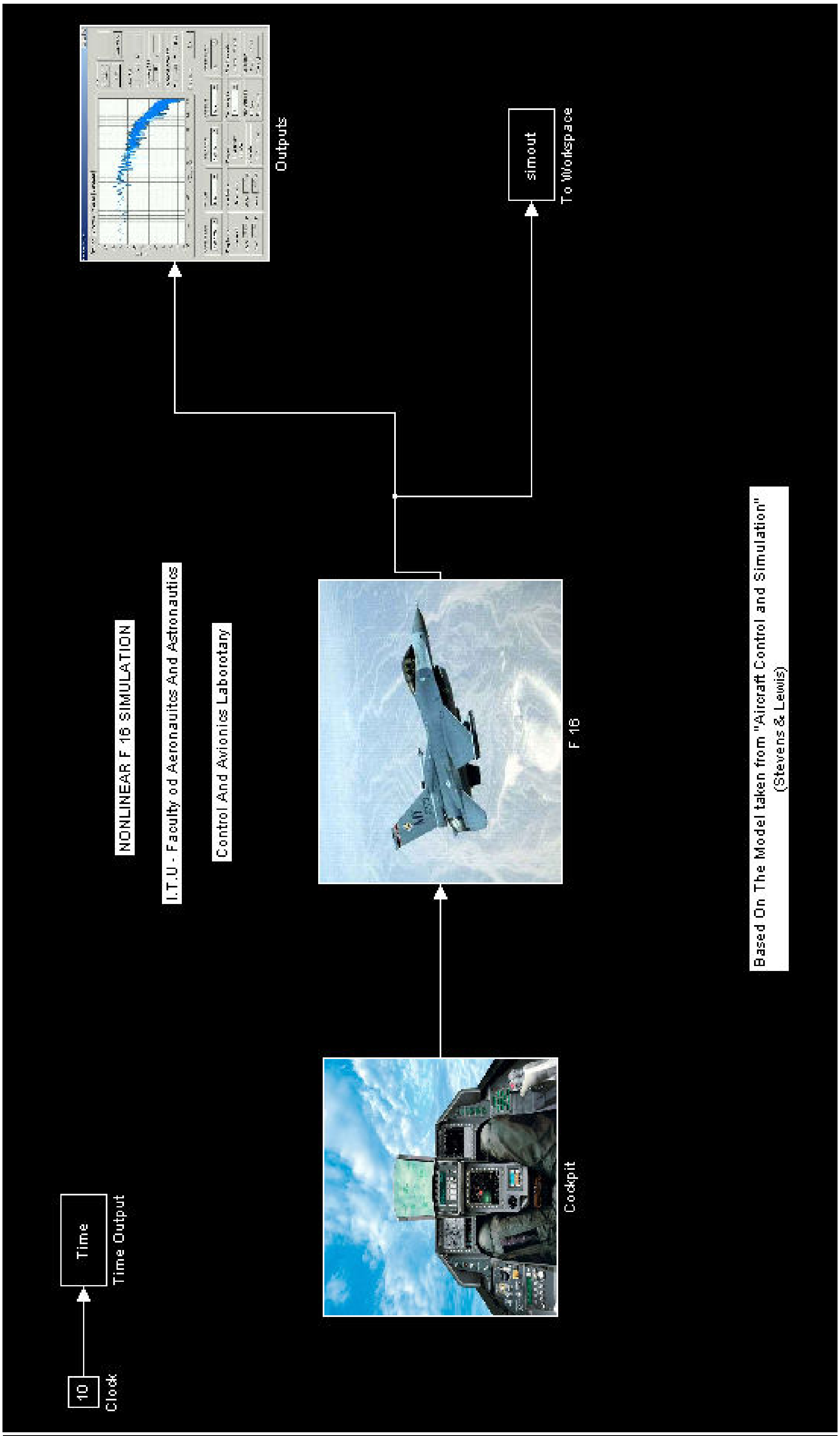


Figure 4.4 : General simulation diagram.

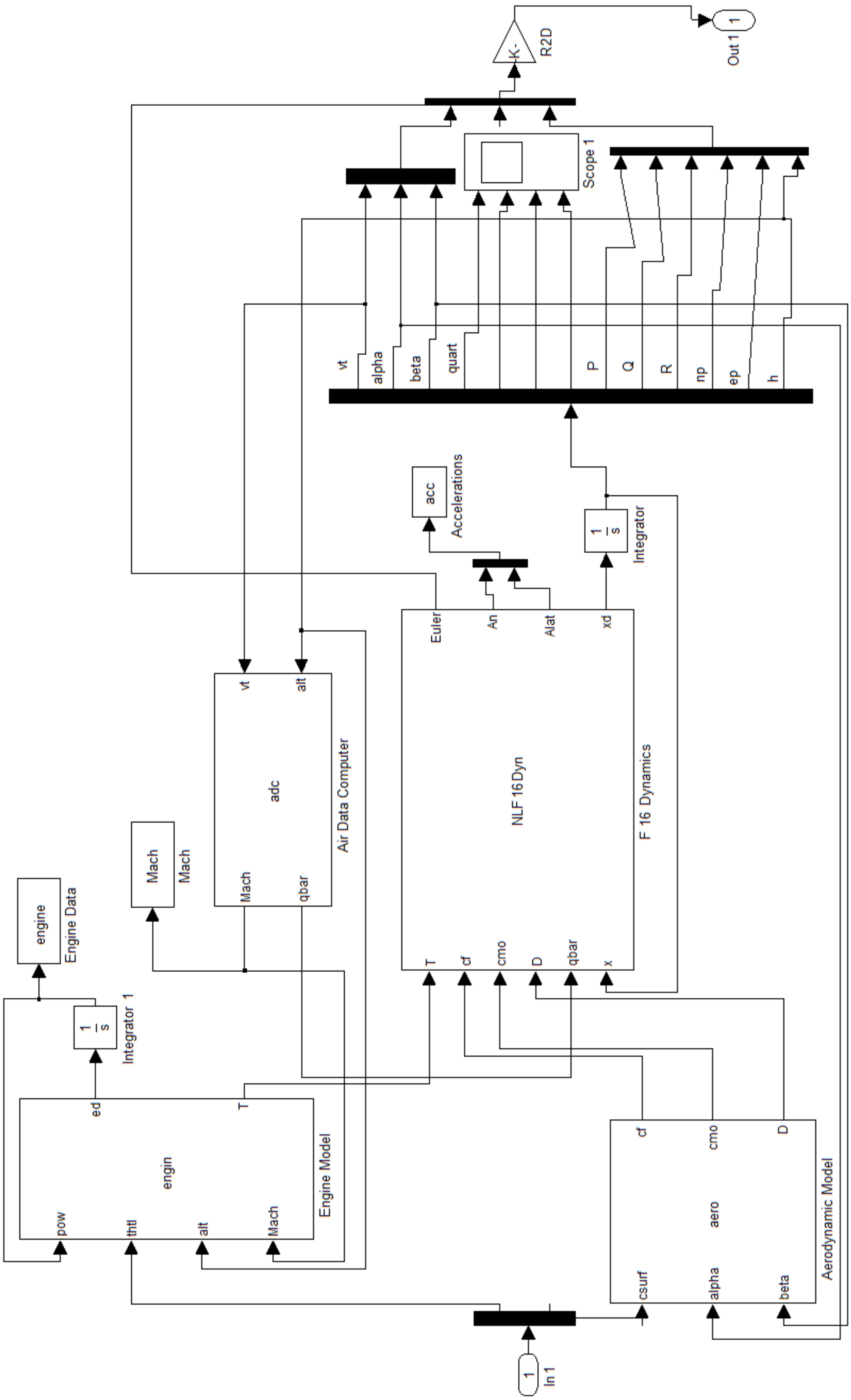


Figure 4.5 : Dynamic model and submodels is SIMULINK.

As an example, F 16 Trimmed for 502 ft/sec at sea level (0 altitude) given as initial condition we get the response displayed in Figure 4.6, Figure 4.7, Figure 4.8 and Figure 4.9

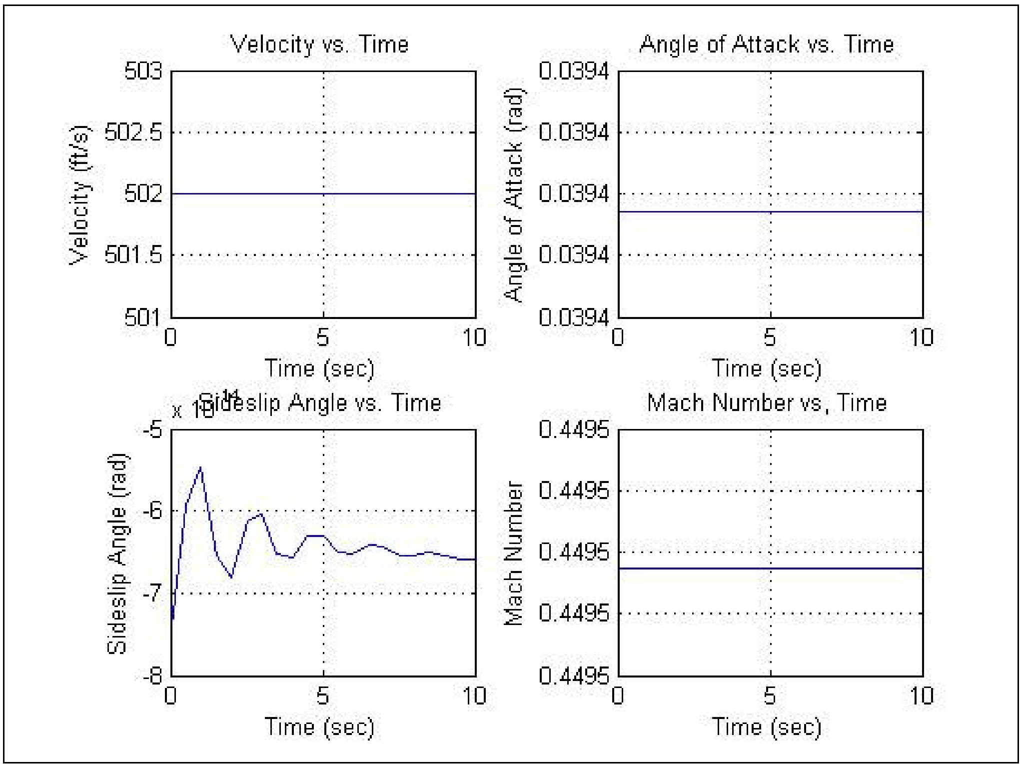


Figure 4.6 : F-16 level flight simluation – velocity variables.

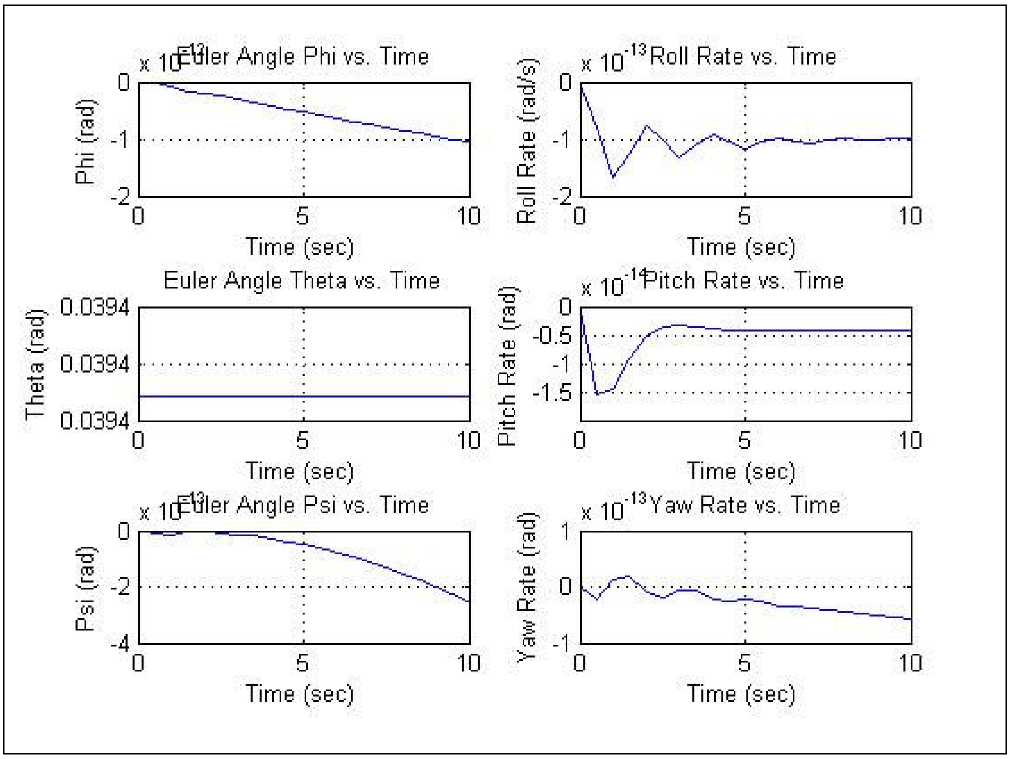


Figure 4.7 : F-16 level flight simluation – angular variables.

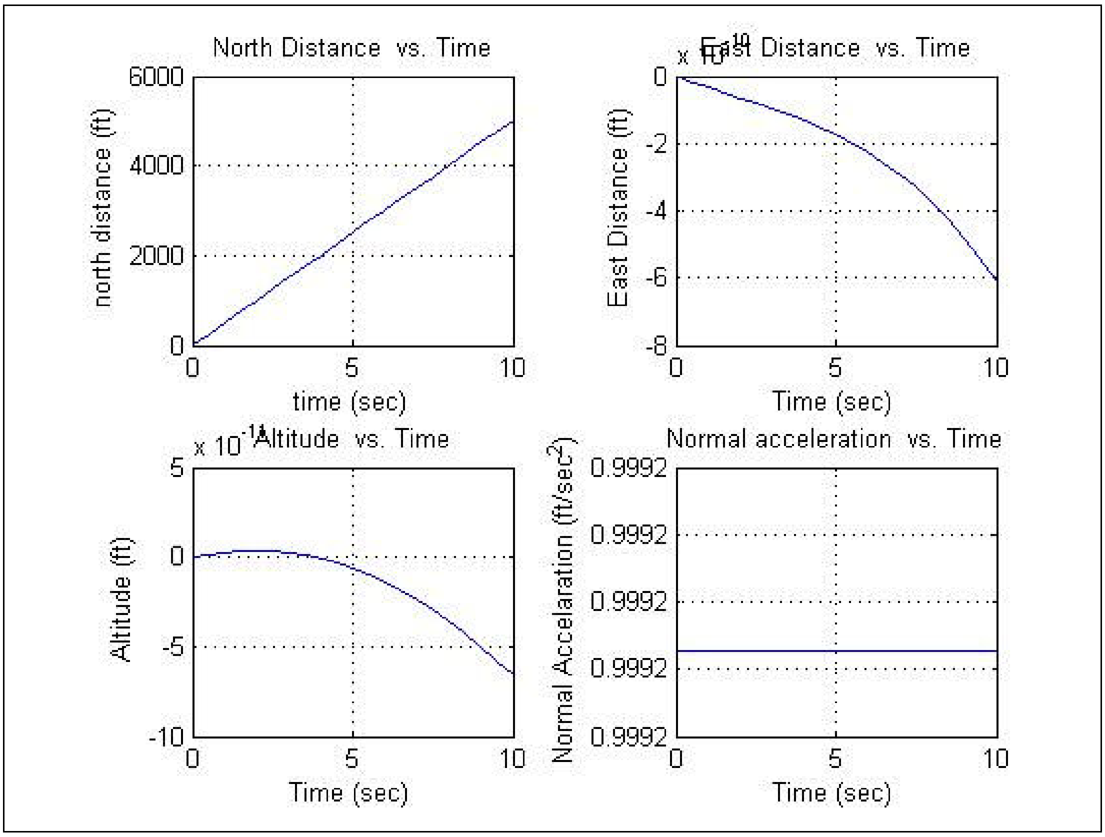


Figure 4.8 : F-16 level flight simluation – position variables.

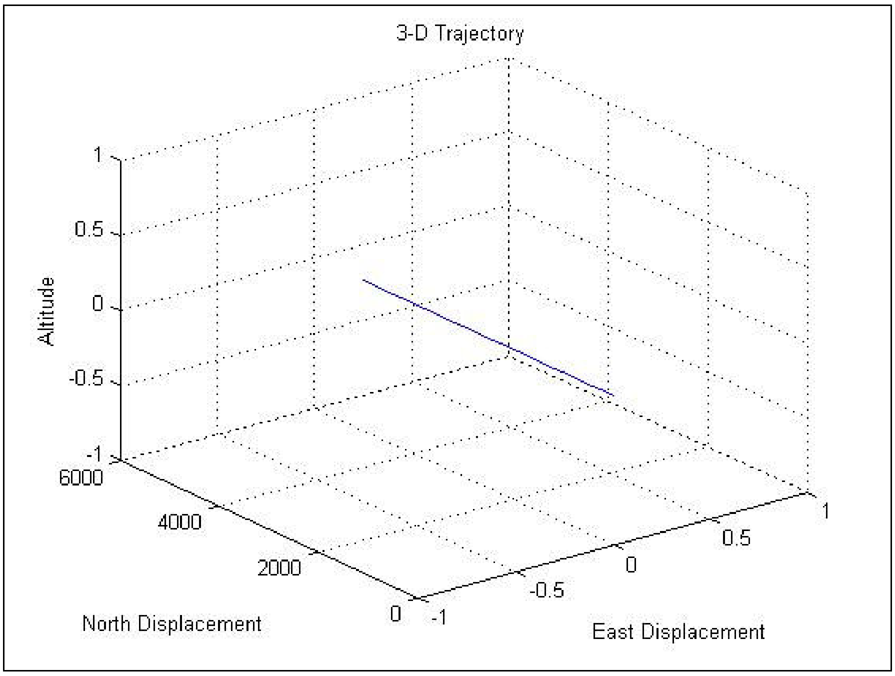


Figure 4.9 : F-16 level flight simluation – flight trajectory.

By following similar steps we can also display more complex situations, like a turning flight with a climb angle

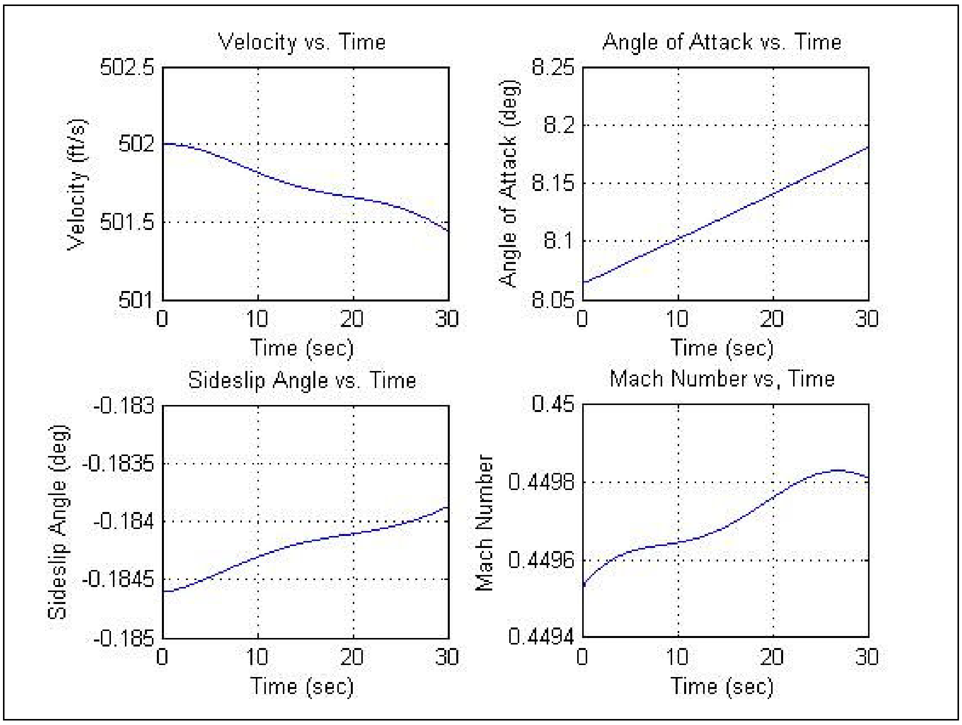


Figure 4.10 : F-16 climbing turning flight simluation – velocity variables.

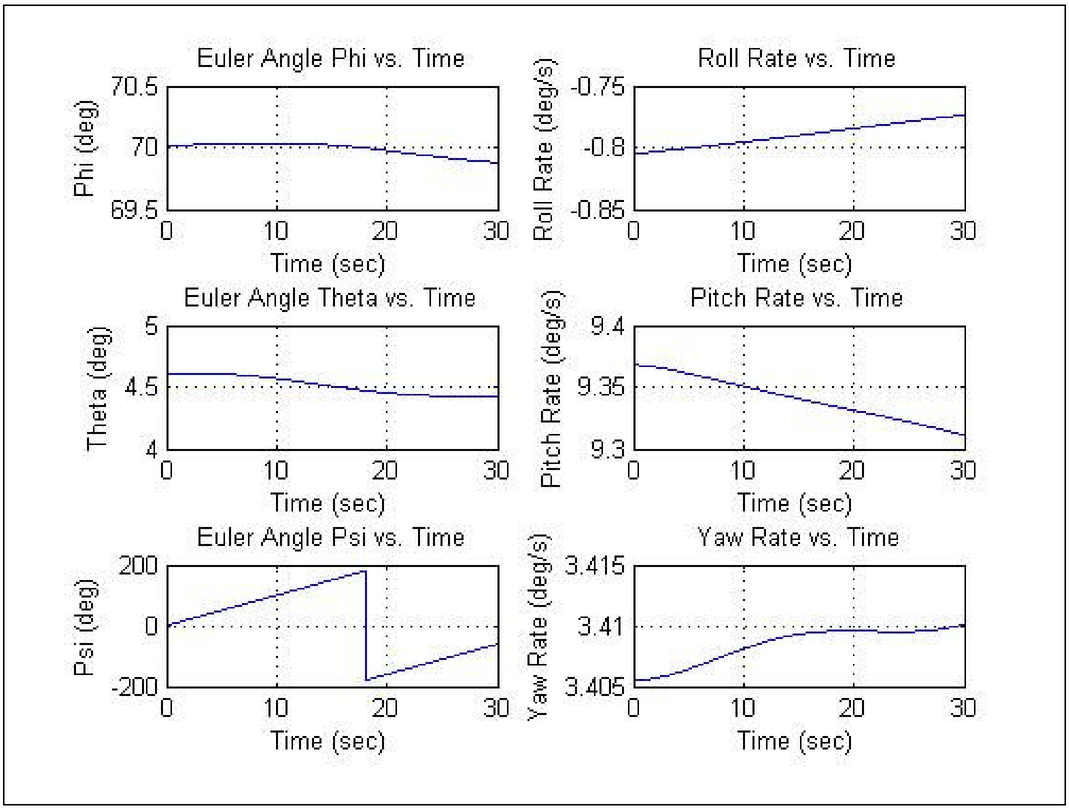


Figure 4.11 : F-16 climbing turning flight simluation – angular variables.

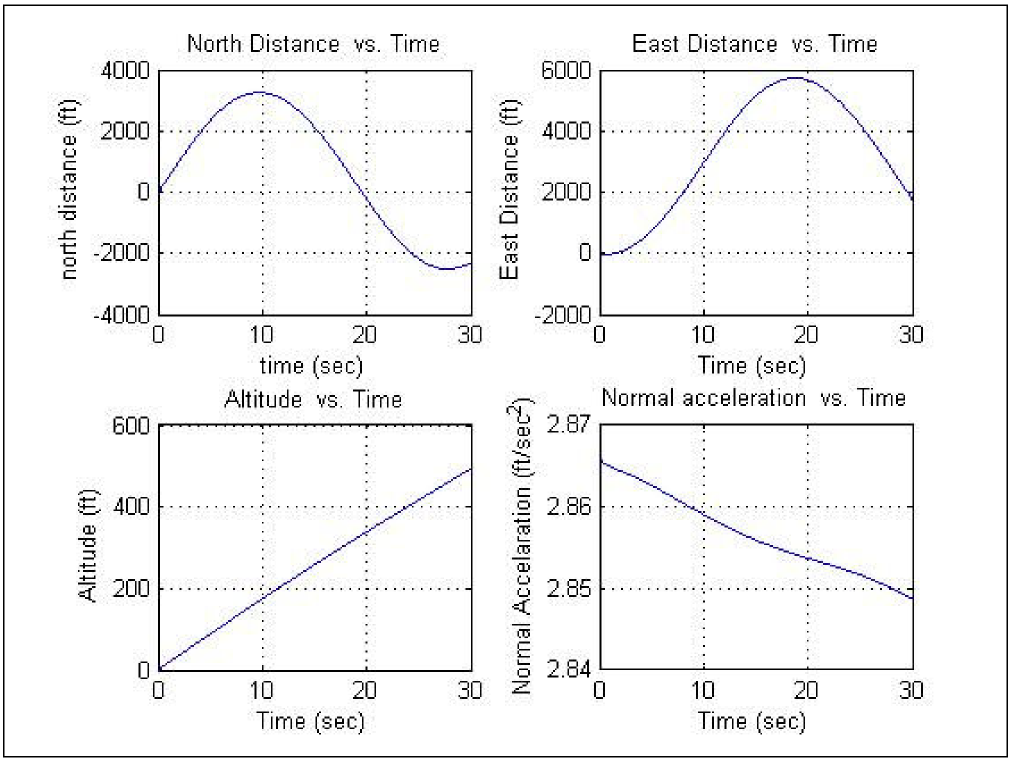


Figure 4.12 : F-16 climbing turning flight simluation – position variables.

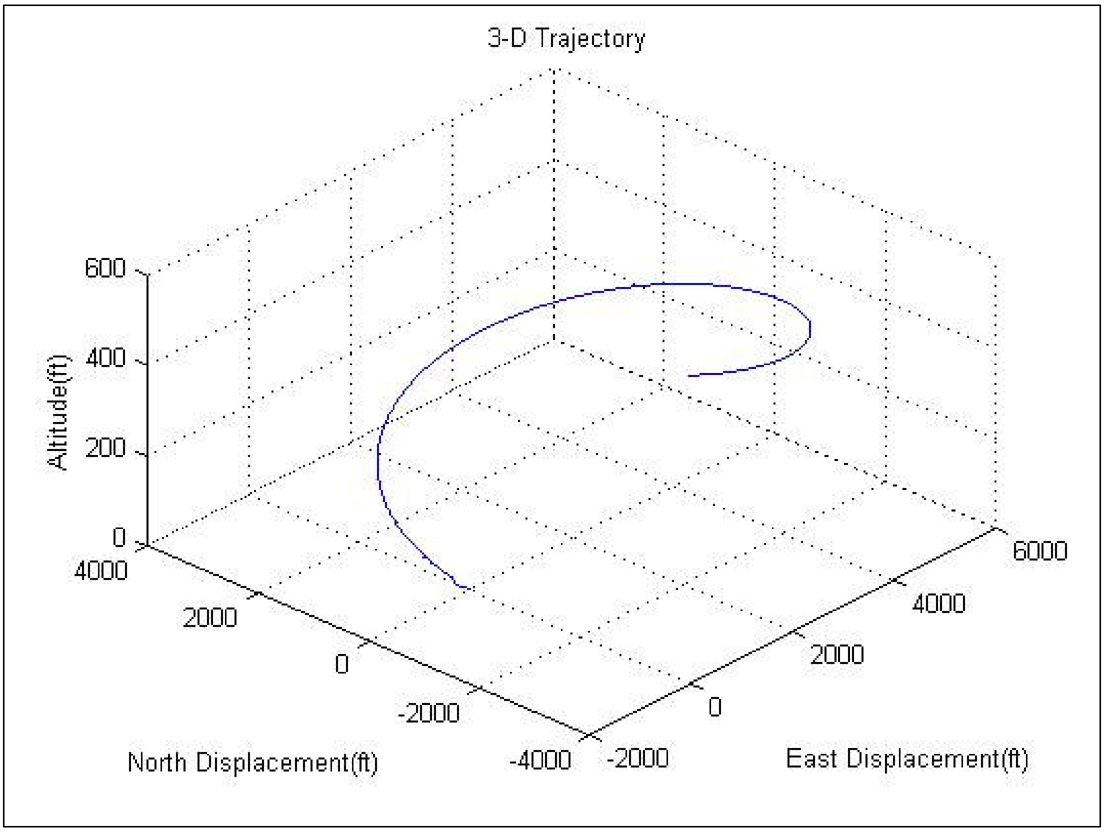


Figure 4.13 : F-16 climbing turning flight simluation – flight trajectory.

4.4 Real time simulation

Numerical simulation is adequate for control system design purposes, however to expand the simulation capability, and to simulate the behavior of the controlled airplane, simulation in real time is needed. For this purpose, SIMULINK model was compiled in Real-Time Workshop of MATLAB, and real time model was sent to an X-Pc Target. Target is a separate computer with an operating system in its own. Target creates state outputs in real time. These states are sent to visualization program Flight Gear.

Both manned/unmanned simulations can be made on this system. For manned simulation, a cockpit system consists of standard aircraft inputs and display can be used as seen in Figure 4.14. These inputs are converted to enter SIMULINK model so that pilot can control the model in real time with aid from Flight Gear. See Figure 4.15 for Flight Gear visualization.

Unmanned simulations can be made in a similar manner, by removing the pilot inputs, and replacing them by a feedback control system, which forces performance outputs to track reference signals.

Figure 4.14 : Cockpit system.

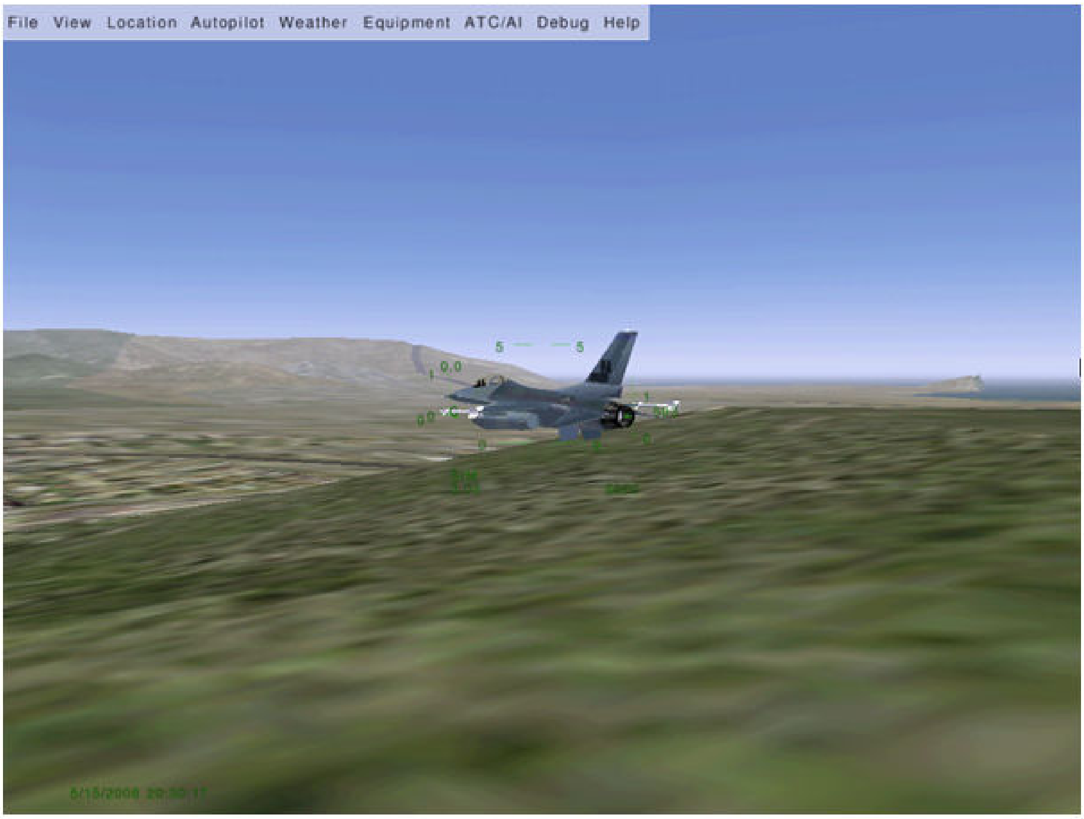


Figure 4.15 : F-16 visualized in FlightGear.

5. NONLINEAR SWITCHED CONTROL SYSTEM DESIGN

This chapter is concerned with problem of designing a low level control system for the mode based maneuvering framework introduced in Chapter 2 so that maneuver profiles generated by the algorithm in Chapter 3 would be tracked by the controller.

First section of the chapter states the inefficiency of linear control techniques for agile maneuvering and second section introduces the idea of switched control. Third section briefly defines Higher Order Sliding Mode (HOSM) techniques; which is applied in a two loop sliding mode control problem in the fourth section. Final section provides two simulations to show competence of the control system.

5.1 Limits of Linear Control

For F-16 Aircraft linear control design techniques and applications are thoroughly discussed in [50].However linear control techniques, no matter designed by which method becomes very inefficient for tracking of aggressive maneuvers.

Aggressive maneuvers have usually large amplitude reference trajectories, which have also high amplitude derivatives (very fast changing). When we want to track such trajectories linear control methods fails, because the neglect nonlinearity of the system, which is dominant characteristics of these maneuvers.

Inefficiency of linear control is shown by high amplitude, switching reference trajectory in Figure 5.1.

When we look at trajectory of the maneuvers controlled by linear and nonlinear systems; difference is more significant. Linear control simply doesn't work for aggressive maneuvering, it even de-stabilizes the system. This situation is displayed on Figure 5.2.

Due to reasons explained above, we will rely on nonlinear control techniques to build the set of controllers which are going to be defined in next section.

5.2 Multi Modal Control

Although it is theoretically possible to design one controller which tracks the inertial trajectory of the aircraft, it is much more structured and performance wise feasible to design specific control laws to each mode (or a set of similar modes). After designing such a set of controllers, control of a complete agile trajectory can be achieved by switching between these controllers following the mode transitions. Due to fact that mode transition sequence is already given by decomposed maneuver profile, transition logic of these controllers is automatically set (it is the same as mode sequence). This idea is illustrated in Figure 5.3, in which three different output tracking controllers are switched between to track the Immelmann Turn.

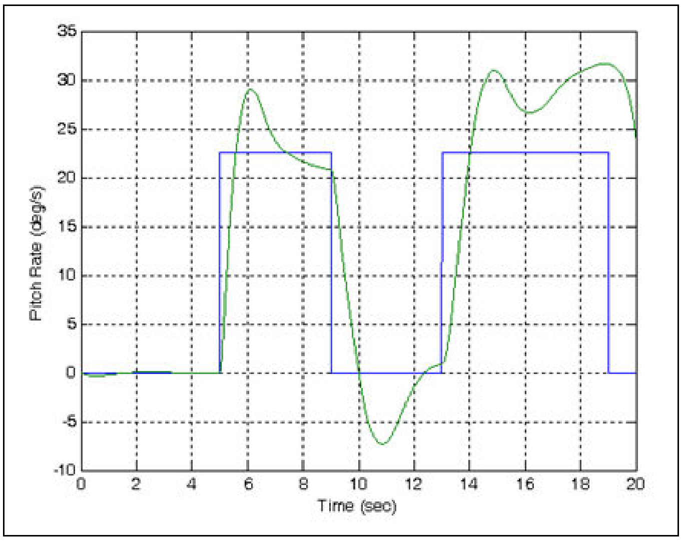


Figure 5.1 : Performance of a linear controll for agile pitch-up.

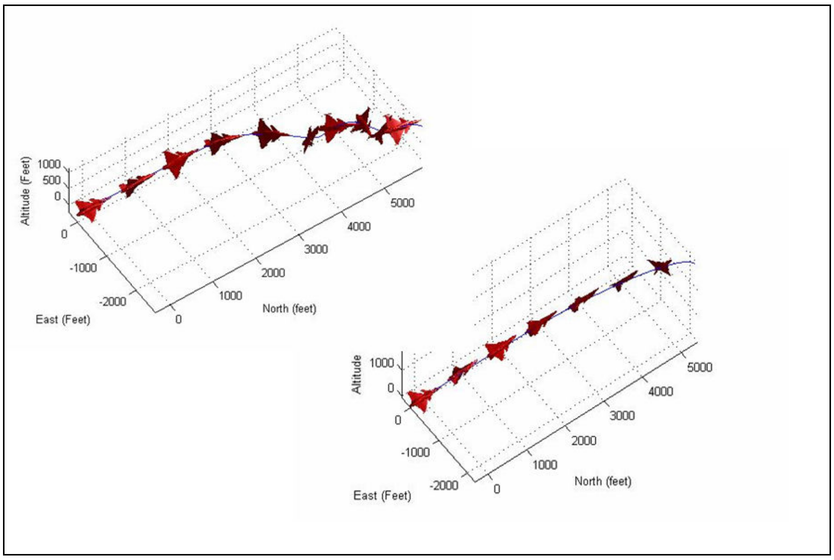


Figure 5.2 : Manuever tracking comparison for linear and non-linear control.

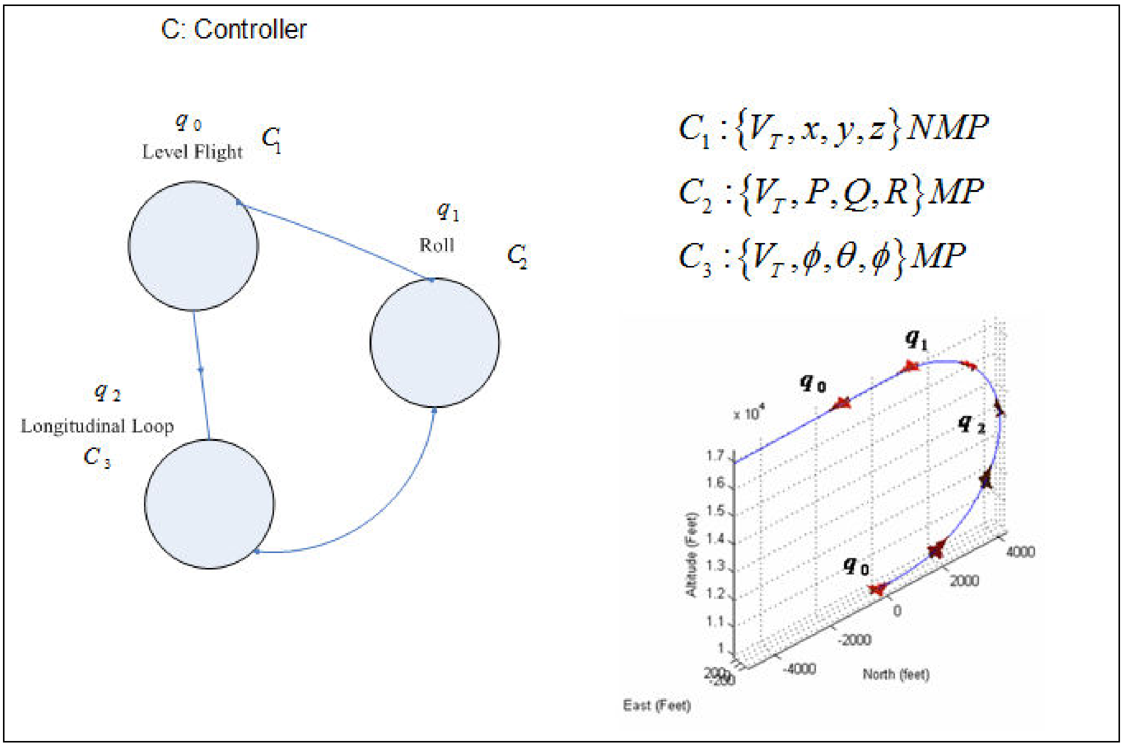


Figure 5.3 : Controller switching diagram for Immelmann Turn.

Following the maneuver descriptions and their modal inputs (maneuver parameters) in Table 2.2, Table 5.1 and Table 5.2 summarizes the output tracking controllers assigned to each mode.

Choice of the control set as defined in Table 5.1 is based on both the modal inputs and the mode constraints of each mode. For example, the roll mode controller combines both modal input integration of wind axis roll rate, and mode constraint of zero sideslip angle. In Table 5.1, although C_4 can be observed as an ultimate controller which controls any maneuver given by a flight path, the effectiveness of this controller is limited. This is based on the fact that C_4 is an NMP attitude controller which requires restrictions on Cartesian coordinates of the aircraft. This is in comparison to the MP controllers which stabilize both rotational and translational elements. Note that if every mode is locally controllable by the set of output controllers defined as above, then we can globally control any maneuver, by decomposing it into sub modes and switch between controllers at each transition step.

Table 5.1 : Mode list extended by set of output tracking controllers.

	Mode	State Constraints	Modal Inputs	Controller
q_0	Level Flight	$\dot{h}=0,\left(\dot{\phi},\dot{\theta},\dot{\psi}\right)=0$	V_T,α	C_2
q_1	Climb/Descent	$\left(\dot{\phi},\dot{\theta},\dot{\psi}\right)=0$	$V_T,\left(\dot{h},\theta_w\right)$	C_2
q_2	Roll	$\left(\dot{\theta},\dot{\psi}\right)=0$	$\int P_w dt$	C_1
q_3	Longitudinal Loop	$\left(\dot{\phi},\dot{\psi}\right)=0$	$\left(V_T,r_{loop}\right),\dot{\theta}$	C_1,C_3
q_4	Lateral Loop	$\dot{h}=0,\left(\dot{\phi},\dot{\theta}\right)=0$	$\left(V_T,r_{loop}\right),\dot{\psi}$	C_2
q_5	3D Mode	$\{\ \}$	V_T,P,Q,R $V_T,\phi_w,\theta_w,\psi_w$	C_1,C_3,C_4
q_6	Safety	$\{\ \}$	$\{0,1\}$	C_5

Table 5.2 : Set of output tracking controllers and their types.

	Controlled Variables	Type
C_1	V_T,P,Q,R	MP
C_2	$V_T,\phi_w,\theta_w,\psi_w$	NMP
C_3	V_T,ϕ,θ,ψ	MP
C_4	V_T,x,y,z	NMP
C_5	$h,\theta,\int P_w dt,\beta$	MP
C_6	$V_T,\{Quaternions\}$	MP

5.3 Higher Order Sliding Modes

In this chapter we discuss the main aspects of the higher order sliding mode control (HOSM). Chapter begins with the definition of HOSM and outlines the most important properties. Two control algorithms which provide finite reaching time to sliding surfaces are provided. Chapter ends with an example, comparing existing controller's performance with HOSM control.

5.3.1 Definition of higher order sliding modes

The classical sliding mode control (SMC) defines a sliding surface $\sigma=0$ for the dynamical system; on which dynamics of the system obeys to the desired motion (system is either stabilized or tracks the reference output signals). After selection of the sliding surface, control law is designed such that $\sigma\dot{\sigma}<0$ and sliding surface becomes attractive in selected domain (see figure 4.4). Classical sliding mode control has two main drawbacks;

First one is ;the control which maintains $\sigma = 0$ is called equivalent control, is found in the same method in feedback linearization (FL) (Canceling the nonlinearities of the system). If the system is non-minimum phase (NMP), this can lead to instability of internal dynamics of the system.

Secondly, to satisfy the constraint $\sigma\dot{\sigma} < 0$, sign of the control signal must switch according to sign of the quantity σ (which is a measure of distance to the sliding surface). This is achieved by multiplying the control law with $sign(\sigma)$. Unfortunately, when the uncertainty is present in the system, this term creates a fast switching effect which results in an issue called chattering. Chattering creates high frequency control signals which is unfeasible and destructive for actuators (see Figure 5.5)

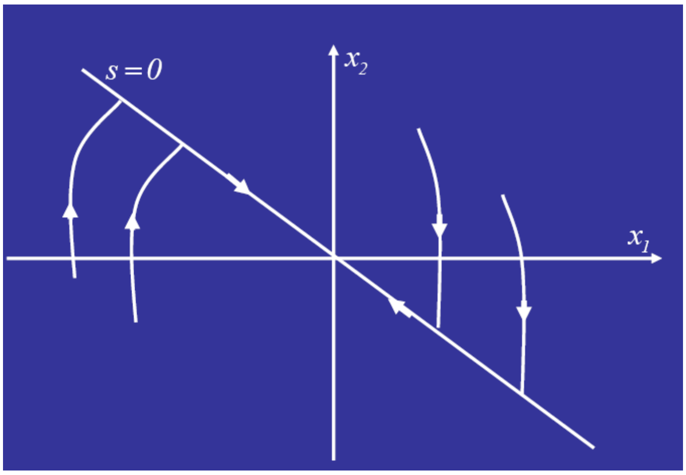


Figure 5.4 : Realization of sliding mode in two dimensions.

More on SMC and its applications can be found on [20,27]. Although SMC has two main disadvantages it is also known as one of the most effective robust controllers that could be applied to nonlinear systems. So we would like to eliminate these disadvantages while keeping the robustness and accuracy of SMC.

This idea is the main starting point of the HOSM control. In this chapter we will define HOSM as an aid for chattering and robustness but in the next chapter we will also see that it works much better in NMP systems compared to FL and SMC.

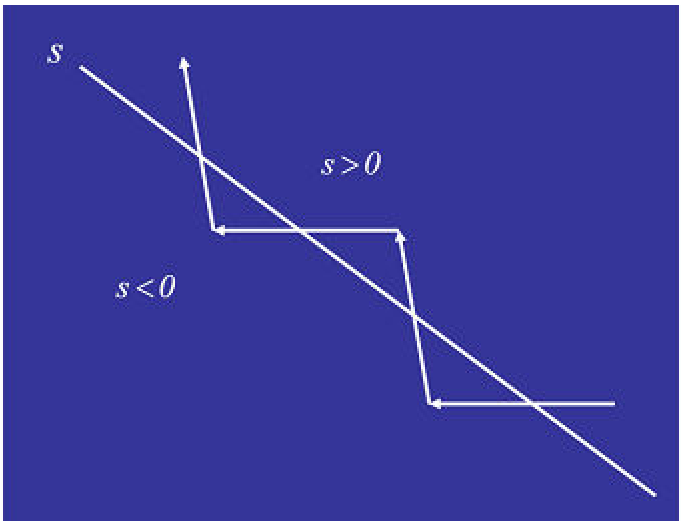


Figure 5.5 : Chattering problem.

To avoid chattering some approaches were proposed in [20]. The main idea was to change the dynamics in small vicinity of the discontinuity surface in order to avoid real discontinuity and at the same time to preserve the main properties of the whole system. However, the ultimate accuracy and robustness of the sliding mode were partially lost. Recently invented higher order sliding modes (HOSM) generalize the basic sliding mode idea, acting on the higher order time derivatives of the system deviation from the constraint instead of influencing the first deviation derivative as it happens in standard sliding modes.

Along with keeping the main advantages of the original approach, at the same time they totally remove the chattering effect and provide for even higher accuracy in realization. A number of such controllers were described in the literature (HOSM ref). HOSM is actually a movement on a discontinuity set of a dynamic system understood in Filippov's sense (Flippov). The sliding order characterizes the dynamics smoothness degree in the vicinity of the mode. If the task is to provide for keeping a constraint given by equality of a smooth function s to zero, the sliding order is a number of continuous total derivatives of s (including the zero one) in the vicinity of the sliding mode. Hence, the rth order sliding mode is determined by the equalities

$$\sigma = \dot{\sigma} = \ddot{\sigma} = \ldots = \sigma^{(r-1)} = 0 \qquad \textbf{(5.1)}$$

forming an r-dimensional condition on the state of the dynamic system. The words "rth order sliding" are often abridged to "r-sliding". HOSM control design consists of continuing to take derivatives of sliding surface and designing the control law based on the higher order derivatives of control signal such that the constraint (5.1) is satisfied.

The standard sliding mode on which most variable structure systems (VSS) are based is of the first order (s is discontinuous). While the standard modes feature finite time convergence, convergence to HOSM may be asymptotic as well, r-sliding mode realization can provide for up to the rth order of sliding precision with respect to the measurement interval. In that sense r-sliding modes play the same role in sliding mode control theory as Runge-Kutta methods in numerical integration. Note that such utmost accuracy is observed only for HOSM with finite time convergence.

In the light of above discussion we can give two important properties of HOSM control

Firstly, due to fact that switching (signum function) is moved into higher order sliding modes (or higher derivatives of control). control law is integrated and high frequency parts are cleaned, which means that chattering effect is weakened. Unlike the boundary layer methods robustness and accuracy is not lost, because control law is designed such that constraint (5.1) is kept [20].

Secondly, in the 2-sliding control constraint becomes $\sigma = \dot{\sigma} = 0$, which means that sliding accuracy is $\mathrm{O}\left(\tau^2\right)$, unlike the 1-sliding control which keeps the constraint $\sigma = 0$ by accuracy $\mathrm{O}(\tau)$. So HOSM control not only keeps removes the chattering it also improves accuracy and robustness.

In the next section, two of the most significant HOSM control algorithms will be explained.

5.3.2 Control on 2-Sliding manifolds

In this section we briefly describe the algorithms developed in[28,29]. We are interested in nonlinear systems of the form;

$$\dot{x} = f\left(x,u\right), \sigma = \sigma(x), u = u\left(t,x\right) \tag{5.2}$$

The control task is to keep output $\sigma = 0$. Differentiating successively the output variable σ, depending on the relative degree of the system, different cases should be considered depending on relative degree is one or higher (note that sliding surface can be always arranged in a form such that relative degree is one). In the case it equals to one, the classical VSS approach solves the control problem by means of 1-sliding mode control; nevertheless 2-sliding mode control can also be used in order to avoid chattering. For that purpose u will become an output of some first-order dynamic system. For example, the time derivative of the plant control u(t) may be considered as the actual control variable. A discontinuous control u steers the sliding variable s to zero, keeping $\sigma = 0$ in a 2-sliding mode, so that the plant control u is continuous and the chattering is avoided.

When considering classical VSS the control variable u(t) is a feedback designed relay output. The most direct application of 2-sliding mode control is that of attaining sliding motion on the sliding manifold by means of a continuous bounded input u(t) being a continuous output of a suitable first-order dynamic system driven by a proper discontinuous signal. Such first-order dynamics can be either inherent to the control device or specially introduced for chattering elimination purposes.

Differentiating the surface once twice, we got the following equation;

$$\dot{\sigma} = \frac{\partial}{\partial x}\sigma(x) f(x,u) \tag{5.3}$$

Differentiating twice;

$$\ddot{\sigma} = \frac{\partial}{\partial x}\dot{\sigma}(x) f(x,u) + \frac{\partial}{\partial x}\dot{\sigma}(x)\dot{u}(t) \tag{5.4}$$

The control goal for a 2-sliding mode controller is that of steering σ to zero in finite time by means of control u(t) continuously dependent on time. In order to state a rigorous control problem, the following conditions are assumed:

First, control values belong to the set $U = \{u : |u| \le U_M\}$, where $U_M \ge 1$ is a real constant; furthermore the solution of the system is well defined for all t, provided u(t) is continuous.

Second, there exists $u_1 \in (0,1)$ such that for any continuous function u(t) with $u(t) > u_1$, there is t_1, such that $\sigma u > 0$ for each $t > t_1$ Hence, the control $u = -sign(s(t_0))$, where t_0 is the initial value of time, provides hitting the manifold $\sigma = 0$ in finite time.

Third, Let $\dot{\sigma}(x,u)$ be the total time derivative of the sliding variable. There are positive constants $\sigma_0, u_0, \Gamma_m, \Gamma_M$ such that if $\sigma < \sigma_0$, then

$$0 < \Gamma_m \leq \frac{\partial}{\partial u}\dot{\sigma} \leq \Gamma_M \tag{5.5}$$

And the inequality $u > u_0$ entails $\dot{\sigma} u \geq 0$

Last of all, There is a positive constant Φ such that within the region $\sigma < \sigma_0$ the following inequality holds,

$$\left|\frac{\partial}{\partial x}\dot{\sigma} f \leq \Phi\right| \tag{5.6}$$

The above condition 2 means that starting from any point of the state space it is possible to define a proper control u(t) steering the sliding variable within a set such that the boundedness conditions on the sliding dynamics defined by conditions 3 and 4 are satisfied. In particular they state that the second time derivative of the sliding variable s, evaluated with fixed values of the control u, is uniformly bounded in a bounded domain. It follows from the theorem on implicit function that there is a function $u_{eq}(x)$ which can be considered as equivalent control satisfying the equation $\sigma = 0$. Once $\sigma = 0$ is attained, the control $u_{eq}(x)$ would provide for the exact constraint fulfillment. Conditions 3 and 4 mean that $\sigma < \sigma_0$ implies $|u_{eq}| < u_0 < 1$, and that the velocity of the $u_{eq}(x)$ changing is bounded. This provides for a possibility to approximate $u_{eq}(x)$ by a Lipschitzian control.

The unit upper bound for UQ and u\ is actually a scaling factor. Note

Also note that linear dependence on control u is not required here. The usual form of the uncertain systems dealt with by the VSS theory, i.e., systems affine in u and possibly in x, is a special case of the considered system and the corresponding constraint fulfillment problem may be reduced to the considered one.

5.3.2.1 Twisting algorithm

Let the relative degree be 1. HOSM stabilization problem is equivalent to control of second order system $\ddot{s} = \varphi(t,s,\dot{s}) + \gamma(t,s,\dot{s})u;$ with uncertain functions satisfying the bounds; $|\varphi| \le \Phi, \Gamma_m \le \gamma \le \Gamma_M$,

Following control law solves the problem;

$$\dot{u}(t) = \begin{cases} -V_M sign(\sigma) & \sigma\dot{\sigma} > 0 \\ -V_m sign(\sigma) & \sigma\dot{\sigma} \le 0 \end{cases} \tag{5.7}$$

Where the controller constants are chosen such that;

$$\begin{aligned} &V_M > V_m \\ &V_m > \frac{4\Gamma_M}{\sigma_0} \\ &V_m > \frac{\Phi}{\Gamma_m} \\ &\Gamma_m V_M - \Phi > \Gamma_M V_m + \Phi \end{aligned} \tag{5.8}$$

Inequalities in (5.8) comes from the fact that, derivative of control signal must dominate the second derivative of the sliding surface. This is achieved by switching between a larger and smaller control effort in different conditions as it is seen in (5.8) . Also note that high frequency chattering action is nested inside the derivative of control. This algorithms is called twisting algorithm because it twists the trajectories in the switching plane as it is seen on Figure 5.6.

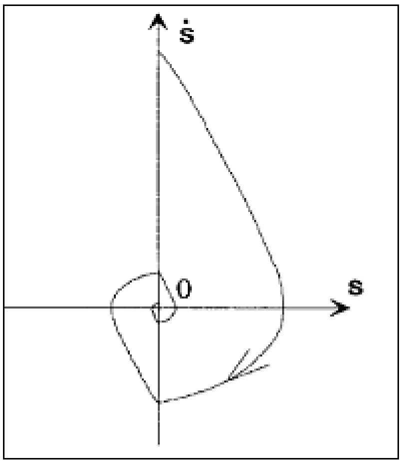


Figure 5.6 : Twisting algorithm

5.3.2.2 Super twisting algorithm

Although twisting algorithm provides 2-sliding motion it requires the measurement of both sliding quantity and its derivative. This algorithm splits the control into two parts and it requires only the measurement of sliding quantity. Control law is given by;

$$
\begin{aligned}
&u(t) = u_1(t) + u_2(t) \\
&\dot{u}_1 = \begin{cases} -ku & |u| > u_0 \\ -W\text{sign}(\sigma) & |u| \le u_0 \end{cases} \\
&u_2 = \begin{cases} -\lambda|\sigma_0|^{\rho}\, sign(\sigma_0) & |\sigma| > \sigma_0 \\ -\lambda|\sigma|^{\rho}\, sign(\sigma_0) & |\sigma| \le \sigma_0 \end{cases}
\end{aligned}
\tag{5.9}
$$

Above control law consists of one continuous and one discontinuous part. Due to fact that continuous part is more dominant the expression chattering is removed (though not completely eliminated). Main advantage of this algorithm is, it requires only the measurement of sliding surface. Sufficient conditions are given by;

$$
\begin{aligned}
&W > \frac{\Phi}{\Gamma_m} \\
&\lambda^2 \ge \frac{4\Phi}{\Gamma_m^{\ 2}} \frac{\Gamma_M (W+\Phi)}{\Gamma_m (W-\Phi)} \\
&0 < \rho \le 0.5
\end{aligned}
\tag{5.10}
$$

Super Twisting controller on 2-D plane can be seen on Figure 5.7.

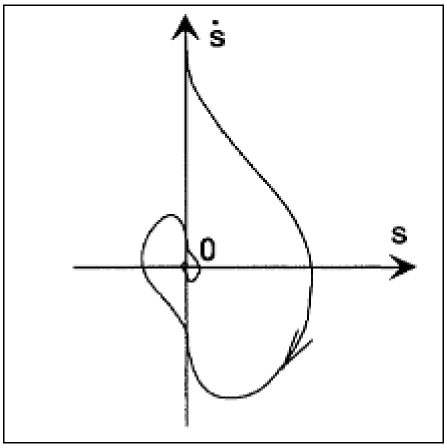


Figure 5.7 : Super twisting algorithm.

5.3.3 Robustness and performance comparisons

After introducing the HOSM controllers, now we can illustrate their capability of controlling nonlinear systems. In this section we will compare the HOSM technique with feedback linearization and sliding mode control.

System to be controlled is a simple pendulum, with external disturbance. Equations of the system are given as;

$$\begin{aligned} &\ddot{y} = -0.25\sin(y) + u + w(t), \\ &|w(t)| < 0.2, |\dot{w}(t)| < 0.1 \end{aligned} \qquad \textbf{(5.11)}$$

Where y is the angle of pendulum, u is the control (torque) input and w is the disturbance. Due nonlinear nature of the system we will seek a solution which will stabilize the system in the existence of disturbance w. Note that input is large compared to dynamics of the system and derivative of disturbance is also bounded

5.3.3.1 Feedback linearization

In feedback linearization design, one simply differentiates the output until control u appears, then u is chosen such that nonlinearities of the system cancels out, and input-output map becomes linear.

$$\begin{aligned} &\ddot{y} = -0.25\sin(y) + u \\ &u = -0.25\sin(y) + v \\ &v = k_1\dot{y} + k_2 y \end{aligned} \qquad \textbf{(5.12)}$$

In control law (5.12) coefficients are chosen such that, output equation is Hurwitz. This control law is not expected to work properly because it doesn't take into account the disturbance and relays on the approximate model.

5.3.3.2 Sliding mode control

First we define the sliding surface, then differentiate it once, then based on the, disturbance bounds and equivalent control we find the following control law:

$$\begin{aligned} &\sigma = y + \dot{y} \\ &\dot{\sigma} = \dot{y} - 0.25\sin(y) + w + u \\ &u_{eq} = 0.25\sin(y) - \dot{y} \\ &u = u_{eq} + K\sin g(\sigma), K > 0.1 \end{aligned} \qquad \textbf{(5.13)}$$

This controller is expected to wok because it is designed such that it will stabilize the system due to information on bounds of disturbance.

5.3.3.3 Higher order sliding mode control

For HOSM we differentiate the sliding surface one more, then apply the sliding algorithm (5.8) with coefficients

$$\begin{aligned} &\ddot{\sigma} = -0.25\sin(y) + \dot{\omega} + \dot{u}' - 0.25\cos(y) + u + w \\ &V_M = 7, V_m = 5 \end{aligned} \tag{5.14}$$

Comparison of these three controllers is given in Figure 5.8

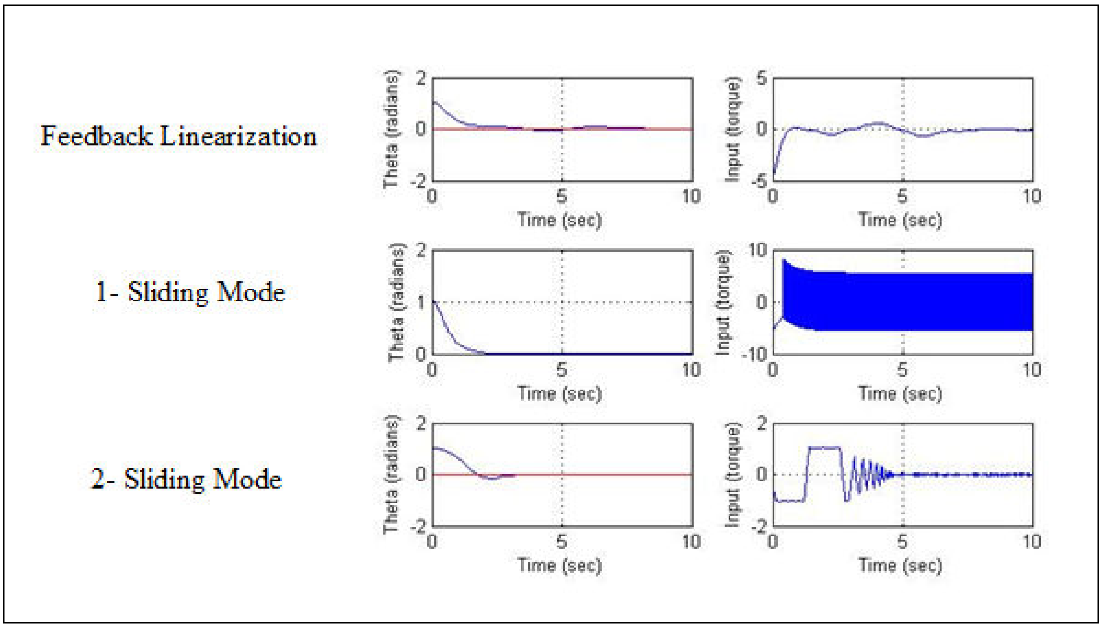


Figure 5.8 : Comparison of different control laws.

We see that feedback linearization generates smoothest control history, but its performance after settling is rather weak and oscillating, due to high amplitude disturbances. Classical Sliding Mode control (which can be also named as 1-Sliding Mode as in the figure) handles the system very well but results in a high frequency unfeasible control signal. Finally 2nd order sliding mode control stabilizes the system and keeps it steady at a moderate control price. Almost no chattering is present and controller is robust. So we can conclude that HOSM is superior to both FL and CSM at tracking systems.

5.4 Two Loop Sliding Mode Technique For Agile Maneuver Control

In this section, we represent the control system design procedure. We will develop and inner/outer loop approach. Outer loop consists of flight path variables, which were used to express maneuvers in section II. As an advantage of hybrid system description, we can design output tracking controller specific to each mode (or a set of modes). At the outer loop, tracking outputs are transformed into aerodynamic angles. This transformation is simplified in each specific case. Then body angular rates are used as inputs to aerodynamic angles. Inner loop controller regulates the control surfaces to track body angular rate profiles created by outer loop.

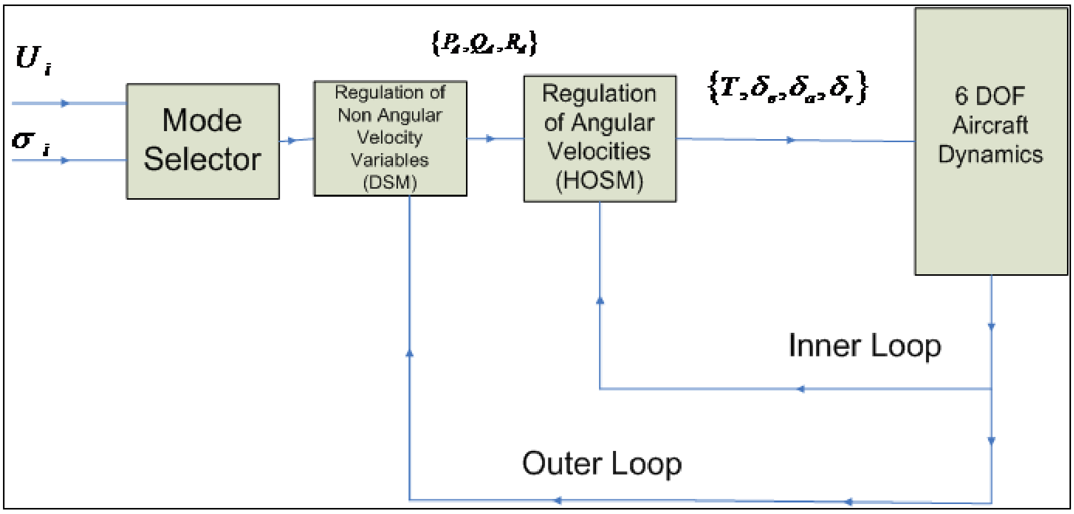


Figure 5.9 : Two loop SMC architecture.

5.4.1 Design of outer loop

Let us review the nonlinear aircraft state equations:

Equations for 6 DOF rigid body dynamics, can be founded almost every book on flight dynamics. Here we adopt the formulation from [50]. We denote aircraft states as, velocity in body axes $V_B = [U \quad V \quad W]^T$ or equivalently $V_B = [V_T \quad \alpha \quad \beta]^T$, north, east up Cartesian coordinates $R_{NEU} = [n_P \quad e_p \quad h]$, angular rates in body axes $\omega = [P \quad Q \quad R]^T$ and Euler angles $\Phi = [\phi \quad \theta \quad \psi]$. Rotation matrix $R \in SO(3)$ was used for axes transformations in the usual 3-2-1 Euler angle notation. Complete state equations describing the aircraft motion are:

$$m\left\{\dot{V}_B + \tilde{\omega} V_B\right\} = mgR(\theta)R(\phi) + F_A + T \qquad \textbf{(5.15)}$$

$$\dot{R}_{NEU} = R(\psi)R(\theta)R(\phi)V_B \qquad \textbf{(5.16)}$$

$$\dot{\Phi} = \begin{bmatrix} 1 & \tan\theta\sin\phi & \tan\theta\cos\phi \\ 0 & \cos\phi & -\sin\phi \\ 0 & \cos\theta\sin\phi & \cos\theta\cos\phi \end{bmatrix}\omega \qquad \textbf{(5.17)}$$

$$I\dot{\omega} + \tilde{\omega}I\omega = M_A \qquad \textbf{(5.18)}$$

We must note that, use of Euler angles is not convenient for agile flight since (5.17) becomes singular for 90 deg of pitch angle, which happens regularly during agile maneuvers. We avoided this singularity with use of Quaternions (Euler parameters) during simulations. Most of the controller design is based on Euler angles, but singularities are avoided during control design process. Only quaternion based controller is design is for safety mode.

These equations are valid for almost every fixed wing conventional aircraft, what makes the model specific to the aircraft modeled (F-16 in our case), is the sub-models used in simulation.

Aircraft's kinematic equations are more conveniently expressed in wind axes as follows:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{h} \end{bmatrix} = V_T \begin{bmatrix} \cos\theta_w \cos\psi_w \\ \cos\theta_w \sin\psi_w \\ \sin\theta_w \end{bmatrix} \qquad \textbf{(5.19)}$$

By using the appropriate transformation matrices, we can pass from wind axes to body axes as follows:

$$R(\psi_w, \theta_w, \phi_w) = R(\psi, \theta, \phi).R^{-1}(-\beta, \alpha, 0) \qquad \textbf{(5.20)}$$

Then we can solve the wind Euler angles as functions of the body Euler angles and aerodynamic angles. After solving (5.20), we can differentiate the equation with the help of (5.15) and (5.17). Resulting dynamical system can be written as:

$$\begin{bmatrix} \dot{V}_T \\ \dot{\psi}_w \\ \dot{\theta}_w \\ \dot{\phi}_w \end{bmatrix} = f_w(\Phi, V_B, \omega, F_A, \delta_T) = f_{w1}(\Phi, V_B) + (B_1(V_B) + ... \qquad \textbf{(5.21)}$$

$$....\Delta B_1(V_B))[\delta_T, \omega], + \overline{w}(\delta_{e,a,r})$$

Here we decomposed the nonlinear system into four functions. First term on the left side comes from the differential equations, body angular rates and throttle is accepted as an input set to this system, B_1 matrix comes from the dynamical equations(5.15), and perturbation term is added to represent uncertainty in throttle-thrust dynamics. Aerodynamic forces, which comes from(5.15), is function of both aerodynamic angles and control surface deflections.

Unfortunately if one takes in account of control surfaces deflection to aerodynamic forces, flight path variables becomes non-minimum phase, this is one of the most critical problems in nonlinear flight control because regulating control surfaces to track flight path variables results in unstable rotational dynamics, because primary function of control surfaces is to create moments around body axes. To overcome this problem, we have decomposed the aerodynamic force dependence into two parts, and accepted control surfaces effect as a disturbance on the nominal system (shown by w). Now we can use sliding mode techniques to track $(\psi_w, \theta_w, \phi_w, V_T) \rightarrow (\psi_{wd}, \theta_{wd}, \phi_{wd}, V_{Td})$

We can cast (5.21) in more compact form as:

$$\begin{aligned} &\dot{x} = f(x) + g(x)u \\ &f(x) = \hat{f} + \overline{w}, g(x) = (B_1 + \Delta B_1), u = [\delta_T, P, Q, R]^T \end{aligned} \tag{5.22}$$

Assume that,

$$\left|\overline{w}\right| < \alpha_i, \left|B_1 \Delta B_1\right| < \beta_i, i = 1, \ldots, 4 \tag{5.23}$$

Bounds on this function can be easily estimated from nonlinear simulations. Next, define the sliding surfaces in state space for each output:

$$\begin{aligned} &S_i = \left(y_{id} - y_i\right) + c_i \int \left(y_{id} - y_i\right) dt, i = 1, 2, 3, 4 \\ &y_1 = V_T, y_2 = \psi_w, y_3 = \theta_w, y_4 = \phi_w \end{aligned} \tag{5.24}$$

In (5.24) the integral terms are added to for a Hurwitz polynomial for desired dynamics and to enhance robustness. Finally by the help of MIMO sliding control theory,

$$\begin{aligned} &u = u_{eq} + K \mathrm{B}_1^{-1} \operatorname{sgn}\left(S\right), K = diag\left\{k_i\right\}, i = 1, .., 4 \\ &u_{eq} = \mathrm{B}_1^{-1} f_w \end{aligned} \tag{5.25}$$

Switching gains are chosen to ensure that they are the dominant terms in derivative of the sliding surfaces; this can be accomplished by inspecting the Lyapunov function:

$$\begin{aligned} V &= \frac{1}{2} S^T S, \dot{V} = S^T \dot{S} = S^T \left\{ f + \overline{w} + gu \right\} \\ &= S^T \left\{ \dot{y}_{id} - \left(f + \overline{w} + g(u_{eq} + K \mathrm{B}_1^{-1} \operatorname{sgn}(S) \right) + Ce \right\} \\ &= S \left\{ \mu + K(I + B_1 \Delta B_1) \operatorname{sgn}(S) \right\} \end{aligned} \quad \textbf{(5.26)}$$

If $|\mu|_i \le \gamma_i$ selection of sliding gains such that:

$$k_i > \max \frac{\gamma_i + \eta_i}{1 - \sum_{j=1}^{3} \beta_i}, i = 1, \ldots, 4 \quad \textbf{(5.27)}$$

Ensures that Lyapunov function decays to zero at finite time:

$$V = \frac{1}{2} S_i^2, \dot{V} = S_i \dot{S}_i \le \eta_i |S_i| \quad \textbf{(5.28)}$$

Now we have the required body angular rate trajectories, we must design an inner loop control law to find control surface deflections.

5.4.2 Design of inner loop

We will mainly follow same procedure in inner loop design (in the sense of choosing sliding surfaces and rendering them invariant), but unfortunately switching control laws in (5.25) cannot be directly implemented in the inner loop, due to resulting chattering will be likely to saturate actuators and cause damage to them. Control laws could be smoothed with use of saturation function instead of signum function, but this will degrade the performance. As an attractive alternative, we will use higher order sliding modes (HOSM) to eliminate chattering. Basic idea behind the HOSM is to keep higher derivatives of the sliding surface to zero along with first derivative. This results in moving the switching term inside the derivative of the input, so that when integrated no discontinuous term is present at actual input. HOSM control laws also provide better accuracy and robustness compared to conventional sliding mode methods [28].

We first consider the moment equation (5.18) and write it in a similar form to (5.22):

$$\dot{x} = f(x) + g(x)u$$
$$f(x) = \hat{f}(x) + \bar{w}, u = [\delta_e, \delta_a, \delta_r]^T \quad \textbf{(5.29)}$$

In order to put the inputs onto linear affine form, aerodynamic tables were approximated with polynomial and trigonometric functions with least squares fit. Disturbance terms accounts for the uncertainty in the aerodynamic model, which is around %10 for each aerodynamic coefficient gathered from wind tunnel tests (Deviation of functional fits from tabular data can also be count as an uncertainty). Defining the sliding surfaces similarly, we obtain:

$$S_i = (y_i - y_{id}), i = 1,2,3$$
$$y_1 = P, y_2 = Q, y_3 = R \quad \textbf{(5.30)}$$

Note that for the HOSM design the constraints that have to be satisfied are:

$$S_i = \dot{S}_i = \ddot{S}_i = 0 \quad \textbf{(5.31)}$$

Defining the differentiation operator as

$$\partial(.) = (\partial / \partial x)(.)(f + gu) \quad \textbf{(5.32)}$$

And differentiating the sliding surfaces two times, we obtain:

$$\ddot{S}_i = \partial\partial S_i + \left(\partial \dot{S}_i / \partial u\right)\dot{u} \quad \textbf{(5.33)}$$

To accomplish the constraints in equation (5.31), the derivative of the input term in (5.33) should become the dominant term. The super-twisting algorithm defined in previous subsection achieves this. Control law is defined as:

$$u = u_{static} + u_{dynamic}$$
$$u_{static} = \begin{cases} -\lambda |S_{i0}|^p \operatorname{sgn}(S_i) & |\mathrm{S}_i| \geq |S_{i0}| \\ -\lambda |S_i|^p \operatorname{sgn}(S_i) & |\mathrm{S}_i| < |S_{i0}| \end{cases} \quad \textbf{(5.34)}$$
$$\dot{u}_{dynamic} = \begin{cases} -u & |u| \geq |u_0| \\ -\alpha sgn(S_i) & |u| < |u_0| \end{cases}$$

In (5.34), undefined constants are derived from the bounds on second derivatives of the nonlinear functions *f* and *g*. These constants can be estimated from nonlinear simulations or they can be tuned// during test of the control system. Note that (5.34) is in the SISO form, but it can be extended to MIMO form with an invertible input transformation $v = \zeta\delta, \zeta \in R^{3x3}$, which decouples the system so that only one input

v_i appears at each sliding surface equation. Then (5.34) is applied to each surface separately and the true inputs are recovered from invertible transformation. The resulting control law can be written as:

$$\delta = \zeta^{-1}\left(v_{static} + v_{dynamic}\right) \qquad \textbf{(5.35)}$$

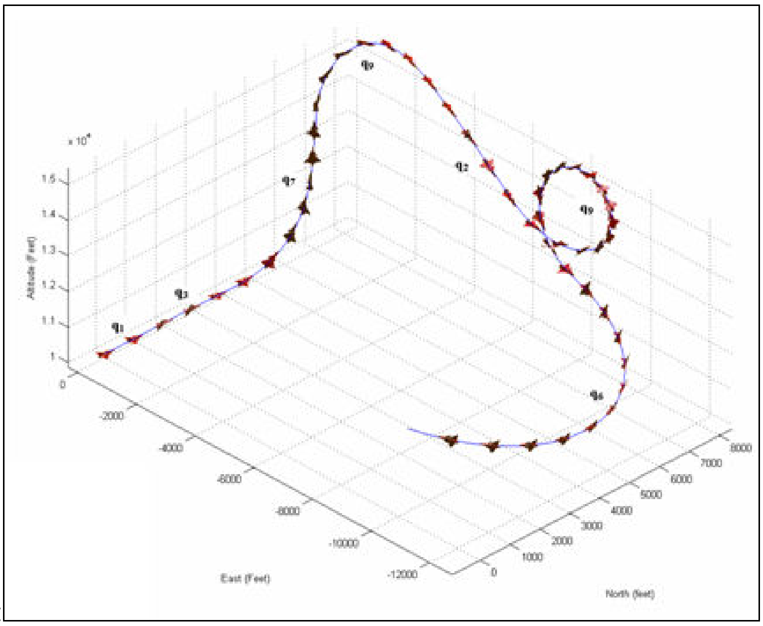
This completes the design of the inner loop control system.

5.5 Agile Maneuvering Control Simulations

In this section two numerical examples are provided to present the capability of the switched control system. In the first example control of a multi modal agile combat maneuver is considered. Second simulation shows the integration of control system with maneuver and path planner in hierarchical setting.

5.5.1 Agile combat maneuver control

To display the capacity of the control system, we will show tracking of an agile combat maneuver inspired from [38]. First maneuver is inspected formally, and decomposed in to modal block sequences as described in Chapter 3. Then mode sequences are transformed to tracking profiles for sliding mode controller. Simulation model is high fidelity F-16 model described in Chapter 4. All control laws (constants and analytic expressions) and agility metrics were computed from this model. Maneuver is displayed in Figure 5.10:



:

Figure 5.10 : Modally decomposed agile combat maneuver.

Key agility elements are displayed in Figure 5.11 and Figure 5.12. We observe that angle of attack has been varied between -1 to 30 degrees during maneuver, this regime is has highly uncertain dynamics (we have added around %30 uncertainty to aerodynamic coefficients in this regime). Load factor also jumps around 8 “g”s, which is almost impossible to overcome by a human pilot. This result also shows that fully automated UCAVs would be a better choice for future air combat.

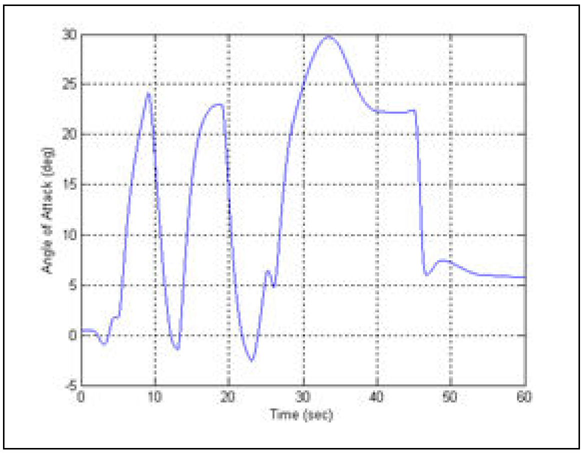


Figure 5.11 : Angle of attack history during maneuver.

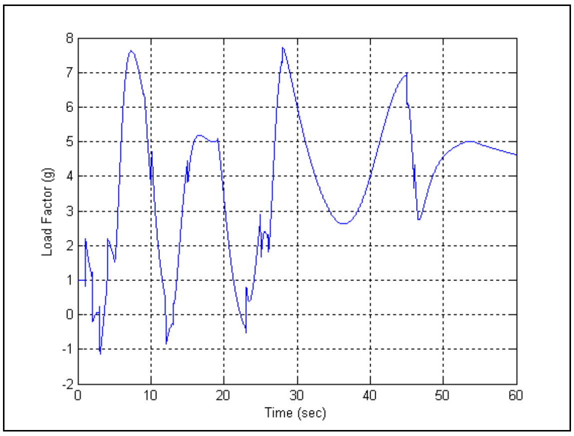


Figure 5.12 : Load factor history during maneuver.

Performance of control systems is displayed inFigure 5.13. We see that inner loop had done its job in tracking body angular rates almost perfectly. Inputs didn’t saturate during maneuver (this is actually due to maneuver synthesis system which selects feasible maneuver sets) and there is no chattering in any channel due to HOSM. We can conclude that control system has been successful for tracking a highly nonlinear agile maneuver in presence of uncertain effects.

5.5.2 Integrated path/maneuver planning and control

For simulation purposes we consider an example 200×200×200 unit cube complex city-like environment. In first part; path planning layer, constructs a 3D flight trajectory which avoids the obstacles while satisfying the velocity and acceleration constraints (by checking the flight envelope). This flight trajectory with velocity is given to maneuver planning layer. Example solution of the Path Planner in the 3D MelCity model environment is seen in Figure 5.14.

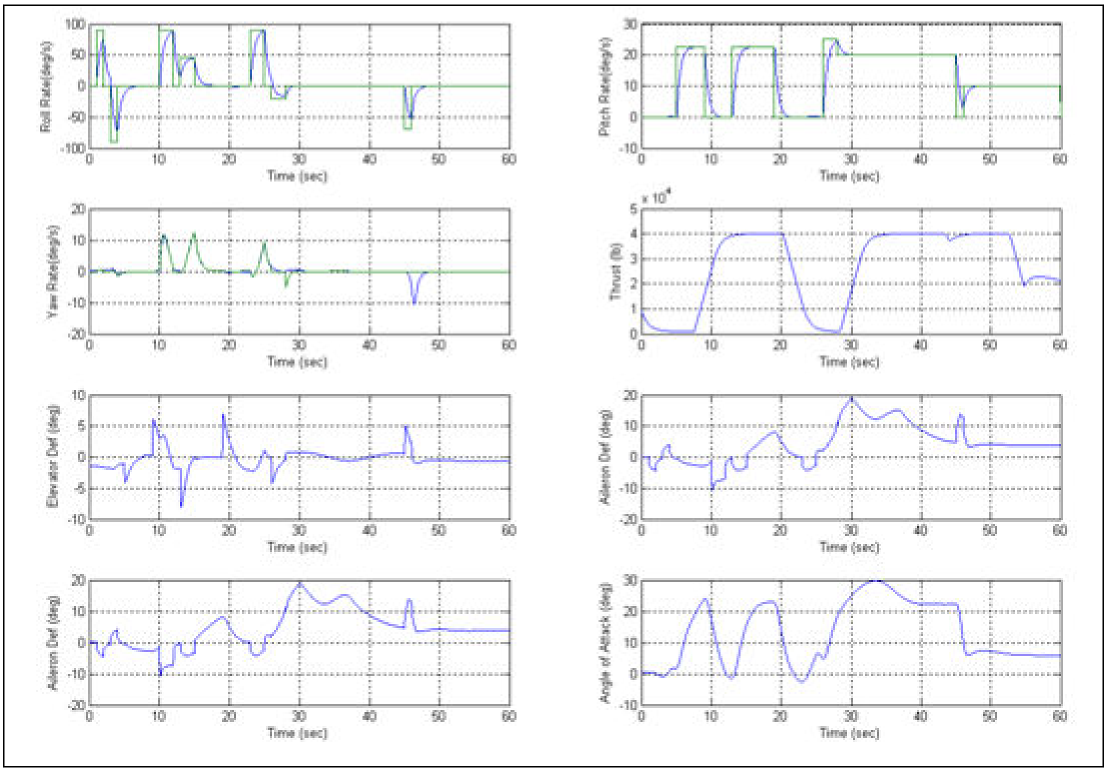


Figure 5.13 : Tracking files and input history.

Maneuver planning algorithm decomposes the path into flight modes in multimodal control framework and derives the feasible modal sequence based on the search of agility metric graphs and mode transition rules. Timetable of the maneuver sequences is seen in Table 5.3. Three transition modes are placed between other modes to obtain the attitude continuity.

Performance of the control system is given in Figure 5.15 and Figure 5.16. Similar to the previous example, angular tracking rates are successfully tracked and input command stays within envelope limits due to feasible modal input selections in maneuver planning part.

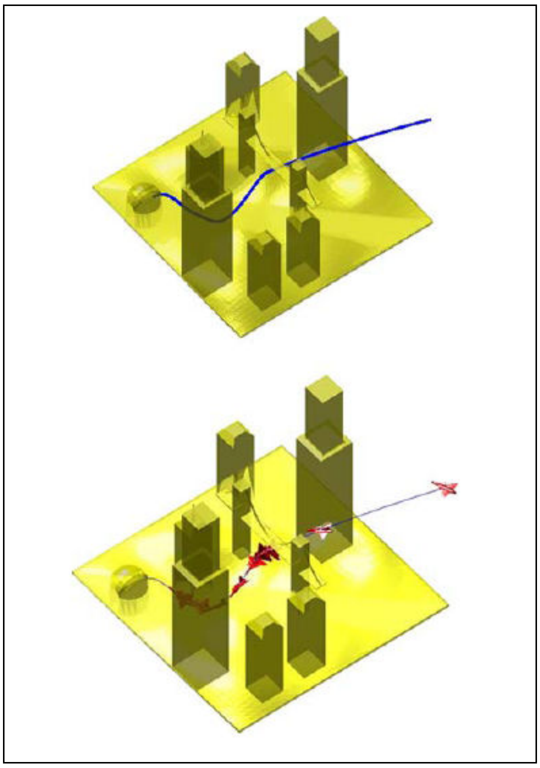


Figure 5.14 : Step by step solutions by path and maneuver planner.

Table 5.3 : Maneuver Mode Sequence.

Mode	Label	Time intervals (sec)
q_0	Level flight	[0, 0.83]
q_3	Longitudinal loop (transition)	[0.83, 1]
q_1	Dive	[1, 3.57]
q_2	Roll mode (transition)	[3.57, 4.57]
q_4	Lateral loop	[4.57, 26.7]
q_2	Roll mode (transition)	[26.7, 27.7]
q_0	Level flight	[27.7, 30.3]

To get a better understanding of what we have gained by this integrated architecture, two extra simulations were done. In the first simulation, path planning step has no flight envelope feasibility check, which results in breakdown in maneuver planning section, because the algorithm fails to compute feasible modal inputs from agility metric graphs and velocity given by path planner is out of range, so no results could be obtained from this simulation. This shows that flight envelope check is a critical part of the integrated architecture.

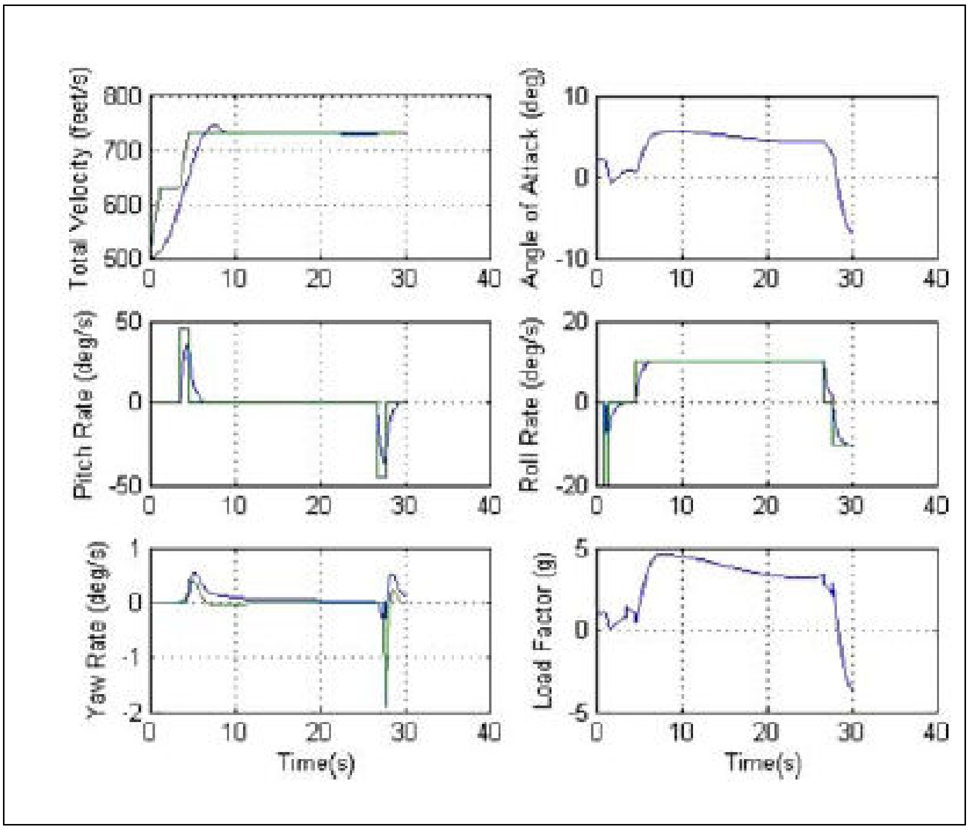


Figure 5.15 : Angular rate tracking performance

In second simulation, no maneuver planning is used; trajectory from path planner is directly given to a trajectory tracking control system. Results of this simulation is shown in Figure 5.17. This simulation results show that, attitude is unstable on flight path and there are deviations from the trajectory, which has two reasons. First, nonlinear controllers based on trajectory references are likely to become unstable on agile trajectories while multi modal control framework uses stable controllers for each mode [34]. Second reason is the absence of maneuver planning part; since there isn't any feasibility check on the agility metrics and attitude continuity, it is very reasonable to get unstable attitude and deviations from flight path due to saturated inputs, which have been supported by the simulation results. Overall, these simulation results show that, when they are isolated path and maneuver planning layers have critical defects. They must be integrated together to get a feasible and controllable flight path.

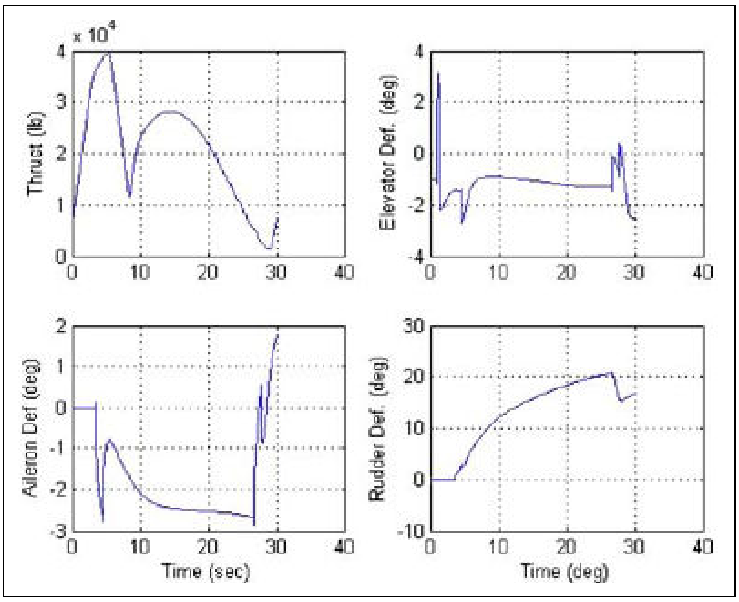


Figure 5.16 : Input time history.

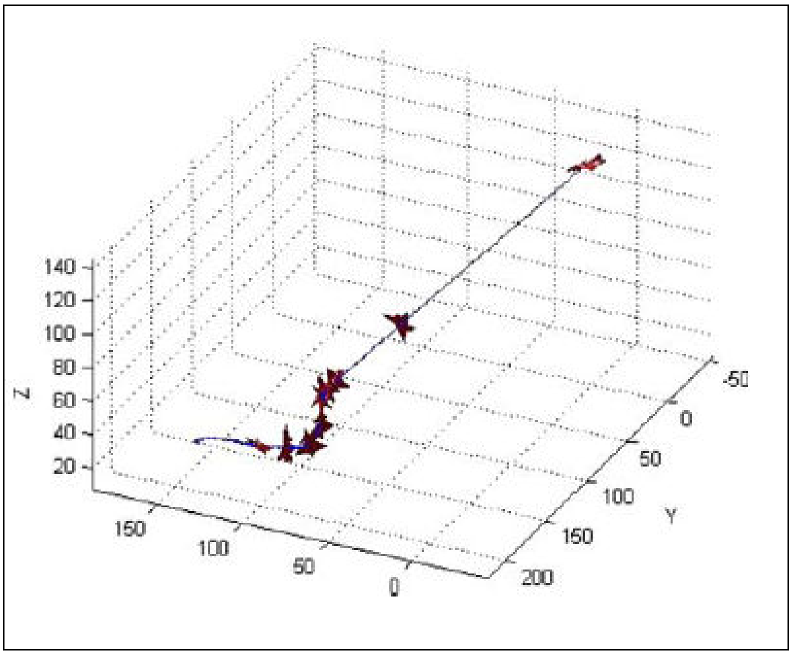


Figure 5.17 : Direct trajectory planning without maneuver planning.

6. CONCLUSIONS AND FUTURE WORKS

In this work we have laid out a multi modal control framework suited for UCAVS, which quantizes the flight maneuver into discrete flight modes and parameters. Decomposition methodology has been shown by examples. Maneuver generation constraints are also discussed in terms of sequential and envelope constraints, which have been overcome by use of a set of transition rules and agility metrics respectively. Lastly, local tracking output of each mode is discussed and system is converted into a hybrid system for future research. Overall system, which is named as Mode Based Maneuvering Automaton has the capability of decomposing and generating complex feasible agile maneuver profiles which are tracked by nonlinear controllers assigned to each mode. System has ability to track every maneuver which has been generated by this system (full flight envelope control) as opposed to control limited set of pre-defined flight trajectories in previous researches.

On the planning side; maneuver identification and generation algorithms based on the multi modal control framework were developed. Identification algorithm showed the capability of the proposed modal decomposition methodology and the fact that state space description is indeed replaceable by the modal sequence. This algorithm also helped to develop the more critical algorithm "Feasible Maneuver Generation Algorithm". Simulation results of the algorithm showed that, when integrated with a path planner, overall system is capable of generating a continuous (in the sense that it satisfies sequential constraints), feasible (in the sense that modal input satisfies envelope constraints via agility metrics) modal sequence, which can be controlled globally via switching control system (multi modal control framework).

On the control part; we have developed a set of low level controllers which can track given maneuver profiles in six degrees of freedom agile flight. The control system has been developed on modes based on the hybrid representation of the aircraft dynamics. Overall system has been numerically tested on a realistic F-16 model, and the simulations demonstrate that the system has the capability of performing complex maneuver segments.

REFERENCES

[1] **Chapman E. R.**, 2002. Unmanned Combat Aerial Vehicles: Dawn of a New Age? , *Aerospace Power Journal*, **16**, 60-72.

[2] **Tiron R.**, 2006. Can UAVs Dogfight? , *Association for Unmanned Vehicle Systems International: Unmanned Systems*, **24**, 39–42.

[3] **Francis, M. S.**, (2008), JUCAS Program Update, DARPA, USA, http://www.darpa.mil/j-ucas/, 02.05.2010.

[4] **Inalhan G.**, 2007. *NATO SCI-186 C2 Architectures for Operations with Manned and Unmanned Air Vehicles,* (NATO-RTA), Paris, France.

[5] **Defense Science Board**. 2006, *Unmanned Aerial Vehicles and Uninhabited Combat Aerial Vehicles*, Department of Defense, USA.

[6] **Koyuncu, E. and Inalhan, G.**, 2010. Dynamically Feasible Probabilitis Motion Planning in Complex Environment for UAVs, in *Robotics 2010 in Current and Future Challenges*, Edited by Houssem Abdellatif, InTech, Vienna, Austria.

[7] **Frazzoli E., Dahleh M. A. and Feron E.,** 2002. Real Time Motion Planning for Agile Autonomous Vehicles, *Journal of Guidance, Control, and Dynamics*, **25**, 116-129.

[8] **Brockett R. W.**, 1990. Formal Languages for motion description and map making, *Symposia in Applied Mathematics*, Louisville, Kentucky, USA, January 16-17.

[9] **Manikonda V., Krishnaprasad P S. and Hendler J.**, 1998. Languages, behaviours, hybrid architectures and motion control, in *Mathematical Control Theory*, Edited by Jailleul J. And Willems J. C., Springer, New York, USA.

[10] **Fainekos G., Gazit H. K. and Pappas G. J.**, 2006. Hybrid controllers for path planning: a temporal logic approach, *IEEE Conference on Decision and Control*, Seville, Spain, December 10-12.

[11] **Tomlin C., Pappas G. J. and Sastry S.**, 1998. Conflict Resolution for Air Traffic Management: A Study in Multi-Agent Hybrid Systems, *IEEE Transactions on Automatic Control*, **43,** 509-521.

[12] **Tomlin C., Lygeros J. and Sastry S.**, 1998. Hybrid Aerodynamic Envelope Protection using Hybrid Control, *American Control Conference*, Philadelphia, USA, June 30-July 2.

[13] **Dever C., Mettler B., Feron E., Popovic J. and McConley M.**, 2006. Nonlinear Trajectory Generation for Autonomous Vehicles via Parameterized Maneuver Classes, *Journal of Guidance, Control, and Dynamics*, **29,** 289-302.

[14] **Schouwenaars T., How J. and Feron E.,** 2004. Receding horizon path planning with implicit safety guarantees, *American Control Conference*, Boston, USA, June 30-July 2.

[15] **Ghosh R. and Tomlin C.,** 2000. Nonlinear Inverse Dynamic Control for Mode-Based Flight, *AIAA Guidance, Navigation and Control Conference and Exhibit*, USA

[16] **Frazzoli E., Dahleh M. A. and Feron E.**, 2005. Maneuver Based Motion Planning for Nonlinear Systems with Symmetries, *IEEE Transactions on Robotics and Automation*, **21**, 1077-1091.

[17] **Oishi M. and Tomlin C.,** 1999. Switched Nonlinear Control of a VSTOL Aircraft, *38th IEEE Conference on Decision and Control,* Phoenix, USA, December 10-12.

[18] **Isidori A.,** 1995. *Nonlinear Control Systems,* Springer, New York, USA.

[19] **Sastry S,** 1999. *Nonlinear Systems,* Springer, New York, USA.

[20] **Utkin V. I.,** 1992. *Sliding Modes in Control and Optimization*, Springer, Berlin, Germany

[21] **Lane S. H. and Stengel R. F.**, 1988. Flight Control Design Using Nonlinear Inverse Dynamics, *Automatica*, **24**, 471-483.

[22] **Azam M. and Singh S. N.,** 1994. Invertibility and Trajectory Control for Nonlinear Maneuvers of Aircraft, *Journal of Guidance, Control and Dynamics,* **17,** 192-200.

[23] **Devasia S., Chen D. and Paden B.**,1996. Nonlinear Inversion Based Output Tracking, *IEEE Transactions on Automatic Control,* **41**, 930-942.

[24] **Hauser J., Sastry S. and Meyer G.,** 1992. Nonlinear Control Design for Slightly Non-Minimum Phase Systems: Application to V/STOL Aircraft, *Automatica*, **28,** 665-679.

[25] **Tomlin C., Lygeros J., Benvenuti L. and Sastry S.,** 1995. Output Tracking for a Non-Minimum Phase Dynamic CTOL Aircraft Model, *34th Inst. of Electrical and Electronics Engineers Conference on Decision and Control*, New Orleans, USA, December 13-15.

[26] **Shkolnikov I. A. and Shtessel Y. B.**, 2001. Aircraft Nonminimum Phase Control in Dynamic Sliding Manifolds, *Journal of Guidance, Control and Dynamics,* **24**, pp. 566-573.

[27] **Slotine J. J. E. and Li W.,** 1991. *Applied Nonlinear Control,* Prentice Hall, New Jersey, USA.

[28] **Bartolini G., Ferrera A. and Usai E,** 1997. Output Tracking Control of uncertain nonlinear second order systems, *Automatica,* **33,** 2203-2212.

[29] **Levant A., et al.** 2003. Higher order sliding modes, differentiation and output feedback control, *Journal of Control,* **26,** 924-942.

[30] **Koyuncu E., Ure N. K. and Inalhan G.,** 2008. A Probabilistic Algorithm for Mode Based Motion Planning of Agile Unmanned Air Vehicles in Complex Environments. *IFAC World Congress*, Seoul, South Korea, July 06-11.

[31] **Ure N. K., Inalhan G.,** 2009. Design of a Multi Modal Control Framework for Agile Maneuvering UCAVs, *IEEE Aerospace Conference*, Big Sky Montana, USA, March 5-12.

[32] **Ure, N. K., Inalhan, G.,** 2009. Feasible Agile Maneuver Identification and Generation Algorithms on Multi Modal Control Framework, *AIAA Conference on Guidance Navigation and Control*, Chicago, Illinois, August 21-24.

[33] **Koyuncu E., Ure, N. K., Inalhan, G.,** 2009. Integration of Path/Maneuver Planning in Complex Environments for Agile Maneuvering UCAVs, *Journal of Intelligent and Robotics Systems,* **57,** 143-173.

[34] **Ure, N.K., Inalhan, G.**, 2008. Design of Higher Order Sliding Mode Control Laws for Multi Modal Agile Maneuvering UCAVs, *2nd Int. Symposium on Systems and Controls in Aerospace* , Shenzen, China, December 10-12.

[35] **Szurovy G., Gouilan M.**, 1994. *Basic Aerobatics,* Mc-Graw Hill, New York, USA.

[36] **Szurovy G., Gouilan M.**, 1997. *Advanced Aerobatics,* Mc-Graw Hill, New York, USA.

[37] **Kershner W. K.**, 1987. *The Basic Aerobatics Manual,* Iowa State University Press, Iowa, USA.

[38] **Shaw R. L.**, 1985. *Fighter Combat: Tactics and Maneuvering,* Naval Institute Press, Annapolis, MD, USA.

[39] **Url-1** <https://www.cnatra.navy.mil//pubs/ppub t45 str.htm>, 01.05.2010.

[40] **Url-2** <http://memebrs.tripod.com/~F15EEagle/manu.html>, 01.05.2010.

[41] **Lygeros J.**, 2004. *Lectures Notes on Hybrid Systems*, (ETH), Zurich, Swiss.

[42] **Liberzon D., Morse A. S.,** 1999. Basic problems in stability and design of switched systems, *IEEE Control Systems Magazine,* **19**, 59-70.

[43] **Kato O., Sugiura I.,** 1985. An Interpretation of Airplane General Motion and Control as Inverse Problem, *Journal of Guidance, Control, and Dynamics,* **9,** 198-203.

[44] **Koo T. J., Pappas G. J., Sastry S.**, 2001. Multi Modal Control of Systems with Constraints, *40th IEEE Conference Decision and Control,* Orlando, Florida, USA.

[45] **Valasek J., Downing D. R.,** 1993. *An Investigation of Fighter Aircraft Agility NASA Technical Paper 588,* (NASA), California, USA.

[46] **Smith D., Valasek J.,** 2001, Agility Metric Robustness Using Linear Error Theory, *Journal of Guidance, Control, and Dynamics*, **24**, 340-352.

[47] **Liefer R. K.,** 1990. *Fighter Agility Metrics NASA Technical Paper Report No: AD-A22447*, (NASA), California, USA.

[48] **Murphy P. C. et. al.,** 1991. *Candidate Control Design Metrics for an Agile Fighter" NASA Technical Paper 4238,*(NASA), California, USA.

[49] **Nguyen L. T. et.al**., 1979. *Simulator Study of Stall/Post-Stall Characteristics of a Fighter Airplane With Relaxed Longitudinal Static Stability NASA Technical Paper 1538*, (NASA), California, USA.

[50] **Stevens B. L. and Lewis F. L.**, 2002. *Aircraft Simulation and Control,* John Wiley& Sons, New Jersey, USA.

CURRICULUM VITAE

Candidate's full name: Nazım Kemal Üre

Place and date of birth: İzmir – 28 July 1987

Permanent Address: Istanbul Technical University, Faculty of Aeronautics and Astronautics, Controls and Avionics Lab, Ayazağa, İstanbul

Universities and
Colleges attended: 2008 B.Sc. Istanbul Technical University Faculty of Aeronautics and Astronautics, Astronautical Engineering

2008 B.Sc. Istanbul Technical University Faculty of Aeronautics and Astronautics, Astronautical Engineering,

Publications:

- **Koyuncu E., Ure N. K. and Inalhan G.,** 2008. A Probabilistic Algorithm for Mode Based Motion Planning of Agile Unmanned Air Vehicles in Complex Environments. *IFAC World Congress*, Seoul, South Korea, July 06-11
- **Ure N. K., Inalhan G.,** 2009. Design of a Multi Modal Control Framework for Agile Maneuvering UCAVs, *IEEE Aerospace Conference*, Big Sky Montana, USA, March 5-12.
- **Ure, N. K., Inalhan, G.,** 2009. Feasible Agile Maneuver Identification and Generation Algorithms on Multi Modal Control Framework, *AIAA Conference on Guidance Navigation and Control*, Chicago, Illinois, August 21-24.
- **Koyuncu E., Ure, N. K., Inalhan, G.,** 2009. Integration of Path/Maneuver Planning in Complex Environments for Agile Maneuvering UCAVs, *Journal of Intelligent and Robotics Systems,* **57,** 143-173.
- **Ure, N.K., Inalhan, G.**, 2008. Design of Higher Order Sliding Mode Control Laws for Multi Modal Agile Maneuvering UCAVs, *2nd Int. Symposium on Systems and Controls in Aerospace* , Shenzen, China, December 10-12.