Application of Maneuver-Based Control in Variable Autonomy Unmanned Combat Aerial Vehicles

Alexander M. G. Walan

APPLICATION OF MANEUVER-BASED CONTROL IN VARIABLE AUTONOMY UNMANNED COMBAT AERIAL VEHICLES

THESIS

ALEXANDER M.G. WALAN, Captain, USAF

AFIT/GAE/ENY/03-09

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

AFIT/GAE/ENY/03-09

APPLICATION OF MANEUVER-BASED CONTROL IN VARIABLE AUTONOMY UNMANNED COMBAT AERIAL VEHICLES

THESIS

Presented to the Faculty

Department of Aeronautics and Astronautics

Graduate School of Engineering and Management

Air Force Institute of Technology

Air University

Air Education and Training Command

in Partial Fulfillment of the Requirements for the

Degree of Master of Science in Aeronautical Engineering

ALEXANDER M.G. WALAN, BAEM, MBA

Captain, USAF

March 2003

AFIT/GAE/ENY/03M-09

APPLICATION OF MANEUVER-BASED CONTROL IN VARIABLE AUTONOMY UNMANNED COMBAT AERIAL VEHICLES

ALEXANDER M.G. WALAN, BAEM, MBA

Captain, USAF

Approved:

Dr. Curtis H. Spenny (Committee Chairman)	10 March 03 Date
Dr. Bradley S. Liebst (Member)	10 Mar 03 Date
Lieutenant Colonel David Jacques (Member)	11 MAR 03 Date

Acknowledgements

I would like to express my deepest appreciation to my parents without whom I could never have dreamed to come this far. It was their sacrifice that allowed me to attend truly exceptional schools. My Father has always been a role model and the standard by which I measure myself. Yet, it was only thru my Mother's immense patience and dedication that I was able to discover that in fact I could pass the 3rd grade. I owe a debt of gratitude to my sisters and brother as well. My big sister, who guided me thru those difficult first weeks of high school. My little sister, who never fails to call just when I need to talk. My brother, whose service and dedication to our nation is a constant source of inspiration and pride. My youngest sister, whose outgoing enthusiasm and sense of humor never ceases to amaze me.

Most importantly, I would like to thank my girl friend. You truly give all this a deeper meaning and I count myself infinitely blessed to have you. You are proof and validation of my faith in a kind and loving God.

In human works, though labour'd on with pain,
A thousand movements scarce one purpose gain;
In God's, one single can its end produce;
Yet serves to second too some other use.
So man, who here seems principal alone,
Perhaps acts second to some sphere unknown,
Touches some wheel, or verges to some goal;
'Tis but a part we see, and not a whole [23]

ALEXANDER M.G. WALAN

Table of Contents

List of Figures

Figure | Page

List of Tables

List of Abbreviations

AFIT/GAE/ENY/03-09

Abstract

The rise in the capability and lethality of unmanned combat aerial vehicles (UCAVs) historically has been paralleled by an increase in the complexity in the command and control of these systems. This trend has continued with the command and control of the current fleet of unmanned aerial vehicles such as the Predator and Global Hawk. The control of these vehicles falls on the extremes on the manual vs autonomous spectrum. As the missions tasked to these vehicles increase in complexity and lethality, operators will increasingly require the ability to tailor the amount of control exercised over the vehicle.

Maneuver Based Control (MBC) offers the potential to give future UCAV operators the ability to vary the autonomy of the vehicle against the amount of control they exercise over UCAV systems. The objective of this research is to validate the concept of Maneuver Based Control (MBC). This is accomplished under the umbrella of a conceptual UCAV mission. Particular attention is paid to the ability of this control scheme to increase operator situational awareness while decreasing the overall operator workload and required piloting skill. In addition, the ability for MBC to ensure effective control integrity over the vehicle is examined; ensuring that what vehicle does in response to a user's input is not divorced from the flight characteristics of vehicle.

Utilizing an existing non-linear computer model for an F-16 aircraft, maneuvers representative of those performed in a real-world mission are computed and stored. These stored maneuvers are then used to illustrate the application of MBC to in-flight replanning and mission execution by way of a representative mission scenario. Particular attention is paid implementing MBC thru manual maneuver input and by modifying waypoints. Results indicate that MBC provides an effective method of variable control for future UCAVs.

APPLICATION OF MANEUVER-BASED CONTROL IN VARIABLE AUTONOMY UNMANNED COMBAT AERIAL VEHICLES

I. Introduction

1.1 General

Beginning with the 1991 Persian Gulf War, air power has assumed an increasing prominence in the projection of US military and political power. Technological advancement has finally lead to the fulfillment of Air Power's long held promise of pin point accuracy and world wide range. Air Power now stands as the weapon of first choice for US policy makers. Among the many tools either currently in the air power arsenal or in development are numerous Unmanned Aerial Vehicles (UAVs).

In 1996 an Air Force Scientific Advisory Board (SAB) study examined the current and future potential of UAVs; finding that the UAV should expand from its then current roles of target and surveillance platform; becoming a weapon platform capable of a full range of offensive and defensive missions [21]. This is in stark contrast to the complete lack of interest in UAVs that characterized the Air Force after the Vietnam War. The post-Gulf War embrace of the UAV is due to many factors including:

- A declining force structure that necessitates innovative thinking
- Technological advancements that have enabled more capable unmanned operations (GPS as an example)
- Potential for cost savings in an era of limited budgets.

- Increasingly effective enemy defensive capabilities making manned missions increasingly dangerous [21]

The same technological innovations that make the UAV such a powerful weapon also make integrating that weapon into the total force very difficult. As UAVs increase in complexity and capability it is increasingly important to develop efficient tools for the command, control, and coordination of these systems. Central to this task is deciding what decisions and tasks to allocate to the vehicle and which need to remain under operator control. Deciding the relative level of manual vs autonomous operation is critical to maximizing mission effectiveness and poses one of the greatest developmental hurdles. [10]

This research examines this issue of autonomy by implementing an Unmanned Combat Air Vehicle (UCAV) control architecture based on pre-computed maneuver profiles; assessing its potential to allow for variable autonomy while increasing overall mission effectiveness.

1.2 Background

For the purposes of this research, a UAV will refer to an "air vehicle specifically designed to operate without an onboard operator or aircraft intended to be manned that have been converted to unmanned operation" (Definition used in 1996 AF SAB Report [21]).

Furthermore, UCAV refers to a UAV whose primary mission is to engage the enemy in combat operations. A system such as the Predator reconnaissance UAV which has been modified with a secondary capability to launch a weapon is not considered to be a UCAV. Finally, both UAV and UCAV is an aircraft designed for use multiple times; thus, cruise and other autonomous missiles are not considered UAVs.

1.2.1 UAV's: Historical Perspective. The first attempts at building a powered pilotless aircraft took place during World War I. The Germans were the first to experiment with a rudimentary UCAV.

> Surplus Eindecker (monoplanes) were used experimentally by the Germans. Loaded with explosives and controls fixed, they were launched using a guide rail - aimed at the enemy position up to fifty miles away. With a timer connected to the ignition, this pioneer UAV was then supposed to fall on the target after the calculated distance was flown. The experiments were inconclusive, with several of these UAVs crashing a few miles from launch or flying off into the distance to be blown off course or even turn back towards the launch site. The Germans dropped the idea in favor of manned aircraft. [5]

After the end of World War I, the US Army Air Corps experimented with the 'Bug'. This small aircraft was designed to carry a 100lb payload to a range of about 100 miles and used a pendulum based stabilization system. [5] However, as with the earlier German experiments, the technology of the day was not up to the task of making a useable and effective UCAV.

Figure 1.1 B-17 UCAV [13]

It was not until World War II that the first true UAVs were operationally employed. The first US example was the Ryan Radioplane, a target drone flown remotely via a three channel radio controller which controlled the rudder, elevator, and throttle. [5] The US experimented with UCAVs during this period as well, modifying a B-17 (loaded with explosives) to fly via a radio remote control. However, while target drones achieved some success, navigation and control technological limits doomed the B-17 UCAV project.

During the cold war, UAV development focused on reconnaissance. The BQM-34 was developed during the 1950's as the first UAV designed specifically for reconnaissance missions. Throughout the 1960's and 1970's numerous other reconnaissance UAVs were developed. In addition, the Air Force finally had developed an operationally suitable UCAV capable of delivering weapons and then returning for reuse:

> By 1971 the USAF had the first workable UCAV in the BQM-34A Firebee; a drone capable of releasing a pair of MK-82 (500lb Class) Bombs [8]

In spite of the contributions made by UAVs during the cold war and Vietnam, the massive drawdown following the Vietnam war spelled the end of US UAV and UCAV development; "including the elimination of Air Force UAV organizations in 1976" [21] However, Israeli success using UAVs during the 1980's rekindled interest within the US and this interest was heighten by the Gulf war in 1991.

1.2.2 Current UAV Developments. The Air Force currently has two major UAVs in service. The Predator medium altitude reconnaissance UAV, and the Global Hawk High Altitude Long Endurance (HALE) UAV. While the Global Hawk is still in the testing stages, Predator is fully operational and has seen combat service in both Operation Allied Force (1999), and Operation Enduring Freedom (2001-Current); these two aircraft are radically different in both design and operation.

Figure 1.2 Vietnam War Era C-130 Carrying Four Mk1 Firebees [5]

The newest Air Force UAV program is the X-45, a developmental effort between the Air Force, the Defense Advanced Research Project Agency (DARPA), and Boeing to develop a dedicated UCAV. The X-45 is designed to be a true combat vehicle and thus will face a more challenging mission and threat environment that either the Predator or Global Hawk [11].

Due to its unique mission requirements, the X-45 and other UCAVs need to be much more flexible than current unmanned systems. The need to avoid 'pop-up' threats, add last minute targets, and adapt to the unforseen are all capabilities that tomorrow's UCAVs will require. Such flexibility and demanding tasks contrast with the long and sometimes boring flight into and out of hostile airspace.

Figure 1.3 Boeing x-45 UCAV

Table 1.1 Current USAF UAV Programs

System	Mission	Status	Primary Operation Mode
Predator	Medium Altitude Recon	Operational	Manual Control
Global Hawk	High Altitude Recon	Testing	Automated
UCAV	Combat	Developmental	Variable Autonomy?

1.3 UAV Control

1.3.1 General Types of UAV Control. UAV control can be broken down into three general types: manual, semi-autonomous, and autonomous [13]. While the Predator is unmanned, its flight is not autonomous. Rather, the Predator and its sensors are manually controlled via a remote operation station throughout all phases of flight. A human operator, not the aircraft, determines the flight path and through the use of a set of remote aircraft controls, flies the aircraft flight [10]. In addition, the human operator must be extensively trained in basic piloting skills because as an AFRL study concluded “manned flying experience is necessary to employ the Predator effectively” [27]. While predator does have an autopilot, it is designed to operate in much the same manner as autopilots in manned aircraft. Thus, it is not designed to perform and is not suitable for complex combat maneuvers.

This level of automation is considered teleoperation or manual control; that is, the human operating the vehicle through remote means [25]. Manual operation is at the bottom left of the control vs monitoring scale as illustrated in Figure 1.4.

In contrast to the Predator, the Global Hawk is a ‘hands off’ system. The Global Hawk relies on extensive mission planning before each mission using the Air Force Mission Support System (AFMSS) [6]. Global Hawk takes the mission plan and autonomously executes the pre-programmed flight plan. Under this level of control the human operator is essentially just supervising the mission as the machine carries it out. This method of control, autonomous operation, is at the top right of the control vs monitoring diagram, Figure1.4. While manual intervention is possible in

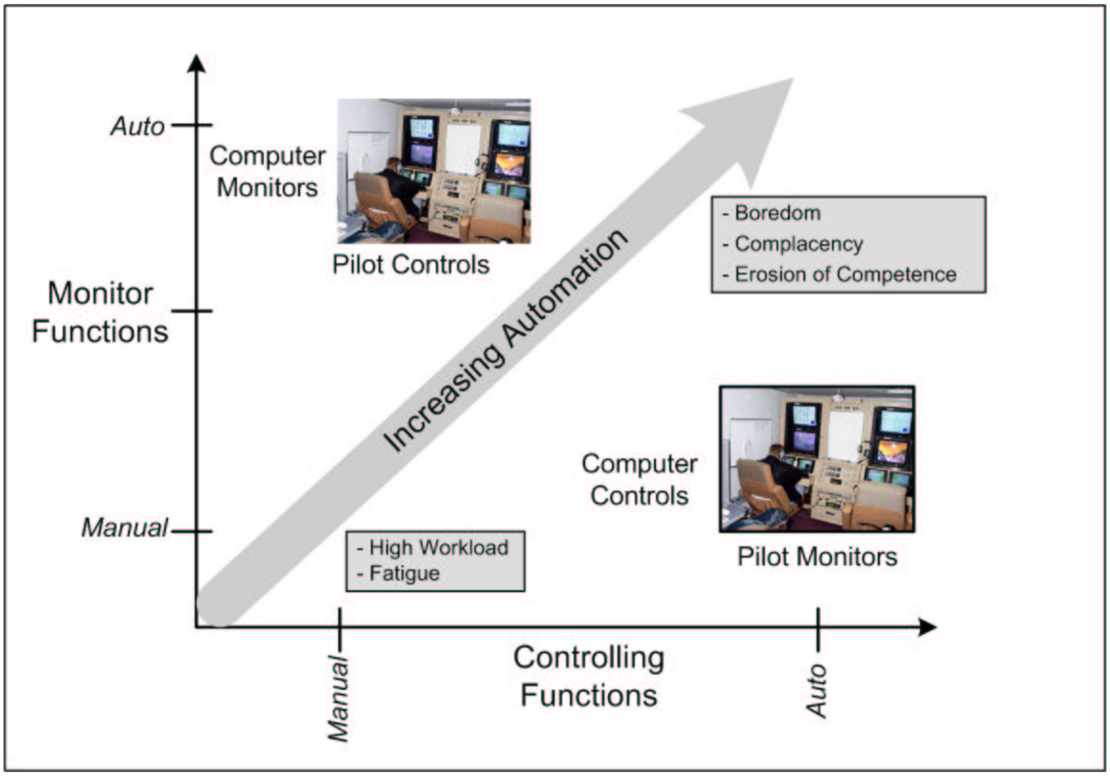

Figure 1.4 Pilot Continuous Control vs. Monitoring [25]

the Global Hawk system, the current system makes man-in-the-loop control slow and cumbersome.

Semi-autonomous control varies in degree and occupies the span of the middle ground between manual control and autonomous control. For a UAV this type of control implies that operator intervention is required for critical phases of flight, such as takeoff and landing, or during critical decision making but that the aircraft executes the rest of the flight autonomously.

1.3.2 Variable Autonomy UCAV Control. As 1.4 illustrates, both manual control and autonomous operations have serious drawbacks. Manual control inflicts a very high work load on the operator which over the course of the mission can degrade mission performance. Automated control, where the human is strictly in supervisory control, can lead to complacency and again decreased mission performance.

> There is a great deal of evidence that in supervisory control systems where most of the work is automated, the human operator typically does not perform well in maintaining vigilance (sustained attention) and making workload transitions from low workload to high workload. When alerts and exceptions require the human to make decisions and intervene after a period of low workload, he is likely to be slow to react and his decisions are likely to be sub-optimal.[1]

The Air Force's evolving Concept of Operations Concept of Operations (CONOPS) for the UCAV calls on the operator to be in control of multiple UCAVs; thus manual control would create too great a workload. In addition, manual control requires a pilot-type skilled operator. The very high level of training required for manual control is cost prohibited.

In contrast, the autonomous operation of a UCAV would require a less skilled operator but significantly more mission planning time and effort . In addition, there are serious legal implications for having an armed aircraft autonomously operating. For a UCAV "the fully autonomous mode presents the most problems legally due to a lack of human-in-the-loop... [manual or semi-autonomous] control pose little [legal] problems by maintaining a human-in-the-loop for authorization to release [weapons] [13]."

The need to maintain situational awareness, control multiple vehicles, and yet make control easy leads to the requirement for a truly variable autonomy UCAV. Variable autonomy is akin to the semi-autonomous concept describe earlier. Semi-autonomous control can be broadly broken down into two categories, sharing control and trading control [25]. "Sharing control means that the human and the computer control different aspects of the system at the same time . . . Trading control means that either the human or the computer turns over control to the other [25]."

Previous UCAV studies have shown the need for variable levels of autonomy to cater to both the varying levels of operator workload desired and changes circumstances during the mission [11]. Both sharing and trading control are applicable to UCAV operations. Table 1.2 illustrates one way to stratify levels of control over mechanical systems. Variable autonomy allows the operator to move between the levels of automation listed in Table 1.2 depending on mission requirements.

Table 1.2 Scale of Degrees of Automation [25]

Scale	Description
1	The computer offers no assistance, human must do it all
2	The computer offers a complete set of action alternatives and. . .
3	Narrows the selection down to a few
4	Suggests one, and
5	Executes that suggestion if the human approves
6	Allows the human a restricted time to veto before automatic execution
7	Executes automatically, then necessarily informs the human
8	Informs him after execution only if he asks
9	Informs him after execution only if the computer decides to.
10	The computer decides everything and acts autonomously, ignoring the human

1.4 Objectives

The objective of this research is to validate the concept of Maneuver Based Control (MBC)for a conceptual UCAV mission. Particular attention is paid to the ability of this control scheme to increase operation situational awareness while decreasing the overall operator workload and required piloting skill. In addition, the ability for MBC to ensure effective control integrity over the vehicle is examined; that is ensuring that what the vehicle does in response to a users input is not divorced from the flight characteristics of vehicle.

The MBC concept presented here is a further development of the work presented in Frazzoli [9]. While previous work focused on using pre-computed flight trajectories for mission planning and coordination purposes, this concept is expanded here to include UCAV in flight reactive control.

To accomplish this, the concept of in-flight replanning and mission execution will be introduced and examined. Building on this foundation, the this study will explore the utility of MBC to make in-flight mission changes.

1.4.1 Approach and Scope. Effective decision making is highly dependent on the accurate and effective presentation of information. Such information display is assumed and not the subject of this study. Rather, this study focuses on the theory and application of MBC as a means to achieve variable autonomy for a notional UCAV. Utilizing an existing non-linear computer model for an F-16 aircraft, maneuvers representative of those performed in a real-world mission will be computed and stored. These stored maneuvers are then used to illustrate the application of MBC to in-flight replanning and mission execution by way of a representative mission scenario. The user interface of the MBC system is not a focus of this effort.

II. Variable Autonomy Maneuver-Based Flight Control Theory

2.1 Overview

Before an in depth analysis of maneuver-based flight control can be undertaken, basic concepts related to aircraft flight and control need to be understood and common definitions established. First, some basic terms related to flight mechanics are presented, followed by current flight control and mission planning practices. Finally, the theory of maneuver-based flight control is established.

2.2 Aircraft Flight Dynamics

2.2.1 Frames of Reference. Three general frames of reference are used in the computation of aircraft states. The first is the body fixed axis which is attached to and moves with the aircraft. The second axis, the wind axis, serves as an intermediate step between the body, the free stream velocity, and the fixed inertial reference frame. The navigation reference frame is attached to the earth and provides the third reference frame. It is the navigation frame that is used as the fixed inertial reference frame of the total system.

The body axis is referenced relative to the frame of the aircraft. With the origin at the center of gravity, the x_b axis point directly out the nose of the aircraft. The y_b and z_b axis point orthogonally out the right wing and downward from the belly of the aircraft respectively. The body fixed axis, Figure 2.1, is used in the development and computation of the Equations of motions for the aircraft. The aerodynamic moments and angular rates the aircraft experiences are referenced from the body fixed axis.

The navigation axis, also known as the North-East-Down (NED) axis, is used as the inertial reference frame of the system. North is represented by the x axis, east by the y, and z is vertical downward toward the center of the earth. This axis allows the aircrafts position to be determined with reference to a point on the ground.

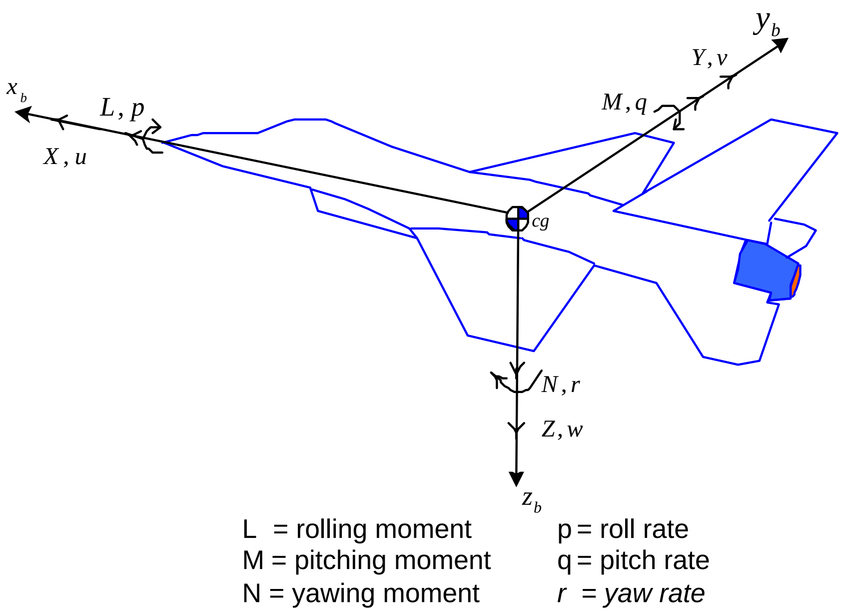

Figure 2.1 Body Fixed Axes

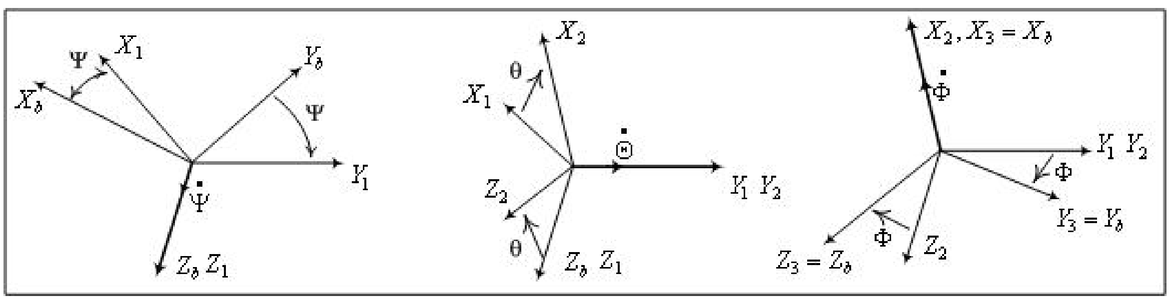

Figure 2.2 3-2-1 Euler Rotation

The NED axis will be used extensively later in this study to describe the position of the aircraft as well as its translation across the ground. The body axis and the navigation axis are related by the Euler Angles and three successive rotations, Ψ, Θ, and Φ, as shown in Figure 2.2.

The absolute velocities in the navigation axis can be found by utilizing matrix algebra and a rotation matrix comprised of the 3-2-1 Euler rotations in Figure 2.2. Equation 2.1 shows the general form of the absolute velocities where the rotation matrix R^{BN} is given by Equation 2.2.

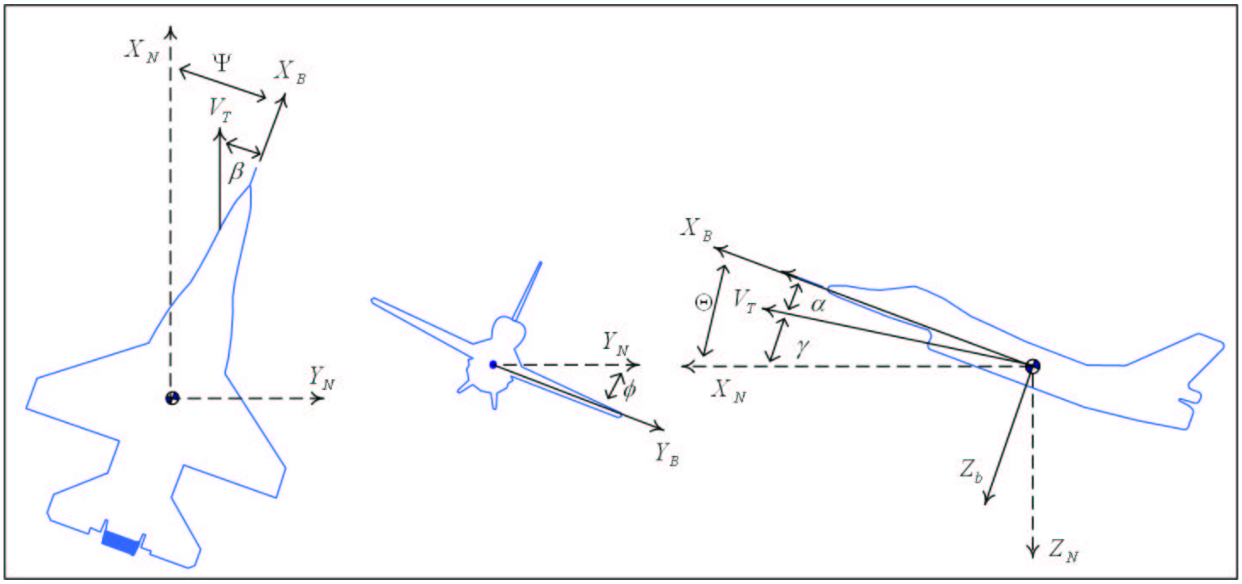

Figure 2.3 Euler Angles

$$\begin{bmatrix} \frac{dx}{dt} \\ \frac{dy}{dt} \\ \frac{dz}{dt} \end{bmatrix} = \begin{bmatrix} R^{BN} \end{bmatrix} \begin{bmatrix} u \\ v \\ w \end{bmatrix} \tag{2.1}$$

$$R^{BN} = \begin{bmatrix} \cos(\Theta)\cos(\Psi) & \sin(\Phi)\sin(\Theta)\cos(\Psi) - \cos(\Phi)\sin(\Psi) & \cos(\Phi)\sin(\Theta)\cos(\Psi) + \sin(\Phi)\sin(\Psi) \\ cos(\Theta)\sin(\Psi) & \sin(\Phi)\sin(\Theta)\cos(\Psi) + \cos(\Phi)\cos(\Psi) & \cos(\Phi)\sin(\Theta)\cos(\Psi) + \sin(\Phi)\sin(\Psi) \\ -\sin(\Theta) & \sin(\Phi)\cos(\Theta) & \cos(\Phi)\cos(\Theta) \end{bmatrix} \tag{2.2}$$

The third reference frame used is the wind axis [17]. The wind axis is used extensively in flight mechanics; both at the conceptual level with flight equations of motion and at the practical level through an aircrafts air data probe and other sensors. The aircraft's true air speed, V_t is referenced to the wind axis. The rotation matrix given by Equation 2.3 is used to transform the air speed in the wind axis to the three velocities in the body axis. These body-axis velocities are used in the numerical calculations of the aircraft states.

$$\begin{bmatrix} u \\ v \\ w \end{bmatrix} = \begin{bmatrix} R^{WB} \end{bmatrix} \begin{bmatrix} Vt \\ 0 \\ 0 \end{bmatrix} \tag{2.3}$$

Where

$$R^{WB} = \begin{bmatrix} \cos(\alpha)\cos(\beta) & \cos(\alpha)\sin(\beta) & -\sin(\alpha) \\ -\sin(\beta) & \cos(\beta) & 0 \\ sin(\alpha)\cos(\beta) & sin(\alpha)\sin(\beta) & cos(\alpha) \end{bmatrix} \tag{2.4}$$

2.2.2 Aircraft Forces. Utilizing the three reference frames described above, the forces exerted on the aircraft can be written and the aircraft states specified. A full description of aircraft forces and moments can be found in Reference [2] and is not presented here. However, because it forms the basis of all the aircraft maneuvers which will later be simulated, flight resulting in turning flight paths is of special interest.

Figure 2.5 illustrates the case of an aircraft in a level turn. Since the flight is at a constant altitude, summation of forces acting on the airplane leads to Equation 2.5. Where the load factor n is defined as $n \equiv \frac{Lift(L)}{Weight(mg)}$

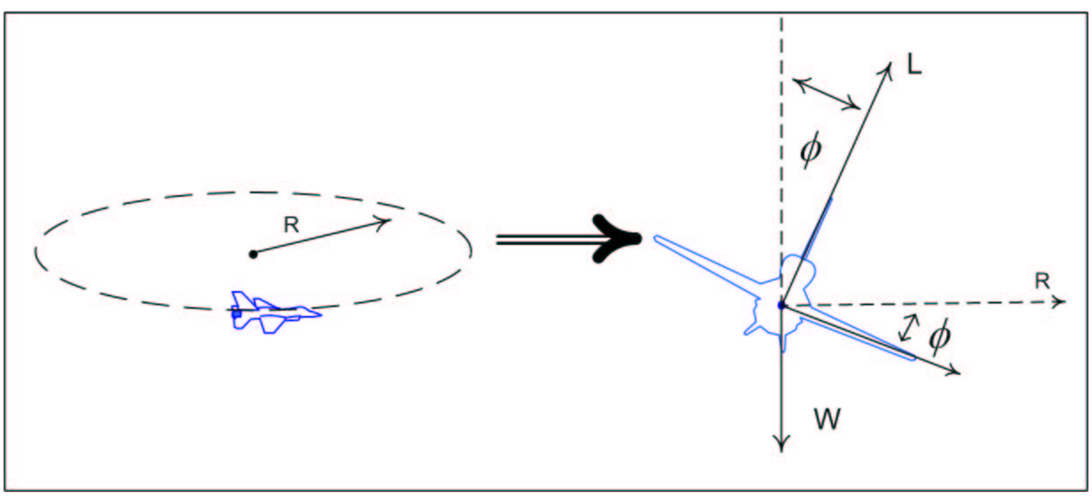

Figure 2.4 Aircraft in steady level turn

$$\Phi = \cos^{-1}(\frac{1}{n}) \tag{2.5}$$

Load factor, n, is most often simply refereed to as the "g's" that the airplane is "pulling." Aircraft maneuvers are often categorized based on the load factor involved. Using Equation 2.5 will later allow either the load factor or bank angle to be used as input into the UCAV non-linear dynamic model, since once one is determined the other can be calculated.

Utilizing the forces in Figure 2.5 as well as the load factor, it follows that the sustained turn radius of the aircraft is given by Equation 2.6. This relationship is useful in planning for situations where high maneuverability is required, such as threat avoidance, and will be used later to examined maneuvering under different mission scenarios and flight regimes.

$$R = \frac{V_t^2}{g\sqrt{n^2 - 1}} \tag{2.6}$$

The pullup, Figure 2.5 is another basic maneuver which involves curved flight path and of interest when considering basic maneuvers. Following the same procedure as above, Equation 2.7 results.

$$R = \frac{V_t^2}{g(n - 1)} \tag{2.7}$$

2.2.3 Ground Track. For operational air sorties, we are usually most interested in the actual path the aircraft travels over the ground. The ground track is the perspective that one sees while looking at a flight path displayed on a map. In addition, for the UCAV it is the threats and the targets on the ground that are of primary interest.

An accurate Inertial Navigation System (INS) or Global Positioning System (GPS) can easily provide the ground track history, but not the ground track for

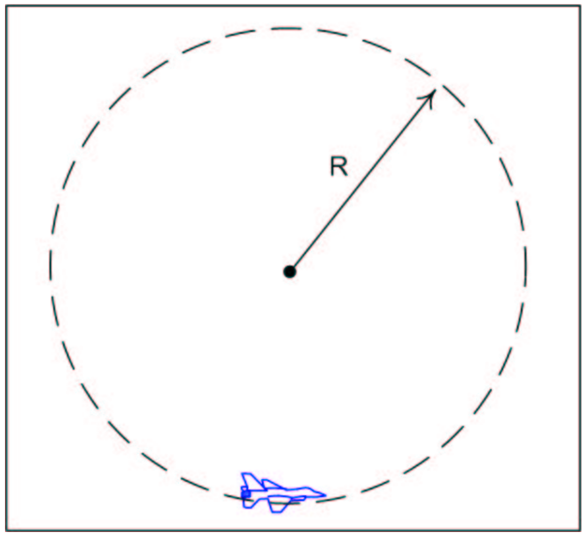

Figure 2.5 Aircraft in steady pull up

maneuvering flight before the fact. Thus, the ground track will have to be computed from the equations of motion.

Utilizing the navigation, NED axes, and taking into account initial positions we can trace the path the aircraft follows over ground. The North, East, Down vector is defined by the time history of the aircraft state vector, Equation 2.8. For discrete time modelling, the aircraft state vector is obtained by integrating the x,y,z displacements at each time step, Equation 2.9. By plotting the state vector consisting of the X,Y,Z displacements the path of the aircraft can be traced out. When plotting the ground track, only the X and Y vectors are needed.

$$\begin{bmatrix} North \\ East \\ Down \end{bmatrix} = \begin{bmatrix} \mathbf{X} & \mathbf{Y} & \mathbf{Z} \end{bmatrix}^T \tag{2.8}$$

Where

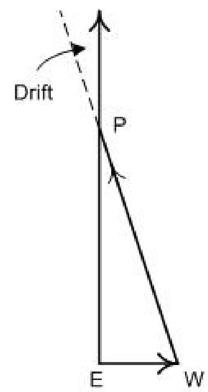

Figure 2.6 Wind Triangle

$$\begin{bmatrix} \mathbf{X} & \mathbf{Y} & \mathbf{Z} \end{bmatrix} = \begin{bmatrix} \int \frac{dx_1}{dt} & \int \frac{dy_1}{dt} & \int \frac{dz_1}{dt} \\ \int \frac{dx_2}{dt} & \int \frac{dy_2}{dt} & \int \frac{dz_2}{dt} \\ \vdots & \vdots & \vdots \\ \int \frac{dx_n}{dt} & \int \frac{dy_n}{dt} & \int \frac{dz_n}{dt} \end{bmatrix} \tag{2.9}$$

In a still atmosphere, the ground track will be the same as the track computed by in the navigation axis. However, if winds are present, these will create a difference in the indicated airspeed the aircraft sees and the actual ground speed achieved.

Standard convention for solving problems involving a non-zero wind speed is to use a vector diagram called a *Wind Triangle*, Figure 2.6 [28]. The wind is represented by the vector EW, the ground track speed is the line EP, while the heading vector is WP.

The six elements of the wind triangle are listed in Table 2.1. If any four of the six elements are known, the others can be found. In the case of the UCAV, the air speed, ground speed, heading, ground track are all known, due to the onboard instruments (INS, GPS, air data probe, etc).

Thus, Equation 2.10 can be used to find the remaining two unknowns. The angle between the wind vector and the ground speed vector is D, while the "wind correction angle" is represented by the angle WCA in Equation 2.10. The Wind correction angle is the angle by which the aircrafts heading must be modified to achieve the desired course in the navigation axis.

Table 2.1 Wind Triangle [28]

Vector	Direction	Magnitude (speed)
WP	Heading (Ψ)	Air Speed (V_t)
EP	Ground Track	Ground Speed
WE	Wind Direction	Wind Speed (V_w)

$$|G|^2 = |V_t|^2 + |V_w|^2 - 2|V_t| * |V_w| * \cos(180° - WCA - D) \tag{2.10}$$

2.3 Flight Operations

> "A prudent [operator's] job begins long before the journey begins. One of the [operator's] tasks is to choose a route and plan alternative courses of action" [29].

Flight operations, for the purposes of this study, are those tasks that are preformed in direct support of the aircraft's flight and mission execution. These operations can be broadly broken-down into two categories: the pre-flight planning and preparation and the in flight mission execution. Of those tasks necessary prior to take off (maintained, intelligence, training, ATO generation etc), only the mission planning portion is of interest in this study.

The mission planning process is closely tied with mission execution and controlling the aircraft in flight specifically. Thus, current practices and capabilities in mission planning, and their impact on the mission execution are discussed below.

2.3.1 Current Mission Planning Systems. Aircraft mission planning is the creation of a flight plan which takes into account terrain, weather, aircraft performance capability, configuration, as well as de-confliction with other aircraft [7]. The mission planner plans weapon delivery, fuel requirements; all while taking into account known enemy threat locations and type. Currently, the Air Force uses the Air Force Mission Support Systems (AFMSS) family of systems to perform these tasks. For UAV's as well as low observable (i.e. stealthy) aircraft, the mission planning aspect of flight operations is especially important due to the difficulty of making in-flight changes that don't adversely affect the survivability of the mission.

Current mission planning systems use kinematic representations of the aircraft to calculate a/c parameters such as fuel and time of flight between waypoints. However, as Frazzoli notes, this may not always lead to achievable aircraft maneuvers:

> . . . it is often assumed that a kinematic description of the vehicle's behavior is sufficient to represent its trajectories; typically, paths are computed as the interconnection of polynomials, or splines. However, such paths are not necessarily executable by the vehicles; rather, they are defined *a priori* , independent of the vehicles dynamics. [9]

Thus, current mission planning systems use large safety margins to insure that achievable routes and mission profiles are created.

AFMSS is the most capable of the mission planning systems used today. The AFMSS system is a set of computer and software tools that perform aircraft and weapon mission planning. Typically, the AFMSS core software is used in conjunction with aircraft specific Aircraft/Weapon/Electronic (AWE) software. These AWE modules provide aircraft performance data that the AFMSS core and other systems use to plan and display aircraft routes.

Once the mission is generated and saved, mission data is transferred to the aircraft via various data transfer devices, ranging from removable hard drives to compact flash cards. In addition, a hard copy of the entire mission is usually produced, the combat mission folder. A combat mission folder include imagery, detailed flight information, other aircraft missions, frequency allocation for communications, and detailed maps.

To create the mission folder and other materials described above, the Air Force and the Navy use a variety of mission planning products, including:

- CLOAR: The Common Low Observable Autorouter automatically plans and de-conflicts multi-aircraft routes that minimize exposure to known threat systems.
- PFPS: The Portable Flight Planning System is a PC based flight planning system designed for ease of use in application to aircraft systems that require low to moderate levels of mission planning.
- JMPS: The Joint Mission Planning System is a developmental mission planning system designed to provide multi-service commonality and AFMSS capability in PC based system.

The map display for a typical PFPS planned mission is shown in Figure 2.7. A majority of the mission time consists of straight ahead flight, including climbs and descents, and is represented by straight lines on the map. Of more interest here, are the waypoints and the various maneuvers they represent.

2.3.2 Way Point Navigation. The flight path shown in Figure 2.7 is an example of waypoint navigation. In waypoint navigation, also referred to as "en-route" navigation, "course changes are determined from the error in the aircrafts position and a selected waypoint" [10]. The waypoint coordinates are at a minimum referenced to some 2 dimensional location on the earth's surface, usually Latitude and Longitude (x,y). However, waypoints may be expanded to three dimensions, lat, long, and altitude (x,y,z) or even four dimensional with the inclusions of time.

In addition to coordinates, each waypoint may have specific mission task associated with it. A course change ($\triangle(\Psi)$), altitude, velocity, or other mission data may all be defined by waypoints. In Figure 2.7 the circles represent heading changes, the oval an orbit location, the square and triangle are the initial point and a target

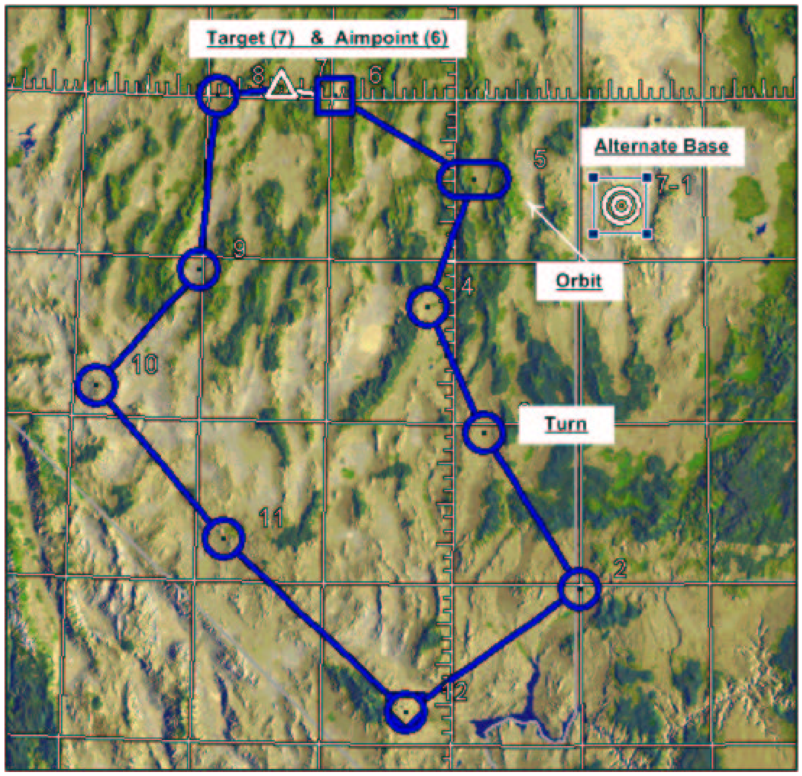

Figure 2.7 Examples of PFPS Route

respectively. By combining waypoints and the information associated with them, a mission profile or plan is created. The complete set of waypoints describe in detail the desired track and behavior of the aircraft.

Waypoint navigation spans the automation spectrum described earlier; fully manual to fully automated. For a manually controlled system, like the predator UAV, the human in the loop determines the aircrafts flight profile between the pre-determined desired waypoints.

2.3.3 In-Flight Mission Changes. High-end mission planners such as CLOAR and JMPS are designed to optimize mission routes. Thus, they use numerical optimization techniques to find local or global extremes for various cost functions. While the output of these programs greatly increases mission effectiveness, they do have drawbacks. Numerical optimizations techniques are computationally intensive and require high end processors and significant time, hours are normal. In addition, a solution is not always found. For these reasons, these systems are generally not suitable for in-flight replanning where short suspense times are required.

While current mission planners are not suitable for short suspense replanning, the control systems for current UAV's vary widely in their responsiveness. For highly manual systems such as Predator, the operator can easily use their manual controls to change the aircrafts flight as mission needs dictate. However, for a highly automated system such as Global Hawk changing the aircrafts flight plan can be a cumbersome process requiring extensive mission-replanning using the mission planning process and systems previously described.

Currently, in-flight replanning is limited for highly automated systems. For manual systems much effort and skill are required throughout entire flight, including adapting to new mission threats or requirements. Just what type of replanning capability is required and what in-flight mission changes need to be made are highly dependent on the specific circumstance. This applies to mission oriented events and environmental events: Threat pop-up vs loss of onboard system or sudden wind gust. Table 2.2 gives examples of the type of events that may dictate an in-flight mission changes and possible methods to make those changes.

Table 2.2 In-Flight Mission Changes

Time Available	Mission Scenario Example	Course of Action
Hours	New Fixed Target Added	User intervention required, replan using existing systems
Minutes	"Pop up" Threat Detected	User decision needed, possible automated execution
Seconds	Missile Launch Detected	Automated execution of pre-programmed manuever

2.4 Maneuver-Based Operator Control

The Maneuver-Based Flight Control concept presented here is a further development of the work presented in Frazzoli [9]. While previous work focused on using pre-computed flight trajectories for mission planning and coordination purposes, this concept is expanded here to include in-flight control for a conceptual UCAV. This flight control concept is radically different from standard waypoint navigation. Rather than defining a trajectory with waypoints and letting the aircrafts flight control system try and achieve it; Maneuver-Based Flight Control defines achievable trajectories in advance, creates maneuvers by splicing achievable trajectories together, then allows the operator to implement a desired maneuver to control the aircraft.

For this study, a library is developed which accurately describes a large class of feasible trajectories for the UCAV system. To create this library, numerical calculations are performed using a previously developed Matlab model of an F-16 aircraft and a Simulink-based control system. These serve as the computational model of the UCAV. Utilizing this library, a set of representative UCAV maneuvers will be computed and the value of the Maneuver-Based infight Control examined.

Key assumptions for this approach include:

- Vehicle dynamics are time in-variant
- Aircraft non-linear dynamics can be accurately modelled via numeric methods (Using Matlab)
- Complicated aircraft maneuvers can be created by piecing simpler maneuvers together

The assumption of time in-variance is the underlying assumption that allows the maneuver library to be constructed and stored *a priori*. However, this assumption is easily verified. In addition, the accurate modelling of aircraft non-linear dynamics, specifically the model used here, has been verified as well [14]. Note, time in-variance is only applicable for the same or similar aircraft configurations. Aircraft dynamics may change as fuel is burned or ordnance is dropped.

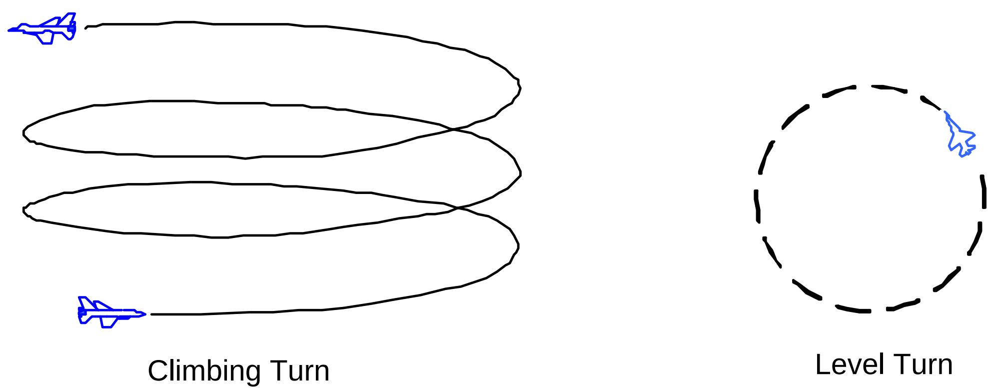

Figure 2.8 Examples of Steady State Trim Turns

Each entry in the maneuver library contains information about the UCAV's current state, changes to that state over time, and final state. Each maneuver begins and ends at the wings level steady state condition.

2.4.1 Steady State Trimmed Trajectories. The first set of maneuvers that are developed in the maneuver library are steady state trim trajectories. As Frazzoli explains:

> "steady state trajectories of the system, in which the velocities in body axes (i.e. as perceived by the [aircraft]) and the control input are constant. . . In the case of aircraft, relative equilibria are segments of helices, with a vertical axis; this includes degenerate helices such as straight lines, and horizontal turns" [9]

Some examples of steady state trim trajectories include:

- Steady Level Flight
- Constant g Climb/Descent
- Constant g Level Turn
- Constant g Climb/Descent Turn

These trimmed trajectories are the building blocks of the basic UCAV maneuvers which will make up the maneuver library. During these trajectories, the velocity and control surface deflections are constant.

2.4.2 Basic Maneuvers. Utilizing the steady state trim trajectories described above, more complicated flight maneuvers can be built and stored in the maneuver library. This set of basic maneuvers can include simple heading changes ($\triangle(\Psi) \neq 0$), simple climbs and descents, loops, and other. More advanced maneuvers such as offsets, the split s, or "bracket maneuvers" are contained of multiple basic maneuver and trim trajectories strung together. These will be covered in the next section.

Basic maneuvers begin and end with a trimmed trajectory. The most basic of maneuvers can consist of simple transitions from one trimmed state to another. For example, Figure 2.9 illustrates a heading change that consists of a steady banked turn connected at the start and finish to steady level flight. To simplify the analysis, all maneuvers begin and end with wings level steady level flight. This is a realistic simplification since, we can define wings level, steady level flight as the nominal aircraft state during flight.

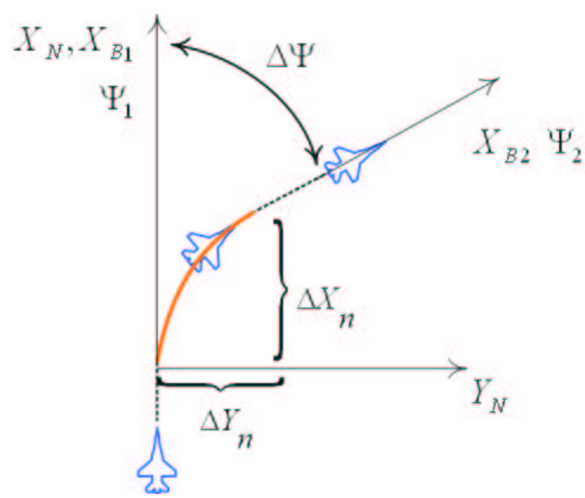

Figure 2.9 Basic Maneuver

Since each maneuver begins at wings-level steady level flight, we can use the navigation, NED, axis described earlier to track the aircrafts change in position and altitude over the earth. Thus, for a given trimmed trajectory maneuver, we can define a $\triangle X$, $\triangle Y$ *and* $\triangle Z$. In addition, the change in heading angle in the NED frame is given by $\triangle\Psi$.

Utilizing a discrete time system, the change in position and heading, as well as the other aircraft states are indexed. For purposes of calculation, each trimmed trajectory states at time zero and lasts a finite period. When constructing the basic maneuvers, the total state vectors can simply be added together to give a complete picture of the aircrafts behavior during the maneuver.

For most cases, the ground track is of greatest interest to the UCAV operator. To find the ground track produced by a maneuver consisting of two trim trajectories, Equation 2.11 is used.

$$\begin{bmatrix} North \\ East \\ Down \end{bmatrix} = \begin{bmatrix} X_1 \\ Y_1 \\ Z_1 \end{bmatrix} + \begin{bmatrix} R^1 \end{bmatrix} \begin{bmatrix} X_2 \\ Y_2 \\ Z_2 \end{bmatrix} \tag{2.11}$$

Where the rotation matrix R^1 is required since all x,y,z displacements for each trajectory vector is assumed to start at zero, with zero initial heading. The rotation matrix translates the second set of displacements into the frame of reference defined by the last x,y,z, Ψ entry of the initial trajectory.

$$R^1 = \begin{bmatrix} \cos(\Psi) & -\sin(\Psi) & 0 \\ \sin(\Psi) & \cos(\Psi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.12}$$

For this study, basic maneuvers are defined by transition thru one non-steady level flight trajectories, while advanced maneuvers may contain multiple different trajectories. For example, a simple heading change is a basic manurer, but multiple turns comprise a more advanced maneuver. Figure 2.10 illustrates this concept.

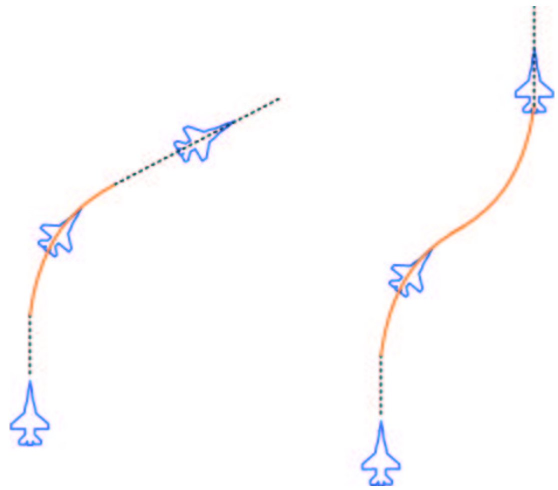

Figure 2.10 Example of Basic and Advanced Maneuver

2.4.3 Advanced Maneuvers. Figure 2.10 illustrates both a basic and advanced maneuver constructed by stringing together series of trimmed trajectories. For this study, only those advanced maneuvers that can be constructed from trimmed state trajectories will be examined. However, mission operators may desire to have available very mission specific maneuvers which involve non-trimmed states.

Such maneuvers may include pre-determined optimized flight paths for such things as minimum time to intercept or minimum time to climb. Optimization of such maneuvers is not the focus here; however, if specific maneuvers are required for a particular mission that require complex control inputs, those maneuvers could be constructed provided they are able to be accurately numerically evaluated [3]

2.4.4 Changing Flight Conditions. For each of the basic and advanced maneuvers described above, the flight regime where the maneuver takes place is constant. However, it is necessary to be able to transit between flight regimes. For example, adding new maneuvers to a pre-planned route may increase the total distance the aircraft has to fly to the target. Thus, in order to ensure the same Time on Target (TOT), the aircraft may need to increase its velocity.

Changes in velocity do not necessarily need to be modelled in the maneuver libraries. Rather, they can be treated and described as transition events. The current mission planning and execution system cane effectively model an increase in speed, calculating the resulting change in fuel consumption and travel time utilizing discrete point kinematic models. MBC then uses a pre-computed maneuver library which corresponds to the new flight regime.

III. Maneuver-Based Flight Control Matlab Simulation

3.1 UCAV Nonlinear Dynamic Model

Before one can effectively examine the concept of maneuver-based flight control it is necessary to accurately model the UCAV flight dynamics as well as its response to both user and external inputs. For this study, an existing and publicly available Matlab model of the US F-16 Fighting Falcon was used as the baseline flight model. While other models are available of different aircraft types, the F-16 is a close representation of the size and performance of the UCAV's likely to be fielded in the near future.

3.1.1 Simulink Based Flight Control System. The model *ucav.mdl* fig 3.1 is the nonlinear Simulink based model used to model the UCAV flight dynamics. This model takes bank angle and g-load inputs and simulates the resulting aircraft dynamics.

The Simulink controller is made up of several sub-controllers, see table 3.1 or reference [14] for additional detail. Each of the controllers listed in Table 3.1 contain control constants which need to be determined for each maneuver and flight regime of interest. These constants are then used as input into the controller along with the specific flight conditions and control inputs. The control constants in Table 3.1 need to be chosen carefully so as to produce the desired maneuver for a given input yet keep the flight of the vehicle controllable and not compromise stability.

Utilizing the variable step-time option in Simulink, the *ucav.mdl* model and the various sub-controllers, take an initial state vector and returns the final state vector for each time step. For simplicity and flexibility, these state vectors are passed to and from the model as Matlab M files.

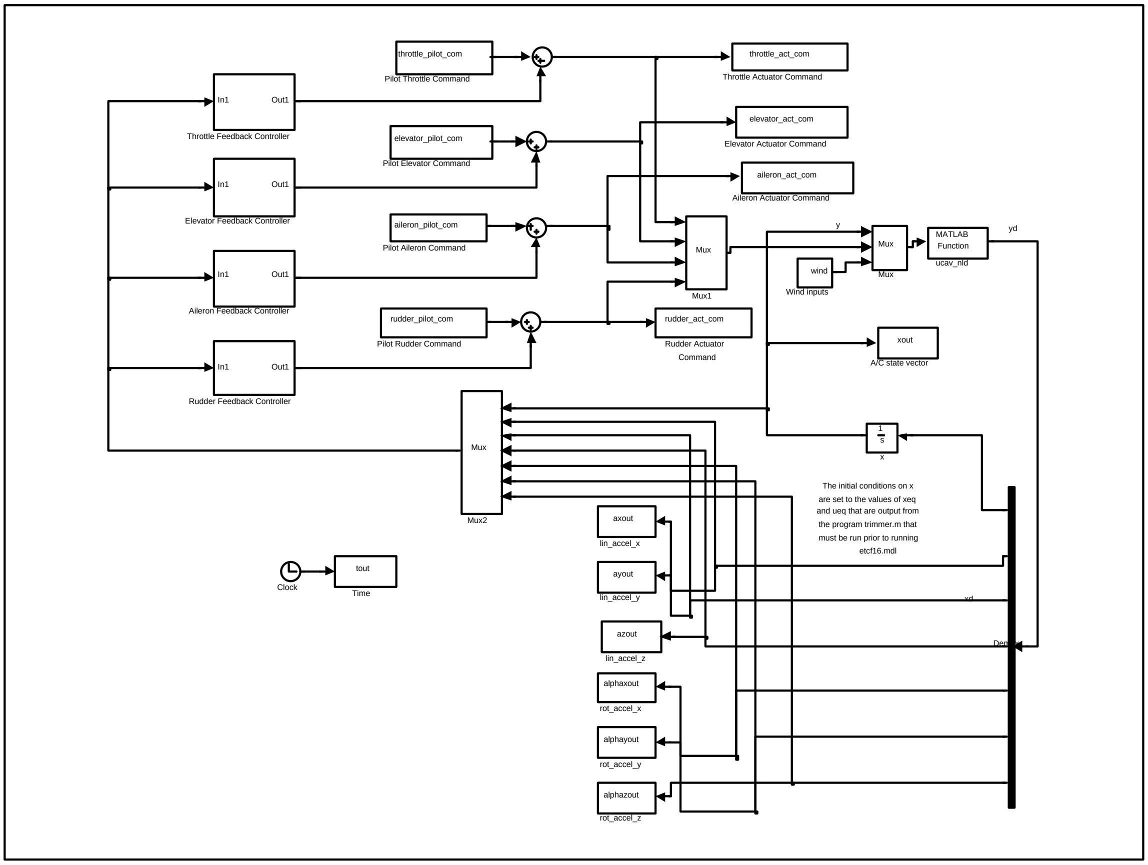

Figure 3.1 UCAV Simulink Model[14]

3.1.2 Matlab M Files. The heart of the *ucav.mld* Simulink model is a Matlab m file called *ucav.nld* This file is a modification of the original *subf16etc.m* file of reference [14]. *Subf16etc.m* is based on NASA aerodynamic data and is a program which calculates the state derivative vector for a baseline F-16 aircraft [14]. *ucav.nld* is essentially the same model and provides the backbone of the nonlinear dynamic simulation; however, wind direction and wind velocity states have been added. Thus, the *ucav.nld* state vector is two states longer than the original.

The *ucav.nld* m file requires an initial state and produces an output vector. The initial state vector is a 1x16 vector and is composed of the initial altitude, velocity, wind direction and speed, as well as the equilibrium trim constants. The equilibrium trim constants are obtained by running the *trimmer.m* file, see Appendix

Table 3.1 Simulink Controller

Controller	Components
Throttle Feedback Controller	Speed Hold Compensator
Aileron Feedback Controller	Bank Angle Hold Compensator
Elevator Controller	g command Hold Compensator Altitude Hold Compensator Pitch Axis SAS

A. *Trimmer.m* must be run for each unique flight condition and for any changes to the aircraft center of gravity (c.g.).[1]

The output vector is the aircraft state vector, each row corresponds to a specific time with each column corresponding to a specific aircraft state. See Appendix A for a complete description of these vectors.

3.2 Simulated Maneuvers

3.2.1 Generating Basic Trim Trajectories. Seven basic trim trajectories were modelled and these were later used to form the core of the maneuver library. These seven trajectories, see Table 3.2, were chosen because they can be used to describe a wide range of flight maneuvers which the UCAV can be expected to perform. The first of these trajectories is steady level flight, the most basic of trimmed flight conditions. Next, a steady climb and descent is modelled.

Four different turning trajectories were modelled which span the UCAV (F-16) flight envelope, Figure 3.2. The first is low g level turn. This type turn would be used in situations where a heading change is needed but time and distance are not major limiting factors. Next, two 3 g turns were modelled, one level and one climbing. These are more representative of situations where greater maneuverability is required.

[1]For all cases in this study a c.g. of .35 chord was used.

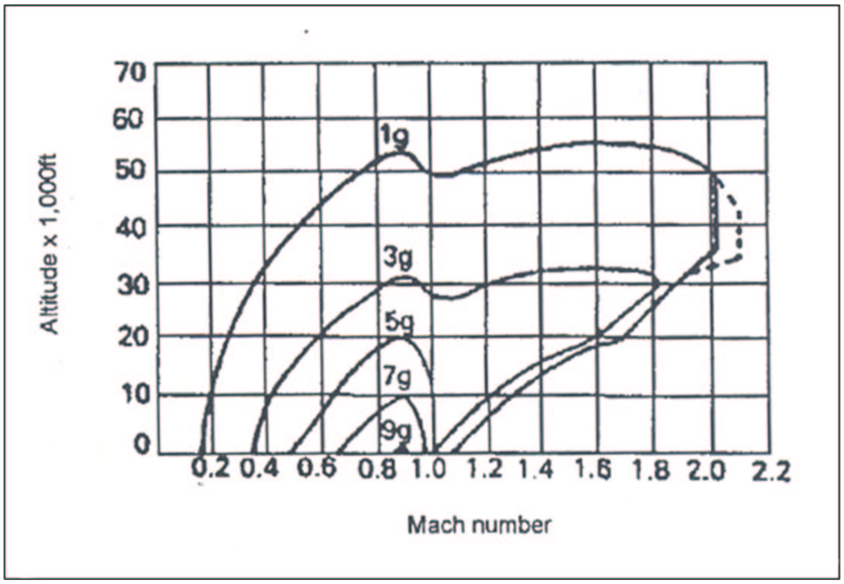

Figure 3.2 F-16/UCAV Sustained Turn Flight Envelope [20]

Finally, a 7g climbing turn was modelled. This is for those situations where high maneuverability is required, such as evading an imminent threat. When taken as a whole, the 7 trim trajectories can be used to describe both the nominal flight conditions as well as those trajectories necessary for combat situations.

Table 3.2 Trim Conditions Computed

Trim Index	Trim Condition	$\Phi InputCommand$	g Input Command
1	Straight and Level Flight	0	1
2	Steady Climb	0	1.2
3	Steady Descent	0	1.2
4	1.5g Level Turn	45 deg	1.5
5	3g Level Turn	70 deg	4
6	3g Climbing Turn	70 deg	4
7	7g Climbing Turn	80 deg	7.5

Before modelling each of the trim trajectories in Table 3.2 the bank angle and g command inputs first had to be determined. Utilizing Equation 2.5, the required bank angle command is easily determined for the required g load. However, it was found that by slightly increasing the g command input above the desired g load, a slightly faster response could be achieved by the UCAV controller. The climb and descent trajectories are performed with wings level, so no bank angle command is needed for these maneuvers but a g load command is still used.

Once the input commands were computed, the controller gains must to be determined for each trim trajectory. Reference [14] provided acceptable gains for the 1.5g level turn and 3g climbing turns trajectory indices 4 and 6 in Table 3.2. These gains formed the starting point for the gains for the other trajectories of Table 3.2.

For each trajectory the same general procedure was followed to determine the controller gains. For a given trajectory, an attempt was made to isolate the effect of the individual controllers and modify the gains thru an iterative process. As an example, the elevator controller contains the altitude, g-load, and pitch controller. After applying the appropriate input, running the system, and examining the resulting altitude, g, and theta outputs, the gains were modified until acceptable performance was achieved.

The optimization of the various controller gains is not a goal of this study; therefore, time was not spent trying to achieve exact tracking of g-loads or other parameters. Rather, only a minimum acceptable performance was required and the gains were modified until it each trajectory input resulted in a sustained and stable maneuver aircraft trajectory.

The final gains for the seven trim trajectories computed are listed in Table 3.3. Once these were determined, a Matlab file was created to create basic maneuvers for each of the 7 trim trajectories listed in Table 3.2.

3.2.2 Generating Basic Maneuvers. Maneuvers are generated by choosing a starting trim trajectory, entering a second trim trajectory, deciding how long to stay in the new trimmed trajectory, and finally deciding what the next trimmed state

Table 3.3 Controller Gain Constants

Gain Constants	Index 1	Index 2	Index 3	Index 4	Index 5	Index 6	Index 7
K_Θ	-3	-5	-5	-3	-7	-3	-3
K_{gi}	0	-5	-5	0	-4	-5	-7
K_{gp}	0	-.25	-.25	0	-.5	-1	-1
K_h	-1	0	0	-1	-4.1	0	0
K_v	1	1	1	1	2	1	2
K_q	-.3	-.3	-.3	-.3	-.3	-.3	-.3
K_Φ	-.5	-.5	-.5	-.5	-.5	-.5	-.5
K_α	-.5	-.5	-.5	-.5	-.5	-.5	-.5

should be. For all basic maneuvers calculated, the beginning and ending trimmed state was steady level flight, trajectory index 1.

The first step in creating the basic maneuvers is determining the flight regime in which the maneuver is to be performed. Nine different flight regimes, composed of three velocities and three altitudes, were chosen for each basic maneuver. These flight regimes, Table 3.4 are representative of flight regimes likely to be used on an operational mission; in addition, they offer good coverage of the UCAV fight envelope, Figure 3.2.

Table 3.4 Flight Regimes Used In Simulation

Velocity (ft/s)	Altitude (ft)	Mach Number
500	1000	.47
	10000	.49
	30000	.53
750	1000	.71
	10000	.73
	30000	.79
1275	1000	1.20
	10000	1.24
	30000	1.35

For each of the 9 flight regimes, three basic maneuver types were generated. These maneuvers were then used to construct a basic maneuver library. The three basic maneuvers and the trimmed trajectories which compose them are as follows:

Climb: Steady Level Flight - Climb - Steady Level Flight

Descent: Steady Level Flight - Descent - Steady Level Flight

Heading Change: Steady Level Flight - Turn - Steady Level Flight

The climb and descent maneuvers utilize the trimmed trajectory indices 1-3 from Table 3.2. The heading change maneuvers use the both trajectory index 1 as well as the low and high g turns in Table 3.2. The heading change maneuvers were computed in 15 degree increments from 0 to ±180 degrees of heading change, i.e. $\triangle(\Psi) = \pm 15°, \pm 30°, \pm 45°... \pm 180°$.

To compute a given heading change maneuver the following process was followed. First, for each flight regime, *trimmer.m* was run to obtain the equilibrium state vector. This vector along with the bank angle and g command are then loaded into the *ucav* Simulink model. The desired trim trajectory is then computed. Once a suitably large trim trajectory data set has been generated, this is stored and used to determine the time required to stay in the maneuver and when to enter the next trim state. The Matlab m files used to generate data can be found in Appendix B

For example, to model a 3g level turn resulting in a 60° heading change, the 3g level turn trim trajectory is computed and stored as an array. This array is the output of the *ucav* model and contains all the aircraft states as well as x,y,z displacements starting at time zero. The output array is searched to determine the time index when $\triangle(\Psi) = 60°$. This time is labelled t_1.

Time t_1 is close to the time where the roll out command, negative of bank angle, is applied. However, due to the inherent delays in the system, the control input's effect is not instantaneous. This can be seen in the delay between the commanded bank angle and the actual bank angle achieved by the system, illustrated in Figure 3.3.

Thus, in order to prevent the aircraft from overshooting the desired heading change $\triangle\Psi$, it is necessary to begin the roll out slightly prior to t_1. In this way, the

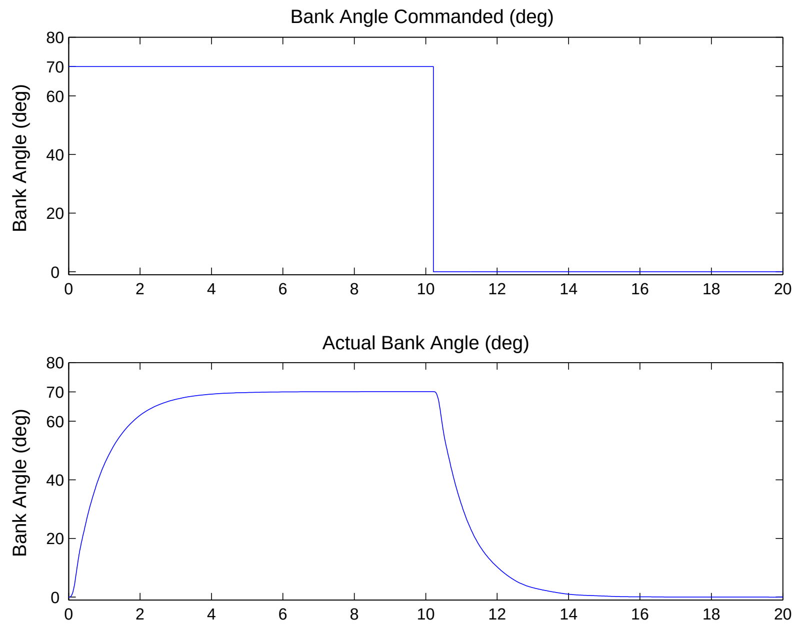

Figure 3.3 A/C Bank Angle Commands and Bank Angle Achieved: 60° Turn

aircraft will continue its turn while the wings are levelling. This offset, $\triangle t$ was found experimentally by running the simulation several times and iterating until acceptable performance was found. For most flight regimes and turn types $.2 < \triangle t < .5$

To perform the maneuver, the *ucav* model is reset to correspond to steady level flight. At $t_0 = 0$, the bank angle and g commands are input and the model is run. At the $t_1 - \triangle t$ the negative bank angle command is input. After allowing for any oscillation to die down, the aircraft resumes steady level flight. Thus, by modelling a roll into the turning trim state, holding that state, then rolling back out, a realistic ground track of a 3g turn resulting in a 60° heading change is established.

Figure 3.4 illustrates the results of the above maneuver. As desired, the altitude and velocity remain constant during the 70 degree banked 3 g turn. The heading angle Ψ is now 60° off the initial heading. The ground track of this basic maneuver is shown in Figure 3.5. The system delay is apparent on the ground track plot, as the aircraft travels nearly 1,200 feet forward before a noticeable heading change is observed.

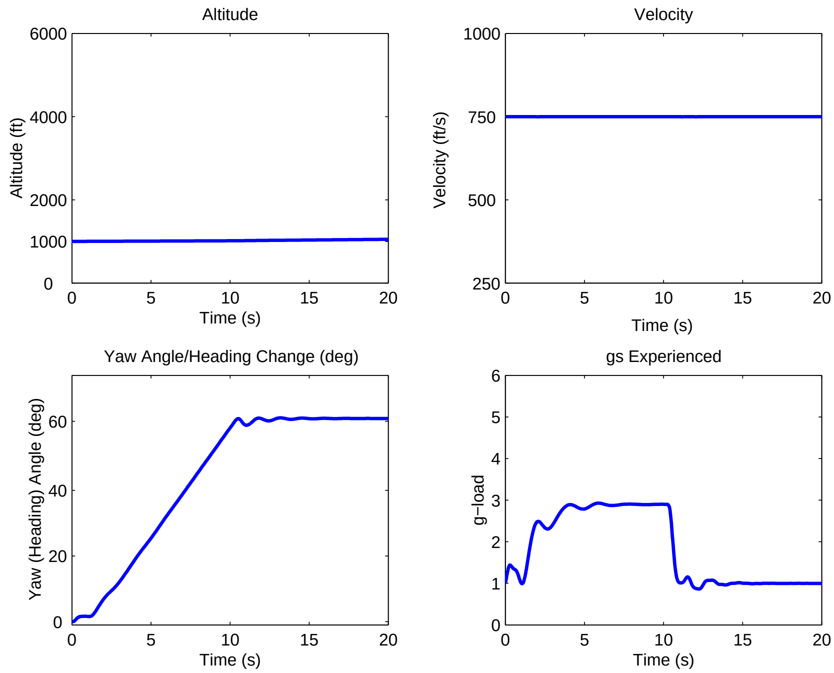

Figure 3.4 A/C States During 60 Degree 3g Turn

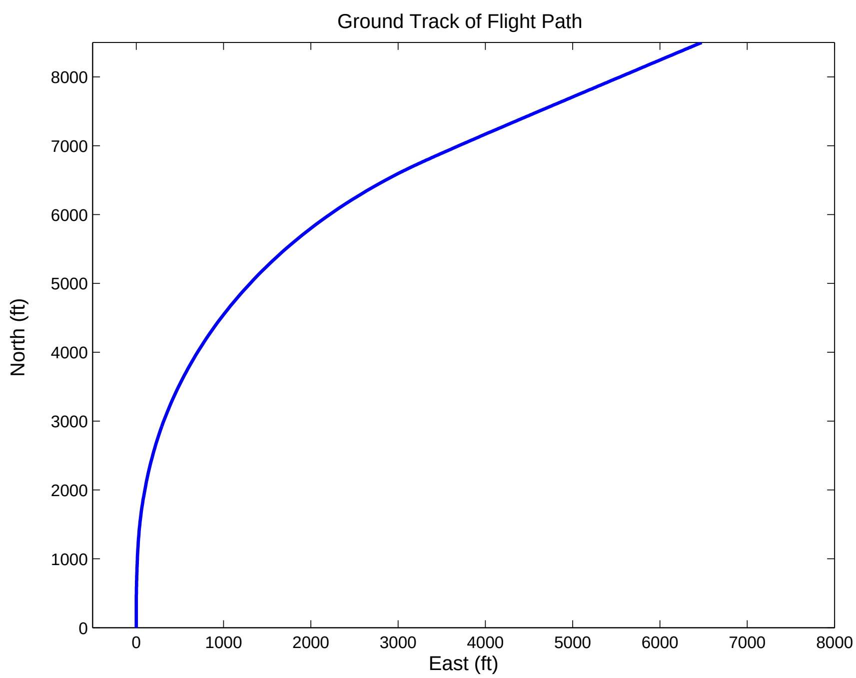

Figure 3.5 Ground Track For 60 Degree 3g Turn

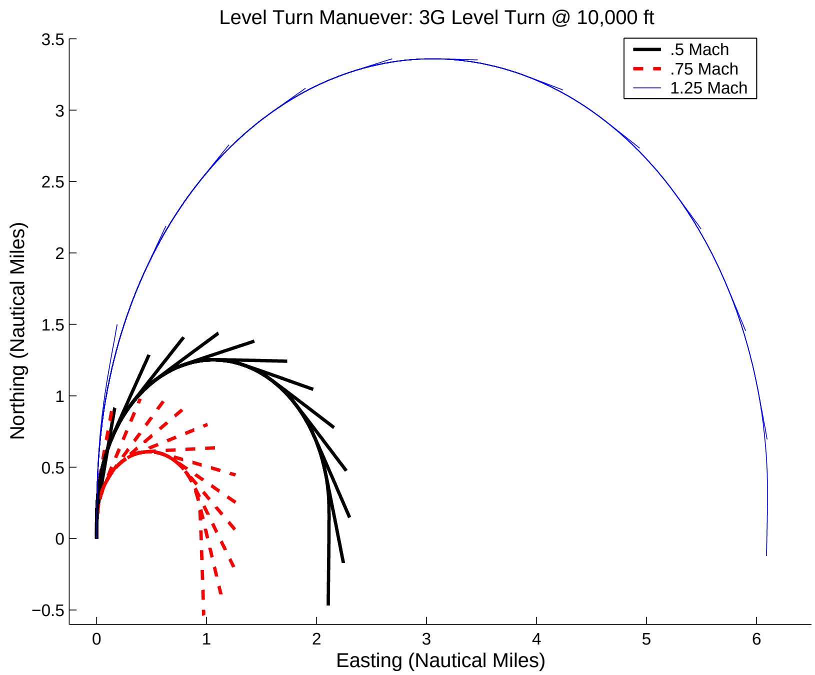

Figure 3.6 Ground Tracks as Velocity is Varied

The entire procedure described was performed for each of the turning trim trajectories in Table 3.2. The Matlab m files used to accomplish this can be found in Appendix B. For each turn type, 24 separate data sets were created, covering turns from 0 to $\pm 180°$ in increments of $\triangle(\Psi) = 15°$. Figure 3.6 illustrates a representative series of ground tracks for a specific turn type and altitude. 15° increments were chosen as a compromise between the need to provide operationally suitable maneuvers and yet the desire to reduce computations. For operational situations smaller turn increments could easily be developed and stored in the manuever library.

3.2.3 Basic Maneuver Library. When generating the basic maneuver, it is necessary to develop a library scheme which allows for the relevant data for each maneuver to be stored and then accessed when needed. For this study, a detailed library scheme was developed that specified maneuver type, heading change, flight regime, and wind conditions. This library scheme was used when constructing the

Matlab files that generated the basic maneuver data, so that successive files could be added to the maneuver library with relative ease.

For each of the basic maneuver calculated, the Matlab output was an array of up to 8 dimensions. Each of these dimensions, or pages, may contain sub-arrays that describe the maneuver. The configuration of this paged array is shown in Table 3.5. The first row of Table 3.5 lists the variables assigned to the array page as well as the range of each variable.

Table 3.5 Basic Maneuver Array: Matlab Output

Time Index	Aircraft States	Trim Index	$\triangle\Psi$ Index	Velocity (ft/s) Index	Alt (ft) Index	Wind Direction Index	Wind Velocity(ft/s) Index
1⟶m m varies	1⟶n n=5	1⟶mn[a] mn=7	1⟶p p=25	1⟶q q=3	1⟶r r=3	0⟶s s=12	0⟶u [b] u=1

[a]Note 'o' Skipped to avoid confusion with zero
[b]Note 't' Skipped to avoid confusion with time

The first page (first column in Table 3.5) of the array is the time index. It's length varies for each maneuver, due to the variable time step size used by Simulink and the different times that each maneuver took to complete. The second column of the array contains the aircraft states. All 22 outputs of the *ucav* model can be stored; however, for the basic maneuver library only 5 were used. These were the x(t),y(t),z(t),Ψ(t), and a vector containing the total changes over the course of the maneuver $(\triangle(x), \triangle(y), \triangle(z), \triangle(time))$.

The third and fourth pages of the arrays contain the trim index, Table 3.2, and the total heading change. Finally, the various flight conditions are contained in pages 5 thru 8. Table 3.6 contains the correlation information on what each of the values in each page mean.

Utilizing the information in Tables 3.5 and 3.6, each basic maneuver is categorized by a manuever index, this index is given by the [mn,p,q,r,s,u] values. For example, a maneuver index of [4,4,2,2,0,0] describes a 1.5g level at turn resulting in a 45 degree heading change while travelling at 750 ft/s and 10,000 ft with no wind

Table 3.6 Basic Maneuver Library Key

Index Type	Index Number	Definition
Maneuver Index	1	Steady Level Flight
	2	Steady Climb
	3	Steady Descent
	4	1.5g Level Turn
	5	3g Level Turn
	6	3g Climbing Turn
	7	7g Climbing Turn
$\triangle(\Psi)$ Index	1	$\triangle(\Psi) = 0$
	2	$\triangle(\Psi) = 15°$
	3	$\triangle(\Psi) = 30°$
	$\vdots$	$\vdots$
	13	$\triangle(\Psi) = 180°$
	14	$\triangle(\Psi) = -15°$
	15	$\triangle(\Psi) = -30°$
	$\vdots$	$\vdots$
	25	$\triangle(\Psi) = -180°$
Velocity Index	1	Velocity =500 ft/s
	2	Velocity =750 ft/s
	3	Velocity =1250 ft/s
Altitude Index	1	Altitude=1,000 ft
	2	Altitude=10,000 ft
	3	Altitude=30,000 ft
Wind Velocity Index	0	Wind Velocity =0
	1	Wind Velocity =40 ft/s
	2	Wind Velocity =75 ft/s
Wind Direction (Ω) Index	0	$\Omega = 0$
	1	$\Omega = 15°$
	2	$\Omega = 30°$
	$\vdots$	$\vdots$
	12	$\Omega = 180°$
	13	$\Omega = -15°$
	14	$\Omega = -30°$
	$\vdots$	$\vdots$
	24	$\Omega = -180°$

present. By storing all basic maneuver data in the library, more advanced maneuvers can quickly and easily be created by stringing basic maneuvers together.

3.2.4 Generating Advanced Maneuvers. As described in Chapter 2, advanced maneuvers consists of multiple basic maneuvers and trim trajectories pieced together to form one continuous maneuver. The first of these advanced maneuvers to be constructed was a simple off set maneuver. The off set consists of two equal and opposite bank turns performed back to back, ending in steady wings level flight.

The result of this maneuver is that the aircraft is following its initial heading, but its flight route is off set in the cross range direction. The amount off offset is determined by the specific flight regime, turn type, and the size (in degrees) of the turns. The off set maneuver would be used in a case where the operator desires the same heading, but wants to change the ground track in the cross range direction. A target that has shifted its location is one scenario where this maneuver would be used.

The offset maneuver is just one of many possible advanced maneuvers that can be constructed using the entries in the basic maneuver library. Figure 3.7 illustrates the offset maneuver as well as variations that can be constructed. A subset of the offset library, the 1.5 g level offset at 10,000 ft, is shown in Figure 3.8. The relationship between velocity and turn rate/radius is clearly illustrated by Figure 3.8.

The off set maneuver is constructed by piecing together two equal and opposite turns from the basic maneuver library. The first determines the turn type (g load, level vs climbing) the second turn is then the equal but opposite of the first. The second matrix is multiplied by the rotation matrix given by Equation 2.12 where $\Psi = \Psi_{final,turn1}$ and added to the end of the first turn.

The two simple variations of the offset maneuver shown in Figure 3.7 are constructed in the same manner as the offset; piecing together equal and opposite

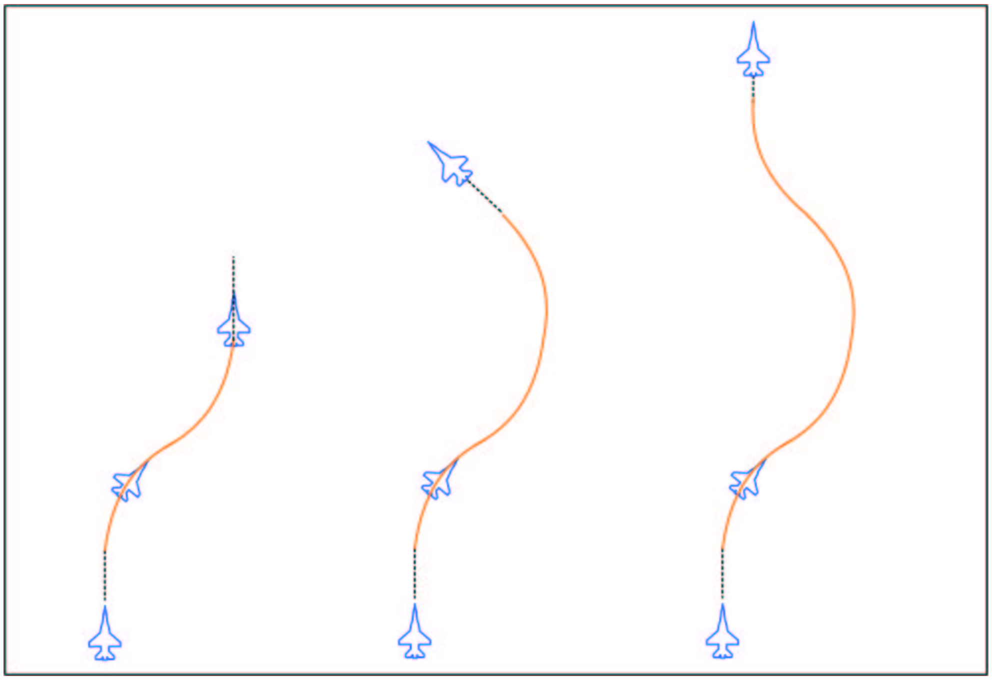

Figure 3.7   Offset Maneuver with Variations

turns from the basic maneuver library. The third maneuver in Figure 3.7 is of special importance, it will be used to illustrate MBC in chapter 4. For this reason, a full series of these, “go-around” maneuver were generated and added to the advanced maneuver library.

The procedure used to generate the go-around is much the same as with the offset maneuver. However, the go-around consists of two equal turns connected to two equal but opposite turn. For example, the aircraft rolls into a 30° turn, then executes two −30° turns back to back, finally ending with another 30° turn. The end result is that the aircraft makes a detour but then resumes its original flight path.

A subset of the go-around maneuver library is shown in Figure 3.9. Again, the dramatic affect of velocity on maneuverability is clearly shown. Each of the go-around's in Figure 3.9 was constructed by using equal magnitude turns with $\triangle(\Psi) = 15°$, ranging from $\triangle(\Psi) = 15°$ to $(\triangle\Psi) = 90°$.

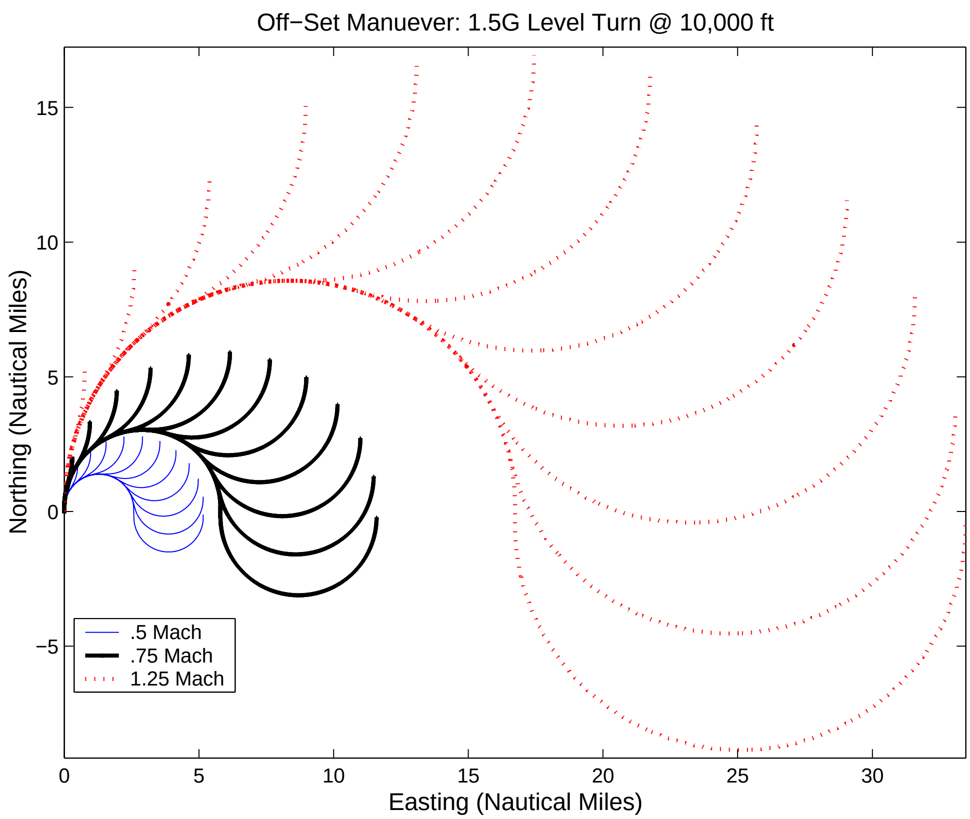

Figure 3.8 Calculated Offset Maneuver

The piecing together of the basic maneuvers to form the more advanced maneuvers of Figure 3.7 is possible because each trim trajectory and hence basic maneuver in the library has wings level flight as the nominal state. Thus, while translation of the aircraft is allowed thru the x,y,z directions, the other aircraft states are bounded on both ends of the maneuver.

Maneuvers which may be utilized frequently can be easily created and stored in an advanced maneuver library. The creation of this library is a greatly simplified compared to the creation of the basic maneuver library. For this study, a level flight offset maneuver library was created by systematically taking each turn type, at each flight regime, and pairing it with a turn of equal and opposite magnitude. Since each basic turn type has 12 entries, where each turn covers a heading change of $\triangle(\Psi) = 15°$.

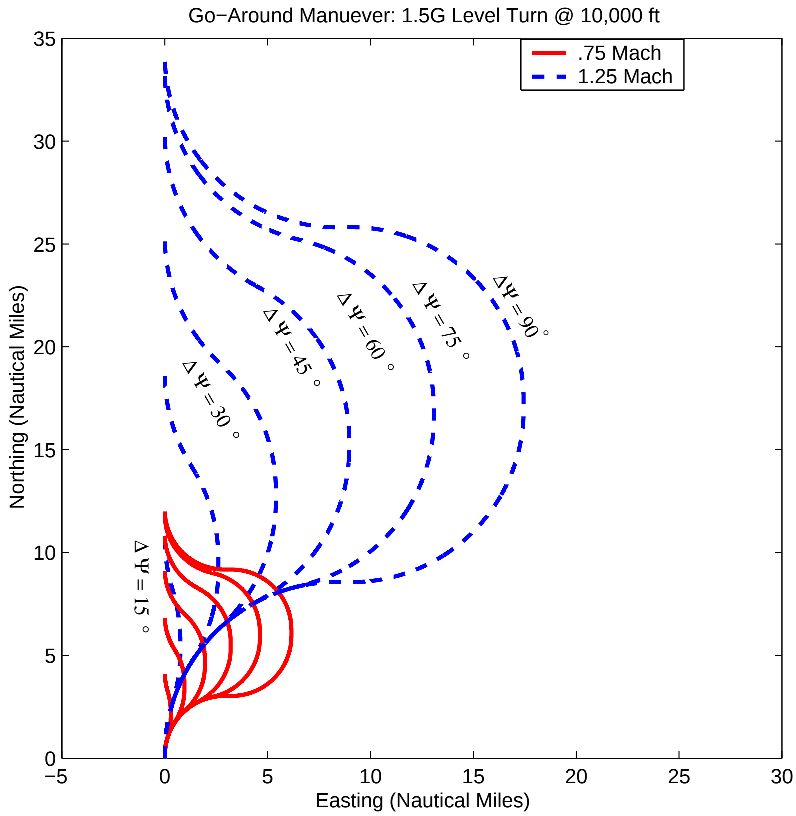

Figure 3.9 Calculated Go-Around Maneuver

Since this library used pre-computed basic maneuvers, the time needed to create the entire library was a fraction of the time to model even the simplest trim trajectory. The advanced maneuver offset library was created in seconds versus hours for the basic turn library. Thus, new advanced maneuvers could easily be created by the UCAV operator while the mission is in-flight and executed within minutes.

3.2.5 Accounting For Winds Aloft. As mentioned in chapter 2, for most instances the UCAV operator will be most interested in observing the ground track of the UCAV as displayed on a map. Thus, it is imperative that winds and their effect on ground track are taken into account. Since the aerodynamic calculations in the non-linear model of the ucav use the true velocity, V_t, moderate winds will not affect the dynamics of the aircraft thru the air for a constant velocity turn; however they will affect the ground track of the vehicle.

During the mission planning process, wind corrections are usually made using forecasted wind data. However, if the aircraft has a reliable INS/GPS, more precise wind data can be used determined and then applied to MBC while the aircraft is in flight. The aircraft air data probe provides the aircraft's indicated velocity V_t, and the INS/GPS can be used to provide the true airspeed. Applying equation 2.10, the wind speed and direction can then be calculated.

For those flight regimes where winds are present, a correction term must be made to the x,y displacements calculated to produce the ground track [2]. This can be easily done by modifying the Matlab m file *ucav.nld*. The x,y displacements are calculated by integrating the d_x and d_y terms. By adding a correction factor which accounts for the wind velocity (w_v) and direction (Ω) the effects of winds on ground velocity and hence ground track can be obtained. Equation 3.1 shows the correction factor where the wind angle is defined as in Figure 3.10.

$$
\begin{aligned}
d'_x &= d_x + w_v * (\cos(\Omega) * \cos(\Psi)) \\
d'_y &= d_y + w_v * (\sin(\Omega) * \sin(\Psi)) \\
d'_y &= d_y
\end{aligned}
\tag{3.1}
$$

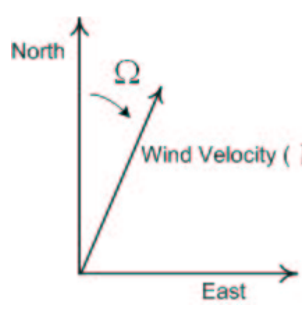

Figure 3.10 Wind Direction Definition

[2]Winds are assumed to have no vertical component for the model used here.

With the wind correction factors shown in Equation 3.1 added to the *ucav* model, it is a straightforward process to generate trimmed trajectories that account for the wind. A 25 kt (40 ft/s) wind speed was used and trajectories calculated for $\Omega = 0$ to $\Omega = 180$ in increments of $\triangle(\Omega) = 15°$. The wind speed was chosen by randomly picking site data from the National Oceanographic and Atmospheric Administration wind data website [19]. Appendix refapp:data contains a representative data set of this wind aloft data.

Figure 3.11 shows a subset of the trajectories calculated. As one would expect, the effect of wind grows as the maneuver progresses. From Figure 3.11 and the closer look provided by Figure 3.12 with this moderate wind level there is little difference between the wind corrected turns and the zero wind case (each turn within 700 ft of zero wind case). However, over the course of many maneuver or in cases of larger winds, these affects may become significant.

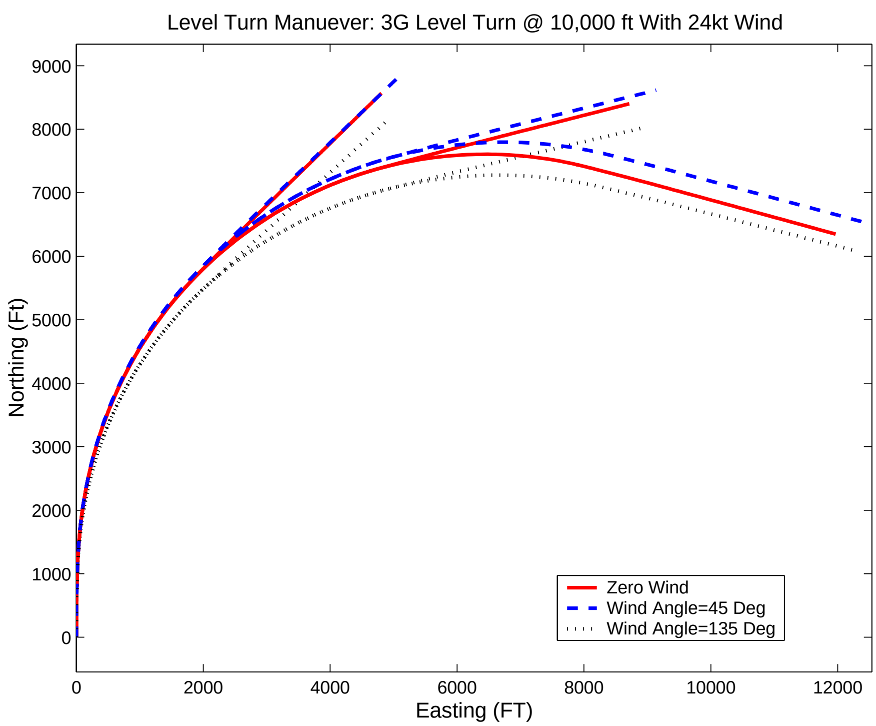

Figure 3.11 Turn Calculated With Winds Aloft

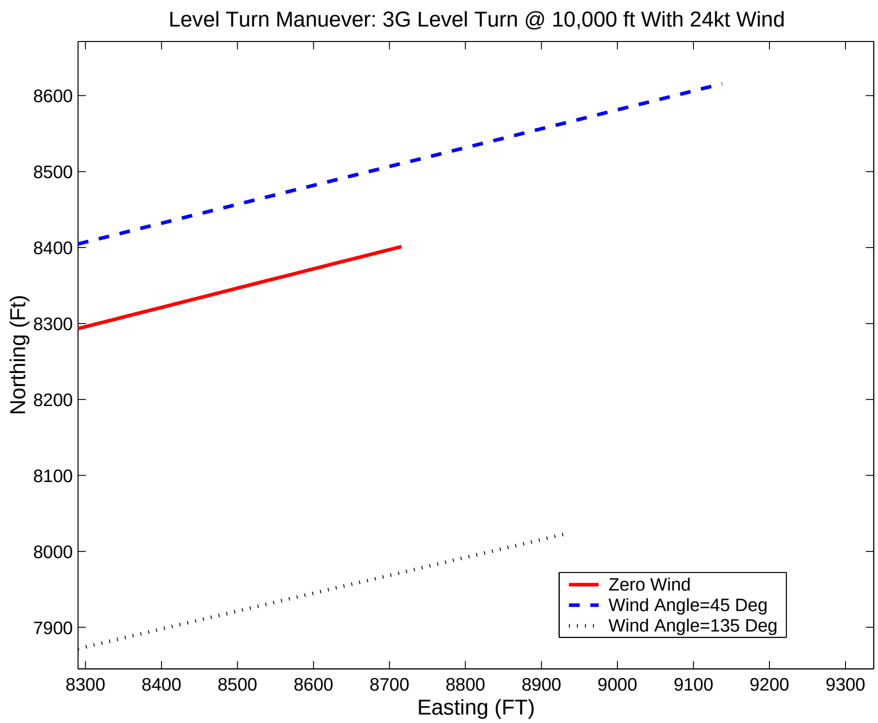

Figure 3.12 Close-Up of Turn Calculated With Winds Aloft

By computing the wind corrected maneuvers ahead of time, an accurate ground track of the aircraft's path can be computed and stored in the maneuver library. These wind corrected maneuvers can then be applied to make more accurate in-flight mission changes than the no wind case would provide.

IV. The Maneuver-Based In-Flight Control System

4.1 General

Maneuver-Based Control (MBC) is not intended as a replacement for the traditional mission planning systems or methods of control described earlier. Nor is it intended to be a comprehensive "take off to landing" approach to flight control. Rather, MBC is intended to be used in very specific circumstances where variable levels of operator input is required for the UCAV system. Thus, MBC augments the full mission planning and execution systems that a UCAV utilizes. In this section, those specific mission scenarios where MBC is applicable will be explored and different implementation schemes proposed.

4.2 Application of In-Flight Maneuver-Based Control System

While tomorrow's UCAV's will undoubtedly use new systems, the concepts and techniques for mission planning and execution are likely to be similar to today's systems. The MBC scheme presented here is designed to compliment the existing mission planning and mission control systems for a UCAV/UAV. MBC can therefore be thought as a sub-system of the UCAV's mission planning and execution systems.

4.2.1 Notional UCAV Control Architecture. MBC is primarily for short suspense in flight mission re-planning; it does not replace the onboard aircraft autopilot. Rather, by modelling the aircraft behavior before the fact, it accurately predicts the aircrafts behavior for a given maneuver. The aircrafts onboard systems remain in control of the actual flight, giving the control commands to move the control surfaces and change the throttle settings during flight.

For this study, the UCAV is assumed to be operating primarily in an automated mode, with the operator in supervisory control. This mode of control was chosen because it matches the CONOPS for UCAVs mapped out by the Air Force SAB.

By choosing a primary operation mode close to the top of the automation scale, Figure 1.4, the operator is "freed from boring tasks to accomplish those functions most suited to human intellect" [21].

Under this notional UCAV CONOPS, a single operator may be in control of multiple UCAVs during a singe mission. Thus, controlling each UCAV via a stick and throttle is not practical. In addition, a skilled operator is assumed to be operating the UCAV; however, the operator need not be a pilot but is assumed to have a detailed knowledge of flight and mission tactics.

Under this proposed MBC control system architecture, there are two primary users of the information in the MBC library, the human operator and the UCAV control system. The human operator uses MBC to make changes to the projected flight path of the vehicle. The input commands to make these changes having been already modelled and stored in the maneuver library are then transmitted to the vehicle. The UCAV control system then executes these commands, using its feedback control system to ensure proper tracking of the desired heading, altitude, and velocity.

4.2.2 Entering the Control Loop.

> No one can anticipate all events that may occur during flight. Malfunctions, retasking, enemy actions and countermeasures, intrusions by friendly forces, and other events may call for mission replanning or other intervention by the controller. [21]

Figure 4.1 graphically illustrates the notional in-flight replanning and control process and MBC's role in the system [24]. MBC is a sub-system of the overall mission control architecture. As shown in Figure 4.1, MBC is designed to be used when external factors necessitate a change in the current mission plan. As mentioned above, changes could be the result of a system internal event such as a malfunction or any number of external factors. For this study, three of these factors will be used. These factors include:

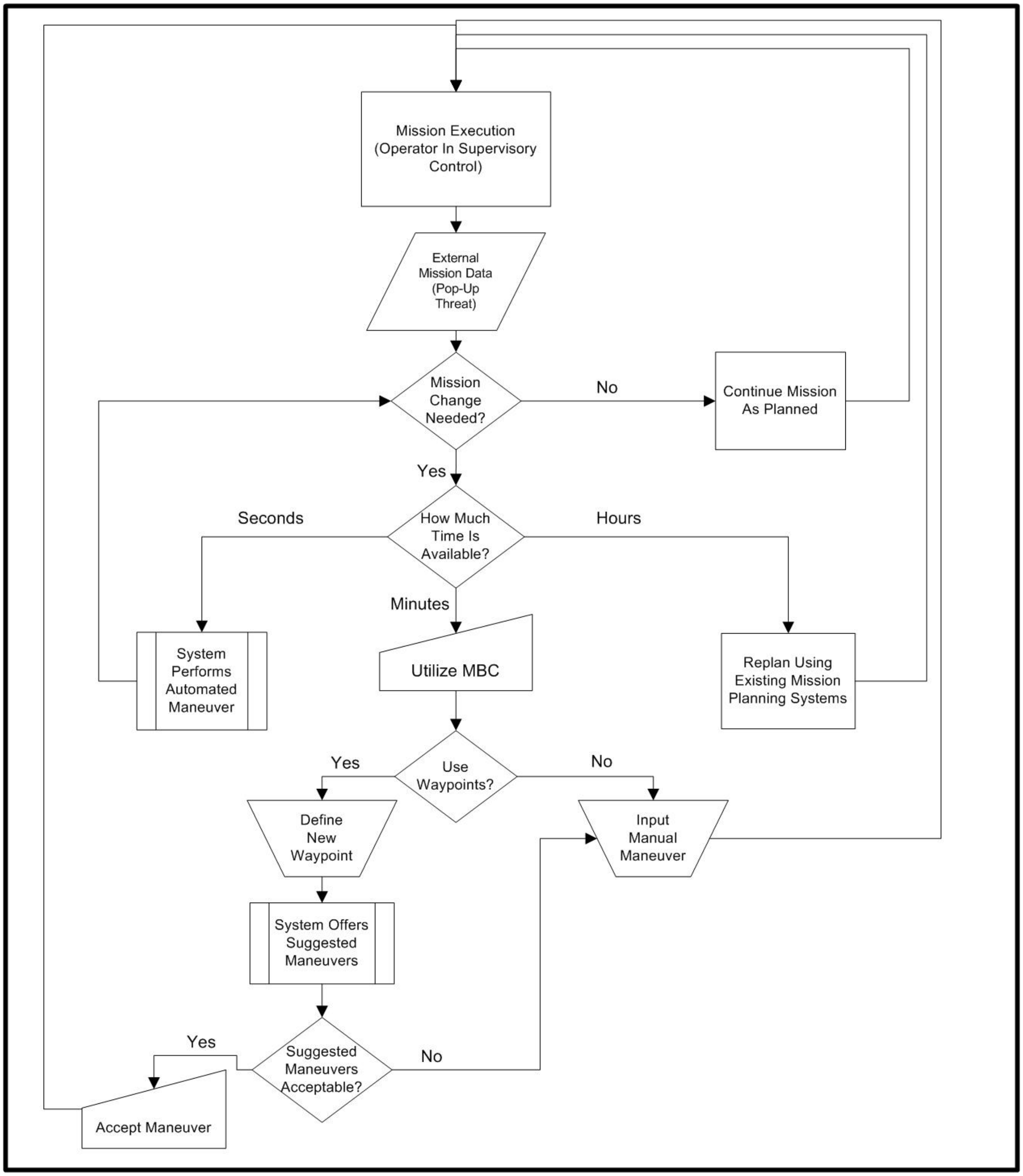

Figure 4.1 Flowchart of Decision Making

Enemy Actions: Pop-up threats that necessitate a flight path change.

Friendly Forces: Control and de-confliction of multiple UCAVs by singe operator.

Retasking: Pop-up target of opportunity.

If a mission change is relatively far off (on the order of an hour), traditional mission planning systems can be used to make re-planning changes. These changes can be made using existing mission planning systems, then uploaded to the aircraft via a data link. The UCAV's systems control system and mission architecture will then executed these changes while the operator may act in a mission-supervisory capacity.

On the other extreme, if only seconds are available there is no time for an operator to be alerted, enter the control loop, decide on a course of action, and execute the maneuver. In this rapid reaction scenario, the system must perform the maneuver automatically. However, if the situation allows minutes or tens of minutes, then MBC may be used. MBC allows much faster implementation than traditional systems and guarantees an achievable solution.

4.3 A Representative Scenario

A notional Suppression of Enemy Air Defenses (SEAD) mission was developed and planned using PFPS. The SEAD mission was chosen because it is the same mission used in the Air Force SAB UCAV study. For a complete description of this route and the methods used to plan it see Appendix D.

4.3.1 Mission Plan. Figure 4.2 shows the notional route and a simple threat layout around the target area (waypoint 7). The mission is approximately 775 nautical miles (NM) in total ground distance and includes a 30 minute orbit (waypoint 5); closely matching the notional SAB scenario of 800 NM with an hour loiter.

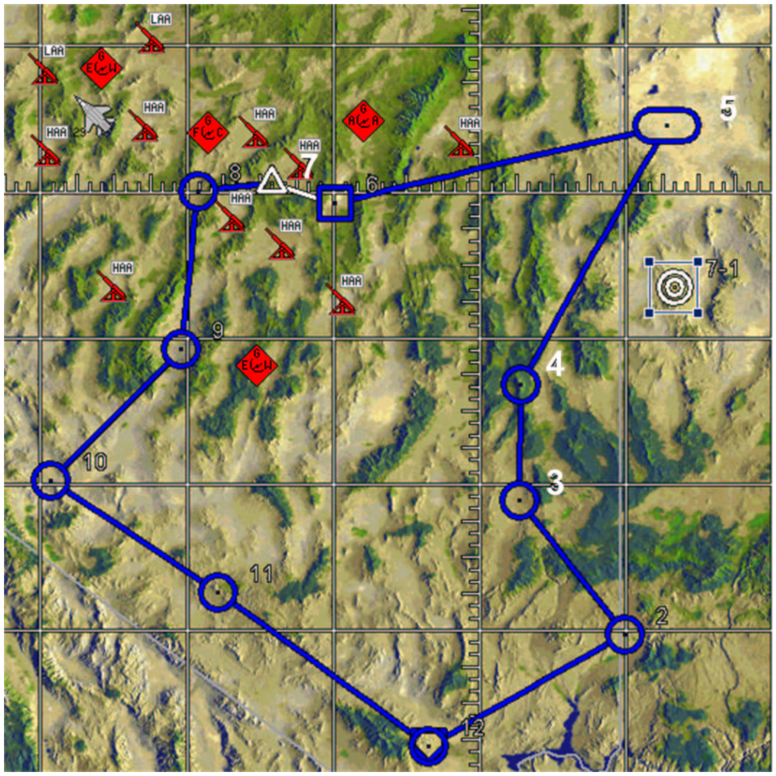

Figure 4.2   Notional UCAV Mission Plan: Planned With PFPS

To illustrate and examine MBC, a segment of the SEAD route was chosen. A 30 NM segment between waypoints 3 and 4 will be the focus of the MBC control. The UCAV is assumed to be travelling at $750 ft/s$ or approximately .75 mach at 10,000 ft. It is assumed that the UCAV has passed thru waypoint 3 when external information causes the need to an in-flight change, see Figure 4.1.

Different scenario inputs will be used to generate the need for mission modifications for both the modified waypoint control and manual input MBC functions. For the modified waypoint control, it assumed that two pop-up threats have been detected along the current flight path. These two threats, two notional Anti-Aircraft

Artillery systems (AAA), are shown with notional threat rings which indicate the lethal range of the systems.[3]

4.3.2 Notional UCAV Capabilities. As detailed in Section 3.1 the UCAV is assumed to have the operating envelope of an F-16A. This is a realistic assumption given the publicly available data on the design and specifications of Boeing's X-45 program. The UCAV is assumed to have a fully capable GPS/INS with autopilot. In addition, the UCAV is assumed to have the following on-board capabilities and systems:

- Synthetic Aperture Radar (SAR)
- Forward Looking InfraRed (FLIR) System
- Data-Link
- Air-to-ground weapons

4.4 Using Maneuver Based Control With Waypoints

> One of the most difficult tasks facing a UAV operator is to make decisions that affect a UAV based on its current tactical environment; which often means mentally transforming their own frame of reference to that of the UAVs[11]

One reason to utilize MBC via waypoints is that it circumvents the difficulty that operators have in trying to orient themselves to the UAV's frame of reference. Waypoints allow the operator to simply use the on screen display of the UCAV ground track to make decisions and command detailed maneuvers. MBC via waypoints essentially allow the user to modify the in flight behavior of the aircraft using similar tools, symbology, and concepts that current mission planning systems use.

- Modify existing waypoint properties

[3]Threat rings are purely notional and presented as an illustrative example only, they should not be taken as representative of an actual AAA system

- Drag existing waypoint
- Create new waypoint

4.4.1 Modifying Existing Waypoints. A waypoint can be modified by either changing its attributes, or by modifying its location. As detailed in Section 2.3, way point attributes include type (orbit, turn, target, etc). In a time critical situation it is unlikely that changing these attributes will address the issue forcing the change. Rather, moving the waypoint is the more likely scenario and is the focus here. Figure 4.3 illustrates the concept of waypoint moving.

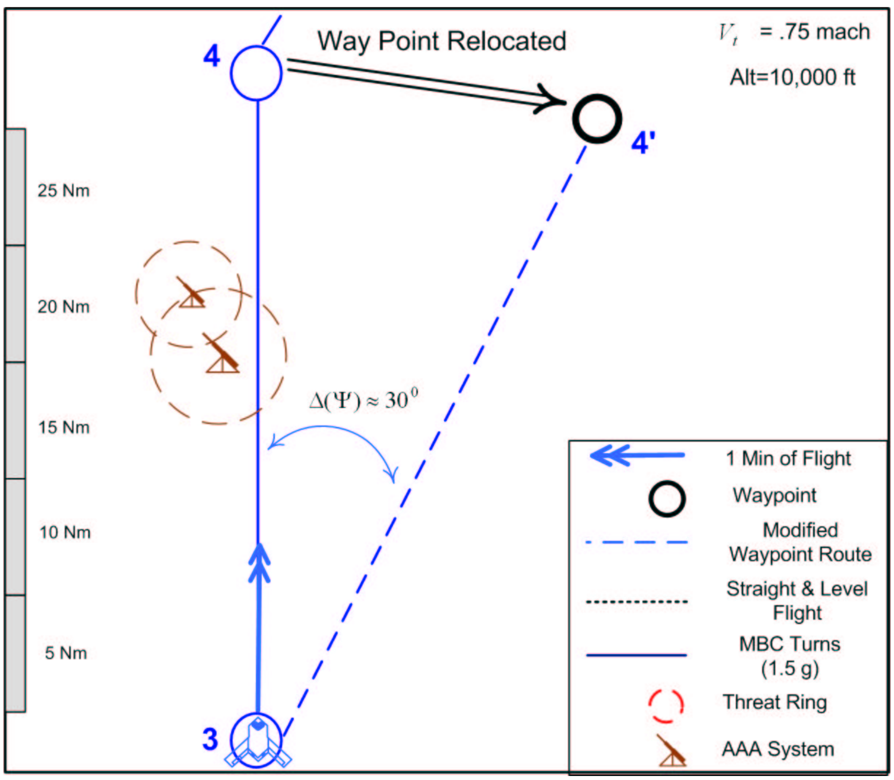

Figure 4.3 Waypoint Moving Maneuver: Notional GUI

While the graphical user interface (GUI) of the operator control station is not the focus here, several features are used that represent the type of information that the UCAV operator would need. The double arrow in Figure 4.3 indicates the distance travelled by the vehicle during one minute of flight. In addition, a distance scale is located on the left side of the figure, a key and graphical representation of current UCAV position are also present.

The GUI of Figure 4.3 as well as those subsequent are designed to illustrate MBC and are purely notional. No existing user executable software was written or exists to create these displays. In an operational implementation of MBC, the actual GUI would be a hybrid of an existing system, such as PFPS, and the functionality developed here. The development of an operationally suitable interface is therefore left for future work.

In Figure 4.3, the operator has moved waypoint 4 to a new location. This new waypoint, approximately 14 NM to the south east is designated $4'$. At the exact instant of time where this snapshot of the UCAV mission, the new waypoint is at a heading change of approximately $\Psi = 30°$ from the current heading. However, if the UCAV operator commands a $\Psi = 30°$ turn at this point, the aircraft will not achieve the desired heading and unless corrected will miss the next waypoint. This is because as seen in Chapter 3, the dynamics of the UCAV are non-linear and the aircraft can not execute an instantaneous turn. This is where MBC becomes useful.

While the onboard INS of the UCAV will correct the path of the vehicle to ensure the waypoint is intersected, the user will not have insight into the path of the vehicle until after the aircraft has reached steady level flight. Utilizing the simple turns generated and stored in the basic maneuver library, MBC allows the operator to see the set of achievable turns that the aircraft can realistically make and the resulting ground track to the desired waypoint.

Utilizing simple geometry, the start location for each achievable turn can be computed and displayed to the user. Figure 4.4 illustrates the manner in which this is done. The point (x_0, y_0) is the actual starting point of the maneuver, chosen here as $(0, y_0)$. The point (x_1, y_1) is the effective starting point, or the point that the turn would take place at if it were to occur instantaneously. The point (x', y') denotes the end of the turn and the start of steady level flight; finally, (x,y) is the location of the relocated waypoint 4.

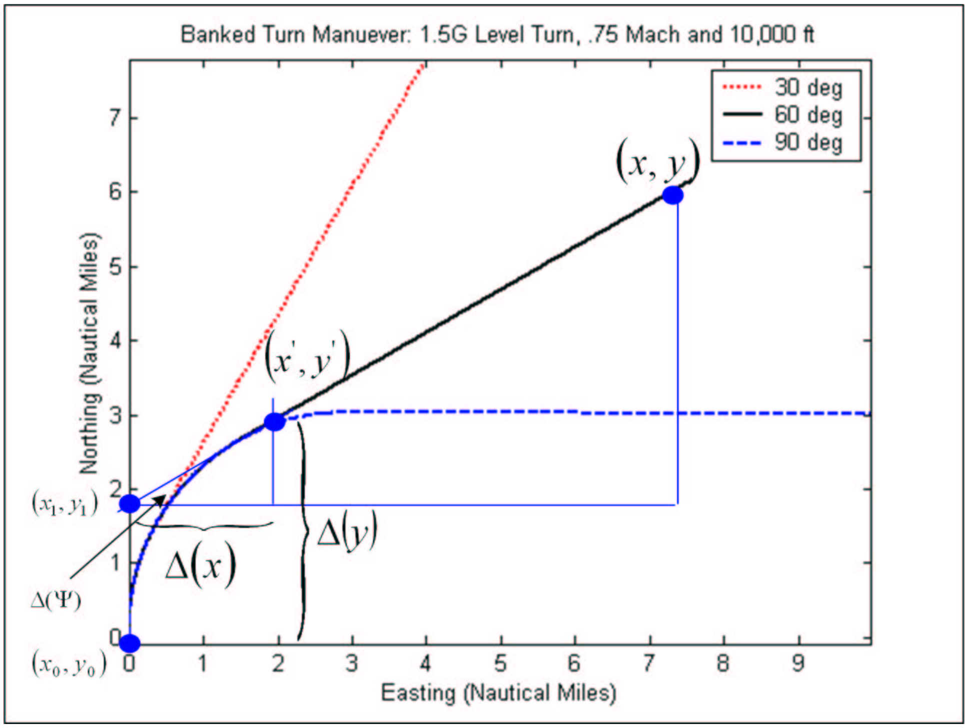

Figure 4.4 Waypoint Moving Maneuver: Geometric Solution

The values of $\Delta(x)$ and $\Delta(y)$ are known, having been saved in the basic maneuver library as a vector containing $(\Delta(x), \Delta(y), \Delta(z), \Delta(\Psi), \Delta(t))$. The operator has defined (x,y) and $\Delta(\Psi)$ is known for each turn type. Thus by application of simple geometry, the starting point can be worked backward using the turn angle Ψ and the known values. Equation 4.1 shows the relationship between the new waypoint and the point at which a given turn must be started to achieve this point.

$$y_0 = y - \Delta(y) - \frac{x - \Delta(x)}{\tan(\Psi)} \tag{4.1}$$

Figure 4.5 shows the culmination of the process described above, illustrating the manner in which MBC can be used to fit turn maneuvers to intersect a waypoint that has been moved. In this particular case, because the waypoint was still ahead of the current position, all turns greater than 90° were not displayed. In addition, the paths of Figure 4.5 were fitted graphically rather than numerically using Equation 4.1. However, the process can easily be automated using the information in the maneuver library and the waypoint and UCAV current path and position.

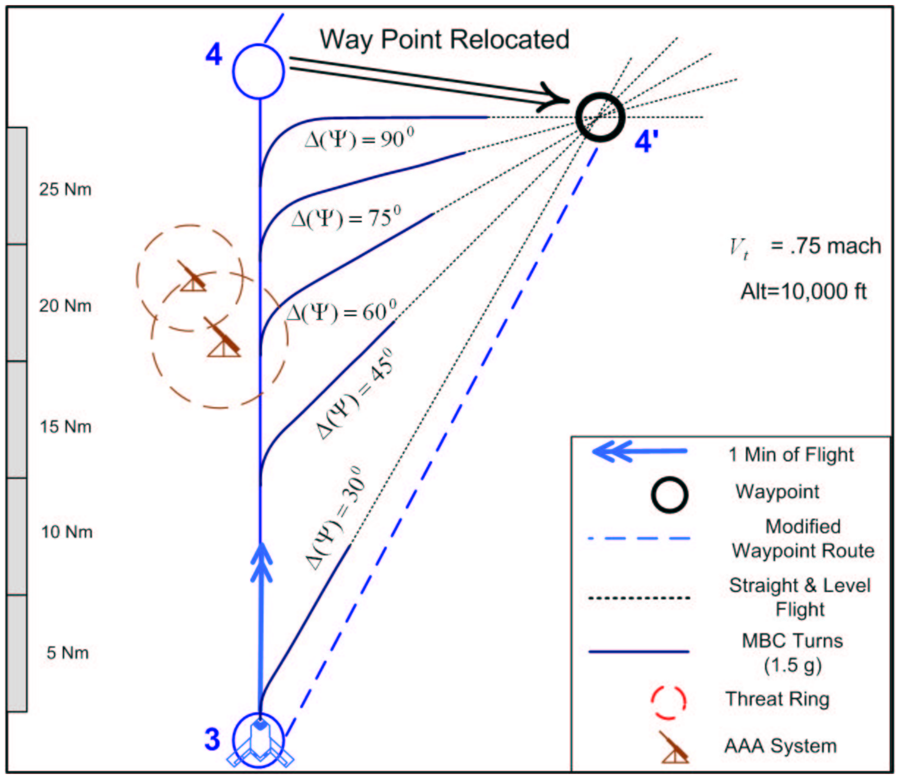

Figure 4.5 Waypoint Moving Maneuver: MBC Solution With Notional GUI

As one can see in Figure 4.5, the 30° and the 45° turns easily provide a route to the new waypoint that avoids the pop-up threat.Under the notional MBC CONOPS developed here, the user is presented the alternative paths shown in Figure 4.5 and then has the option of choosing one, or obtaining a new set of paths computed using a higher g turn type (if such turns are achievable given the aircrafts position in the flight envelope).

Again, this interface is notional only. Other possible interfaces could allow the user to move the waypoint, then drag a 'turn' up and down the current flight path.

The MBC architecture would then insert achievable turns for each starting point as it is dragged along the route.

The advantages of MBC over traditional waypoint navigation is that the operator has complete control over which path is chosen, and only achievable paths are displayed. For some situations the user may find this inconsequential; however in others it may be critical. For the UCAV operator operating multiple UCAVs at once, knowing the exact path of the vehicles allows for easy deconfliction of the vehicles, especially when flying in close formation.

Using MBC, the operator can determine the spacing of the aircraft by having each start its turn at a slightly different position. While turns of $\Delta(\Psi) = 15°$ were used in Figure 4.5, finer turn increments would allow the operator to space the UCAVs by having them execute turns $1°$ or $2°$ apart. It is by modelling the non-linear part of the heading changes, that MBC allows the start point to be precisely determined and this provides the operator the information necessary to execute precise maneuvers.

4.4.2 Inserting A New Waypoint. Another common way for an operator to change a mission plan is to insert a new waypoint in between existing waypoints. Again, the mission leg between waypoints 3 and 4 is used to illustrate MBC's application for mission changes made in this manner. Here, the operator inserts a new waypoint to avoid the pop-up AAA threats as well as a Surface to Air Missile Radar (SAM). Because the new waypoint connects the two existing waypoints, essentiality the operator is calling for a go-around maneuver.

The go-around was one of the advanced maneuvers calculated and utilizing the MBC library, a set of go-around maneuvers can easily be displayed graphically, Figure 4.6. The start point of the go-around maneuvers was chosen so as to avoid the SAM Radar coverage. While only one of the displayed go-around maneuvers actually intersects the new waypoint, others are displayed and are available for user

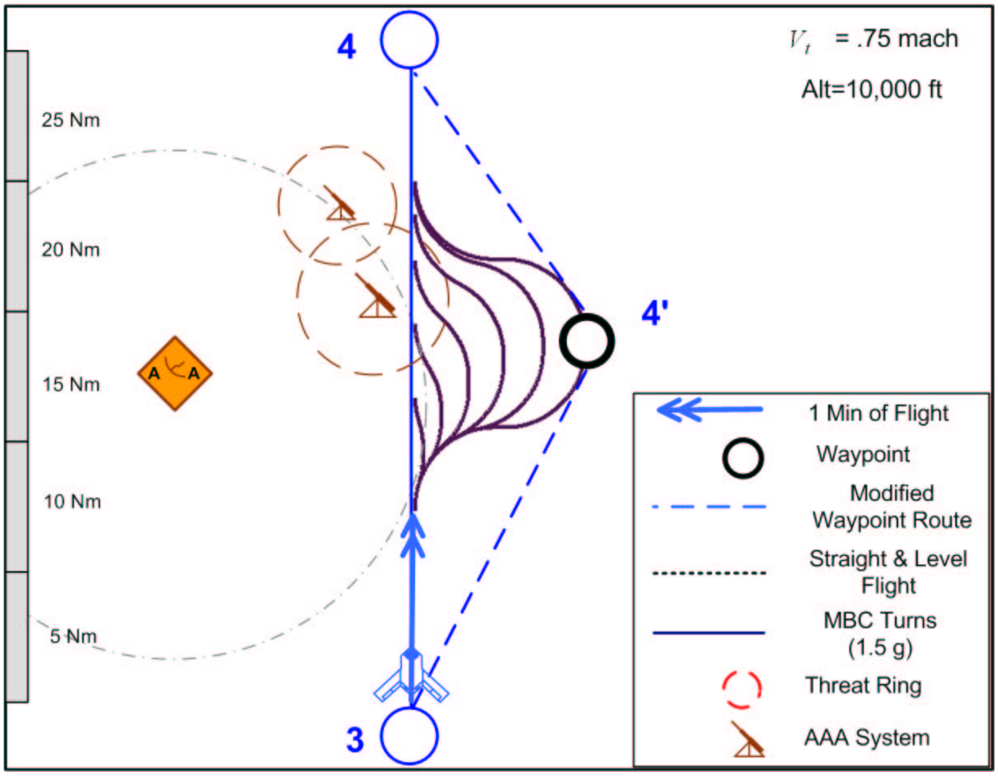

Figure 4.6 Waypoint Bypassing Maneuver

selection. In this case, the waypoint would need to be treated as a guide rather than a must fly point. If desired, the waypoint may be designated as a must-over fly point, as in current systems. However, by treating the waypoint more loosely, a greater number of possible paths are created and the user has more autonomy.

The operator may wish to use the autonomy offered by MBC to choose a route that is shorter than the one that intersects the way point. Reasons for doing this may include:

- Ensure the shortest deviation so as to minimize fuel and total mission time
- UCAV may be imaging other ground targets, so want to minimize distance to the imagery targets
- Operator is controlling multiple UCAVs and wishes each to fly different flight paths.

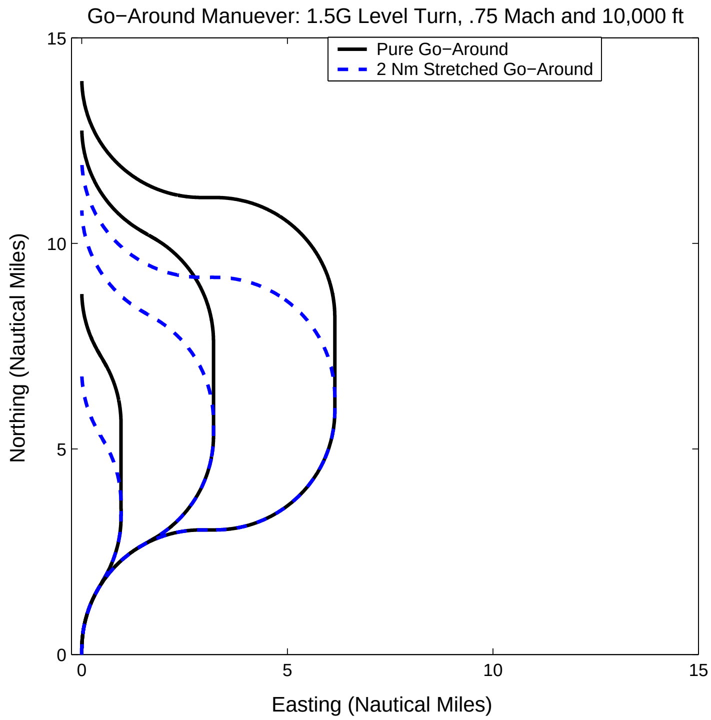

Figure 4.7 "Stretched" Go-Around Maneuver

While three of the go-around maneuvers displayed in Figure 4.6 look to be acceptable, the user may wish to modify them. For example, if the operator wishes to image the threats, yet remain outside its lethal range, the operator may decide that it is only necessary to miss the threat ring by a small amount. In this case, the user can "stretch" the go-arounds by adding a segment of straight and level flights in the middle of the maneuver. In this way, the UCAV still begins its turn before entering the SAM radar coverage, but now the go-around is extended to allow the vehicle to pass out of harms way while maintaining a long segment of straight and level flight suitable to quality image collection.

Figure 4.7 shows what these stretched go-around maneuvers look like. The stretched go-arounds were created by taking the basic go-around and adding a 2 NM segment of straight and level flight (trimmed trajectory 1 as explained in Chapter 3). Any pre-computed maneuver can be stretched by adding segments of straight and level flight in between the other trim turning states. Thus, via this method a

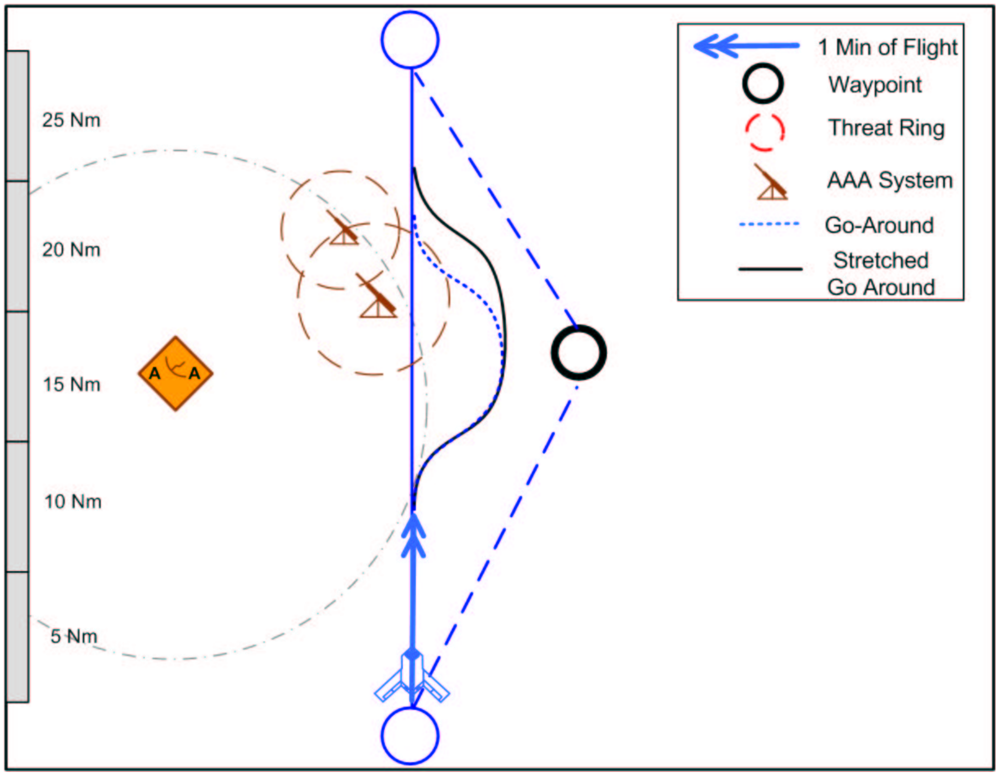

Figure 4.8 "Stretched" Waypoint Bypassing Maneuver

set of maneuvers is created that are both achievable and can be accurately described by the state vectors in the advanced maneuver library.

The ability to stretch pre-computed advanced maneuvers gives the operator a more complete set of alternatives. Figure 4.8 shows the original go-around scenario with the addition of the stretched maneuver. The operator can now tailor the computer suggested maneuver resulting in greater operator control, if desired, than traditional waypoint navigation would allow.

Another possible user interface for inserting waypoints would be the capability to "grab" a segment of the flight path and "pull" it out to form a go-around. MBC would then fit turns and go arounds in as the user drags the path. As the route is pulled away from its initial position, lower g turns are required to make the maneuver.

By inserting multiple new waypoints, extensive changes can be made to the existing route. This allows the operator to define and later refine the flight path of the vehicle. Indeed, adding waypoints one after the other is how most routes are created initially. However, with the aid of MBC, only executable flight paths are created and the operator has precise knowledge of that path *a priori* . In cases of multiple UCAV control, threat avoidance, or simply the desire to limit expose to unknowns on the ground, knowledge of the exact flight path of the vehicle gives the operator one more piece to the mission puzzle.

4.5 Manual Maneuver Input

Another MBC method for the operator to make in-flight mission changes to the flight of the vehicle is to have them manually input the desired maneuver or sets of maneuvers they want the UCAV to perform. Manual maneuver input assumes a well trained operator who wishes to exercise a greater degree of control over the vehicle than the waypoint method of MBC described above. Two primary modes of manual input are envisioned, a graphical method using maneuver icons and a more simple command based system.

Icon based manual input utilizes the same familiar set of tools used in current mission planning programs and the waypoint MBC methods described previously. In this case, icons are created that represent the specific pre-computed maneuvers stored in the maneuver library. The operator can then graphically take these icons and manipulate them to form the specific flight path desired.

The icons for each maneuver are the graphical representations of the (x,y) state vectors computed via Matlab. The graphical connections of the icons can then be translated back into a state vector in much the same way that the maneuver library was created by adding the different state vectors of various trimmed trajectory. Because the icons represent time correlated state vectors, there are a few rules that must be followed when creating routes. Among the rules for icon based MBC are:

- Icons must be connected "end-to-start" (end annotated with arrow)
- Turn icons not scale-able but may be oriented in any direction
- Straight and level icons are fully scalable and may be oriented in any direction
- Altitude and velocity changes may necessitate a change in icons used

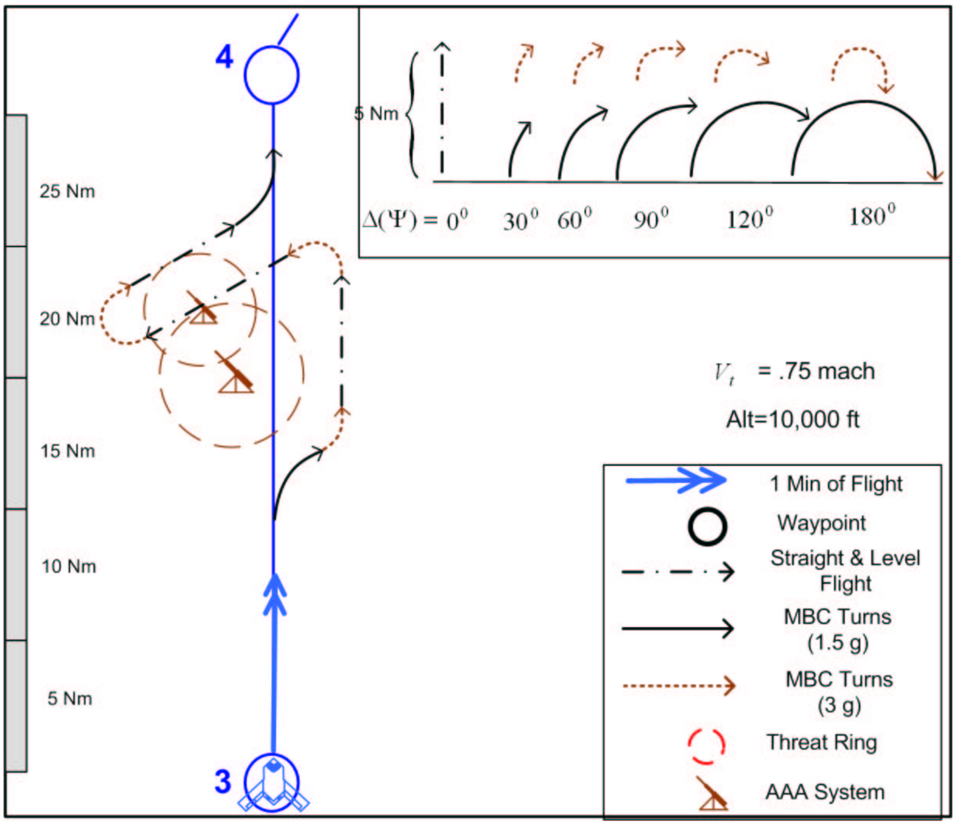

Figure 4.9 Manually Input Maneuver

By following the rules for icon usage, the operator can quickly modify existing routes and plan detailed flight paths. Figure 4.9 illustrates a manually created route adjustment made using icons. In Figure 4.9 icons for turns between $\Psi = 0°$ and $\Psi = 180°$ are shown in the upper right corner for level turns of 1.5 and 3g. These icons were chosen for illustrative purposes only. In an actual applications various icons could be displayed, depending on the GUI and MBC manual input CONOPS.

In this scenario, the operator modifies the route so as to avoid the first threat, but then overfly the second AAA system in order to engage and destroy the target. The UCAV then makes a pass out side the lethal range of the AAA system and using the onboard sensors performs a Battle Damage Assessment (BDA). Table 4.1 details the maneuvers created using the icons and displayed in Figure 4.9.

Table 4.1 Basic Manuever Composing Modified Flight Path

Manuever Order	Maneuver Description
1	1.5g 60° Right Turn
2	5 NM Straight and Level Flight
3	3g 120° Left Turn
4	6 NM Straight and Level Flight
5	3g 180° Right Turn
6	4.75 NM Straight and Level Flight
7	1.5g 30° Left Turn

In the above scenario, the operator wanted to exercise detailed control over the UCAV. MBC allows this to be accomplished quickly and with out the operator having to fly the vehicle thru the entire maneuver manually, freeing them for other tasks.

Icon based manual MBC control gives the operator the ability to control the vehicle with the precision of using an actual stick and throttle without the need for stick and throttle. Thus, the operator work load is lessened without hampering their control. In addition, by removing the reliance on piloting skills, operators can concentrate on a more comprehensive set of mission tasks.

4.6 MBC: Capabilities and Limitations

As shown here, MBC has the capability to aid future UCAV operators in the quest to achieve a truly variable autonomy system. The three methods explored here for implementing MBC: modifying waypoints, inserting waypoints, and manual icon based control, fall on different parts of the automation scale, Figure 1.4, Section 1.3.

Major capabilities and benefits of MBC include:

- Ability to increase manual intervention in otherwise autonomous system
- Improve overall situational awareness by allowing operator to observe flight path *a priori*
- Increased mission effectiveness by allowing time critical changes to be made to ongoing missions

MBC allows the operator to trade off tasks with the computer. Modifying waypoints just moves the operator slightly to the more manual side of the scale, where the system still performs a majority of the tasks. Inserting waypoints modifies the route greater and allows the user to take a more active role in determining the flight path of the vehicle. Finally, manual control via icons allows the user to completely specify a detailed flight path while still relying on the UCAV control system to execute the desired path.

While the benefits on MBC are numerous, MBC is a subsystem in a very complex UCAV system. MBC is not intended to be a total solution, rather it is a piece of the command and control equation. Thus, one needs to recognize potential MBC limitations when evaluating its total benefit. Known limitations of MBC include:

- Need to pre-compute trimmed trajectories and save data in complex library system.
- Routes not necessarily optimized.
- Requires knowledgeable human operator to exploit MBC capabilities.

V. Conclusions and Recommendations

5.1 Conclusions

Previous UCAV studies have shown a need for the ability to vary the amount of control exercised by the operator. To this end, Manuever Based Control (MBC) allows UCAV operators to increase the level of manual control over the air vehicle in situations where quick response time is required. Thus, human operators can choose to enter into the decision making loop, where they excel, while allowing the automated system to retain basic flight control functions.

As proposed here, MBC allows the human operator to make timely in-flight mission changes by modifying and inserting waypoints into the mission plan as well as by manual icon-based input. Situational awareness is enhanced by utilizing the waypoint concept and other existing tools and techniques and by eliminating the need of operators to attempt to transform their frame of reference into that of the UCAV.

Finally, MBC ensures effective control integrity of vehicle by using pre-computed flight paths. Flight paths are chosen so that they reside within the safe flight envelope of the aircraft and the control constants associated with each trajectory are selected to ensure stability. These flight paths are computed numerically using a non-linear model.

Trading control thru the use of MBC will increase overall mission effectiveness and success by allowing the operator to vary the autonomy of the vehicle. Thus, the goal of a variable autonomy UCAV is fully achievable thru the use of Manuever Based Control.

5.2 Recommendations for Follow On Work

The work presented here is intended to verify the operating concepts and theory of MBC. However, before being used operationally, significant work needs to be accomplished. Among the recommendations for further work are:

- Refine the MBC CONOPS
- Develop user interface with realistic GUI
- Expand advanced maneuver library to include advanced fighter maneuvers composed of non-trimmed states
- Include optimized trajectories in manuever library
- Include maneuvers involving changes in velocity
- Develop methods to ensure a Time-On-Target constraint.

Refining the CONOPS for MBC use will allow human factors engineers to develop and integrate an operationally suitable GUI into existing control systems. The user interface will be critical to making MBC easy enough to use for short suspense re-taskings, yet allow it to retain a robust and flexible mission management capability.

In addition, more advanced maneuvers need to be included in the available library. By applying the same general procedure used to model basic trimmed trajectories, more advanced maneuvers can be simulated and stored. Basic fighter maneuvers such as split-s, pitchback, etc. as well as weapon delivery maneuvers such as the pop-up should be modelled and added to the advanced manuever library.

Appendix A. Matlab M Files Used To Model UCAV Dynamics

A.1 ucav.nld

```
 function yd=subf16etc(y)
%  Modified by Capt Alexander Walan
%  yd = ucav(y)
%Note: This routine is similar to subf16.m, the equations of motion are the same,
%      but the input and output format is significantly different.
%  This routine outputs a vector, yd, for the computer model of an F-16 aircraft.
%  This is called by etc.mdl to run nonlinear F-16 simulations.
%  Prior to running, this m-file must be edited to set the proper c.g. location
%  (see 'xcg = ' below). The nominal value of xcg=.35
%  generally produces unstable open loop A/C dynamics.  A value of xcg<=.3 will
%  generally produce stable A/C dynamics.  If you choose to run xcg>=.35 you
%  should definitely add a stablizing feedback
%  controller or the plane will be difficult to control by the pilot.
%
%  The first 16 components of the input vector y is the aircraft state vector, x,
%  where:
%          y(1) = air speed, VT    (ft/sec)
%          y(2) = angle of attack, alpha  (rad)
%          y(3) = angle of sideslip, beta (rad)
%          y(4) = roll angle, phi  (rad)
%          y(5) = pitch angle, theta  (rad)
%          y(6) = yaw angle, psi  (rad)
%          y(7) = roll rate, P  (rad/sec)
%          y(8) = pitch rate, Q  (rad/sec)
%          y(9) = yaw rate, R  (rad/sec)
%          y(10) = northward horizontal displacement, pn  (feet)
%          y(11) = eastward horizontal displacement, pe  (feet)
%          y(12) = altitude, h  (feet)
```

```
%         y(13) = engine thrust dynamics lag state, pow
%            y(14) = elevator actuator deflection, deg
%            y(15) = aileron actuator deflection, deg
%            y(16) = rudder actuator deflection, deg
%  The next 4 components of the input vector y are the control input commands
%
%            y(17) = throttle command,  0 <  thtlc < 1.0
%            y(18) = elevator command, deg
%            y(19) = aileron command, deg
%            y(20) = rudder command, deg
% The Last 2 components of the input vector y are the wind inputs
%               y(21)=wind velocity (ft/sec)
%               y(22)=wind direction (deg)
%  The first 16 components of the output vector yd is dx/dt
%  (i.e., the aircraft state vector derivative), which is the
%  derivatives of the first 16 y vector components :
%       yd(1) = derivative of air speed, VT    (ft/sec^2)
%       yd(2) = derivative of angle of attack, alpha  (rad/sec)
%       yd(3) = derivative of angle of sideslip, beta (rad/sec)
%       yd(4) = derivative of roll angle, phi  (rad/sec)
%       yd(5) = derivative of pitch angle, theta  (rad/sec)
%       yd(6) = derivative of yaw angle, psi  (rad/sec)
%       yd(7) = derivative of roll rate, P  (rad/sec^2)
%       yd(8) = derivative of pitch rate, Q  (rad/sec^2)
%       yd(9) = derivative of yaw rate, R  (rad/sec^2)
%       yd(10) = derivative of northward horizontal displacement, pn  (feet/sec)
%       yd(11) = derivative of eastward horizontal displacement, pe  (feet/sec)
%       yd(12) = derivative of altitude, h  (feet/sec)
%       yd(13) = derivative of engine thrust dynamics lag state, pow  (1/sec)
%       yd(14) = derivative of elevator actuator deflection, deg/sec
%       yd(15) = derivative of aileron actuator deflection, deg/sec
%        yd(16) = derivative of rudder actuator deflection, deg/sec
%  The last 6 components of the output vector yd are the linear and angular
%  accelerations
```

```
%  that the pilot would sense and are the commands to be sent to the centrifuge:
%            yd(17) = output lin_accel_x (ft/sec^2)
%            yd(18) = output lin_accel_y (ft/sec^2)
%            yd(19) = output lin_accel_z (ft/sec^2)
%            yd(20) = output rot_accel_x (rad/sec^2)
%            yd(21) = output rot_accel_y (rad/sec^2)
%            yd(22) = output rot_accel_z (rad/sec^2)
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%  Script/Function calls:
%   adc         cx      cy      cz
%   tgear       cl      cm      cn
%  pdot        dlda    dldr
%  thrust      dnda    dndr
%  dampp
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
yd=zeros(22,1); thtlc=y(17); elc=y(18); ailc=y(19); rdrc=y(20);
% Wind Velocity is wv, Wind Direction is given as omg
wv=y(21); omg=y(22)/57.3;
% The following is the c.g.location which can be modified(nominal xcg=.35)
xcg=.35;
 s=300;b=30;cbar=11.32;rm=1.57e-3;xcgr=.35;he=160.0;
c1=-.770;c2=.02755;c3=1.055e-4;c4=1.642e-6;c5=.9604;c6=1.759e-2;c7=1.792e-5;
c8=-.7336;c9=1.587e-5; rtod=57.29578;g=32.17;
%
vt=y(1);alpha=y(2)*rtod;beta=y(3)*rtod;
phi=y(4);theta=y(5);psi=y(6); p=y(7);q=y(8);r=y(9);alt=y(12);
[amach,qbar]=adc(vt,alt);
%
pow=y(13); if(thtlc>=1.0),thtlc=1.0; elseif(thtlc<0.),thtlc=0.;
end; cpow=tgear(thtlc); yd(13)=pdot(pow,cpow);
t=thrust(pow,alt,amach);
%
el=y(14); sel=sign(el); yd(14)=20.202*(elc-el); if(abs(el)>=25 &
```

```
sign(yd(14))==sel) yd(14)=0; el=sel*25; end if(abs(yd(14))>=60)
yd(14)=sign(yd(14))*60; end
%
ail=y(15); sal=sign(ail); yd(15)=20.202*(ailc-ail);
if(abs(ail)>=21.5 & sign(yd(15))==sal) yd(15)=0; ail=sal*21.5; end
if(abs(yd(15))>=80) yd(15)=sign(yd(15))*80; end
%
rdr=y(16); srd=sign(rdr); yd(16)=20.202*(rdrc-rdr);
if(abs(rdr)>=30 & sign(yd(16))==srd) yd(16)=0; rdr=srd*30; end
if(abs(yd(16))>=120) yd(16)=sign(yd(16))*120; end
%
cxt=cx(alpha,el); cyt=cy(beta,ail,rdr); czt=cz(alpha,beta,el);
dail=ail/20;drdr=rdr/30;
clt=cl(alpha,beta)+dlda(alpha,beta)*dail+dldr(alpha,beta)*drdr;
cmt=cm(alpha,el);
cnt=cn(alpha,beta)+dnda(alpha,beta)*dail+dndr(alpha,beta)*drdr;
tvt=.5/vt;b2v=b*tvt;cq=cbar*q*tvt; d=dampp(alpha);
cxt=cxt+cq*d(1); cyt=cyt+b2v*(d(2)*r+d(3)*p); czt=czt+cq*d(4);
clt=clt+b2v*(d(5)*r+d(6)*p); cmt=cmt+cq*d(7)+czt*(xcgr-xcg);
cnt=cnt+b2v*(d(8)*r+d(9)*p)-cyt*(xcgr-xcg)*cbar/b;
cbta=cos(y(3));u=vt*cos(y(2))*cbta;
v=vt*sin(y(3));w=vt*sin(y(2))*cbta;
sth=sin(theta);cth=cos(theta);sph=sin(phi);
cph=cos(phi);spsi=sin(psi);cpsi=cos(psi);
qs=qbar*s;qsb=qs*b;rmqs=rm*qs; gcth=g*cth;qsph=q*sph;
ax=rm*(qs*cxt+t);ay=rmqs*cyt;az=rmqs*czt; udot=r*v-q*w-g*sth+ax;
vdot=p*w-r*u+gcth*sph+ay; wdot=q*u-p*v+gcth*cph+az; dum=(u*u+w*w);
yd(1)=(u*udot+v*vdot+w*wdot)/vt; yd(2)=(u*wdot-w*udot)/dum;
yd(3)=(vt*vdot-v*yd(1))*cbta/dum; yd(4)=p+(sth/cth)*(qsph+r*cph);
yd(5)=q*cph-r*sph; yd(6)=(qsph+r*cph)/cth;
yd(7)=(c2*p+c1*r+c4*he)*q+qsb*(c3*clt+c4*cnt);
yd(8)=(c5*p-c7*he)*r+c6*(r*r-p*p)+qs*cbar*c7*cmt;
yd(9)=(c8*p-c2*r+c9*he)*q+qsb*(c4*clt+c9*cnt);
t1=sph*cpsi;t2=cph*sth;t3=sph*spsi;
```

```
s1=cth*cpsi;s2=cth*spsi;s3=t1*sth-cph*spsi;
s4=t3*sth+cph*cpsi;s5=sph*cth;s6=t2*cpsi+t3;
s7=t2*spsi-t1;s8=cph*cth;
% Compute x,y,z with wind direction and magnitude added
yd(10)=u*s1+v*s3+w*s6+wv*(cos(omg)*cos(psi));
yd(11)=u*s2+v*s4+w*s7+wv*(sin(omg)*sin(psi));
yd(12)=u*sth-v*s5-w*s8;
if(alt<=0 & sign(yd(12))<0)   % can't fly underground
yd(12)=0; end
xa=15.0;                 % sets distance normal accel is in front of the c.g.
                        %  (xa=15.0 at pilot)
az=az-xa*yd(8);          % moves normal accel in front of c.g.
ay=ay+xa*yd(9);          % moves side accel in front of c.g.
yd(17)=ax;                   % output lin_accel_x (ft/sec^2)
yd(18)=ay;                   % output lin_accel_y (ft/sec^2)
yd(19)=az;                   % output lin_accel_z (ft/sec^2)
yd(20)=yd(7);                % output rot_accel_x (rad/sec^2)
yd(21)=yd(8);                % output rot_accel_y (rad/sec^2)
yd(22)=yd(9);                % output rot_accel_z (rad/sec^2)
```

A.2 trimmer.mod

```
function
[Xequil,Uequil]=trimmer(Xguess,Uguess)
% [xequil,uequil]=trimmer(xguess,uguess)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%  Program: trimmer
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%  This program numerically calculates the equilibrium state and control
% vectors of an  F-16 model given certain parameters.  Inputs include
% initial guesses for the equilibrium state and input vectors. If the
% routine is called with no inputs the user will be prompted to key the
% equilibrium initial guesses in by hand.  The user will be prompted to
% pick one of the following A/C orientation options and provide the
```

```
% desired altitude, airspeed, gamma, turn rate, pitch rate,etc. :
%     1. Wings Level (gamma = 0)
%     2. Wings Level (gamma <> 0)
%     3. Steady Constant Altitude Turn
%     4. Steady Pull Up
%  The user will also be prompted for the number of iterations to be used
% in the numerical minimization search.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
%  states:                                            controls:
%   x1 = Vt     x4 = phi       x7 = p     x10 = pn    u1 = throttle
%   x2 = alpha  x5 = theta  x8 = q    x11 = pe        u2 = elevator
%   x3 = beta   x6 = psi    x9 = r    x12 = alt       u3 = aileron
%                                     x13 = pow       u4 = rudder
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%  Script/Function calls:
%   getinput
%  adc
%  clf16
%  fminsa
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
global ay az
 format long
 if(nargin==2) x=Xguess; u=Uguess; else x=zeros(13,1);
u=zeros(4,1); end

%   gamma singam rr  pr  tr  phi cphi sphi thetadot coord stab orient
const = [0.0    0.0   0.0 0.0  0.0 0.0 1.0  0.0   0.0      0.0 0.0
1];
 rtod = 57.29577951; orient=1;
%orient = menu('Choose an A/C Orientation','Wings Level (gamma = 0)',...
%'Wings Level (gamma <> 0)','Steady Turn','Steady Pull Up');
const(12) = orient;
 ndof = 6;
```

```
 if orient == 1
   x(1) = Xguess(1);
   x(12) = Xguess(12);
end
 if orient == 2
   x(1) = input('Velocity Vector (ft/s) (VT): ');
   x(12) = input('Altitude (ft) (h): ');
   gamm = input('Gamma (deg): ');
   const(1) = gamm/rtod;
   const(2) = sin(const(1));
end if orient == 3
   x(1) = input('Velocity Vector (ft/s) (VT): ');
   x(12) = input('Altitude (ft) (h): ');
   psidot = input('Turn Rate (deg/s) (Psi dot): ');
   const(5) = psidot/rtod;
end
 if orient == 4
   x(1) = input('Velocity Vector (ft/s) (VT): ');
   x(12) = input('Altitude (ft) (h): ');
   thetadot = input('Pitch Rate (deg/s) (Theta dot): ');
   const(9) = thetadot/rtod;
end
% Set up the initial guess for the state and control vectors
if nargin~=2 disp(' ') disp('Next Input The Initial Guess For The
Equilibrium State And Control Vectors') disp('Remember To Match
The Altitude and Air Speed You Just Keyed In:') disp(' ') getinput
end
%
yesno = 1; clear s if orient == 3 s(1)=u(1); s(2)=u(2); s(3)=u(3);
s(4)=u(4); s(5)=x(2); s(6)=x(4); s(7)=x(5); else s(1)=u(1);
s(2)=u(2); s(3)=x(2); end lcost=1; while lcost>1E-4;
   options = [0 1.0E-9 1.0E-9 0 0 0 0 0 0 0 0 0 0 1000];
  options(14) = 1000;
   [s,options,x,u,fcost,lcost] = fminsa('clf16',s,options,[],x,u,const);
```

```
  [amach,qbar]=adc(x(1),x(12));
  fprintf('\n');
  if ndof > 3
     fprintf('Throttle (percent):      %g\n', u(1))
     fprintf('Elevator (deg):          %g\n', u(2))
     fprintf('Ailerons (deg):          %g\n', u(3))
     fprintf('Rudder (deg):            %g\n', u(4))
     fprintf('Angle of Attack (deg):       %g\n', rtod*x(2))
     fprintf('Sideslip Angle (deg):        %g\n', rtod*x(3))
     fprintf('Pitch Angle (deg):       %g\n', rtod*x(5))
     fprintf('Bank Angle (deg):        %g\n', rtod*x(4))
     fprintf('Normal Acceleration (g): %g\n', az/32.2)
     fprintf('Lateral Accereration (g):    %g\n', ay/32.2)
     fprintf('Dynamic Pressure (psf):      %g\n', qbar)
     fprintf('Mach Number:         %g\n', amach)
  else
     fprintf('Throttle (percent):      %g\n', u(1))
     fprintf('Elevator (deg):          %g\n', u(2))
     fprintf('Alpha (deg):         %g\n', x(2)*rtod)
     fprintf('Pitch Angle (deg):       %g\n', x(5)*rtod)
     fprintf('Normal Acceleration (g): %g\n', az/32.2)
     fprintf('Dynamic Pressure (psf):      %g\n', qbar)
     fprintf('Mach Number:         %g\n', amach)
  end
 fprintf('\n')
  fprintf('Initial Cost Function:      %g\n', fcost)
  fprintf('Final Cost Function:        %g\n', lcost)
end Xequil=x; Uequil=u;
```

Appendix B. Matlab M Files Used To Generate Data

B.1 Files Written To Compute Basic Manuever

B.1.1 Computing Straight and Level Flight.

```
% Captain Alexander M.G. Walan: GAE-03M-09
% Written January 2003
% Thesis Code: data_gen_man_1
% This code creates the manuever library for straight and level flight
% Library is a array of form: S=[i,x,p,q,r] where
% S=[index,X,turn number,vel, altitude]
% X=[East,North,Down,psi,Time,delta]
%Generation of state vector for steady level flight
clear tic load mode1 Man_Library=zeros(1500,18,1,3,3); for i=1:3
    for ii=1:3
        if ii==1;
        vt=500;
    elseif ii==2;
            vt=750;
        else;
            vt=1275;
        end
    if i==1;
        alt=1000;
    elseif i==2;
            alt=10000;
        else;
            alt=30000;
        end
Xguess=[vt;0;0;0;0;0;0;0;0;0;0;alt];Uguess=[.2,-.2,0,0];
[Xequil,Uequil]=trimmer_mod(Xguess,Uguess); xeqq(:,ii,i)=Xequil;
ueqq(:,ii,i)=Uequil'; end end
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Steady Level Flight
for c=1:3;
    for d=1:3;
       index_number= [c,d];
       index_number
        xeq=xeqq(:,c,d);
        ueq=ueqq(:,d,d);
    [time,states]=sim('ucav',[0,120]);
    m=length(states);
    for i=1:1
    x_vector=zeros(1:m,18);
    x_vector(1:m,1:16)=states(:,1:16);
    x_vector(:,17)=time(:);
     delta_x=states(m,11);
    delta_y=states(m,10);
    delta_z=states(m,12);
    delta_t=time(m);
    delta_vector=[delta_x;delta_y;delta_z;delta_t];
    x_vector(1:4,18)=delta_vector;
    %
    Library(1:m,:,i)=x_vector(:,:);
end
    Man_Library(1:m,:,1,c,d)=Library(:,:);
    clear Library;
end end toc; time=toc/60 ML_1=[Man_Library(:,11,:,:,:)
Man_Library(:,10,:,:,:) Man_Library(:,12,:,:,:)
Man_Library(:,6,:,:,:) Man_Library(:,17:18,:,:,:)]; save ML_1;
```

B.1.2 Computing 1.5g Level Turn.

```
%Captain Alexander M.G. Walan: GAE-03M-09
% Written January 2003
% Thesis Code: data_gen_man_4
% This code creates the manuever library for trimm trajecotry #4,
```

```
% a 1.5g level turn.
% Library is a array of form: S=[i,x,p,q,r] where
% S=[index,X,turn number,vel, altitude]
% X=[East,North,Down,psi,Time,delta]
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Generate the equilibrium values for the 3 velocities and 3
%altitudes we will calculate.
tic clear load mode3 for d=1:3
    for c=1:3
        if c==1;
        vt=500;
    elseif c==2;
            vt=750;
        else;
            vt=1275;
        end
    if d==1;
        alt=1000;
    elseif d==2;
            alt=10000;
        else;
            alt=30000;
        end
Xguess=[vt;0;0;0;0;0;0;0;0;0;0;alt];Uguess=[.2,-.2,0,0];
[Xequil,Uequil]=trimmer_mod(Xguess,Uguess); xeqq(:,c,d)=Xequil;
ueqq(:,c,d)=Uequil'; end end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Begin by initializing library and running the trim trajectory case.
Man_Library=zeros(1800,18,24,3,3);
%Right Hand Turns
clear c d for c=1:3;
    for d=1:3;
       index_number= [c,d];
       index_number
```

```
        xeq=xeqq(:,c,d);
        ueq=ueqq(:,c,d);
            if d==3;
                kgi=-.25
                kgp=-2
            else
             kgi=0
                kgp=0
            end
% Compute Trim Trajectory for each Velcoity & Altitude
    phiamp=45;phiamp2=-45;
    t1=200;t0=0;
    [t_total,xout_total,YY] = sim('ucav2', [0,180]);
    clear t1
%  Compute Turns
    number=12;
            for i=1:number
                phiamp=45;phiamp2=-45;
                turn=i*15
                turn_index=find(xout_total(:,6)*57.3>turn-.1);
                    if d==3;
                         t1=t_total(turn_index(1))+1;
                    else
                         t1=t_total(turn_index(1))-.3;
                    end
                [time,states]=sim('ucav2',[0,t1+15]);
                 m=length(states);
                 n=find(states(:,6)*57.3<(states(m,6)*57.3-.2));
                Lngh=length(n);
                I(i)=n(Lngh);
    %
                delta_x=states(I(i),11);
                delta_y=states(I(i),10);
                delta_z=states(I(i),12);
```

```
            delta_t=time(I(i));
            delta_vector=[delta_x;delta_y;delta_z;delta_t];
            x_vector=zeros(length(I(i)),18);
            x_vector(1:I(i),1:16)=states(1:I(i),1:16);
            x_vector(1:I(i),17)=time(1:I(i));
            x_vector(1:4,18)=delta_vector;
            Library(1:I(i),:,i,1)=x_vector(:,:);
  %
            psi_rt=Library(I(i),6,i)*57.3
            Library(1:I(i),:,number+i,1)=[x_vector(:,1:5) -1*x_vector(:,6)
            x_vector(:,7:10)
            -1*x_vector(:,11) x_vector(:,12:18)];
            psi_left=Library(I(i),6,number+i)*57.3
          end
% Outer Loop
        mm=length(Library);
        mmm=length(Library(1,1,:));
        Man_Library(1:mm,:,1:mmm,c,d)=Library(:,:,:);
        clear Library;
    end
 %Clear Variables in space
 %clear t_total xout_total
    end
end toc; time=toc/60
%ML_4=[East, North, Down,Psi,Time, Delta]
ML_4t=[Man_Library(:,11,:,:,:) Man_Library(:,10,:,:,:)
Man_Library(:,12,:,:,:) Man_Library(:,6,:,:,:)
Man_Library(:,17:18,:,:,:)];
save ML_4t;
```

B.2 Files Written To Compute Advanced Manuevers

B.2.1 Computing the Offset Maneuver.

```
% Captain Alexander M.G. Walan: GAE-03M-09
```

```
% Written January 2003
% Thesis Code: basic_offset_man
% This code creates the manuever library for a series of offset manuevers
% Library is a array of form: S=[i,x,p,q,r,s] where
% S=[index,X,turn number,vel, altitude,wind condition]
% X=[East,North,Down,psi,Time,delta]
clear tic
% Decide Which Basic Turn Type To Use
for tp=1:1
    if tp==1
        load ML_4;
        ML=ML_4;
    elseif tp==2
        load ML_5;
        ML=ML_5;
    elseif tp==3
        load ML_6;
        ML=ML_6;
    end
%Define # of Turns and Altitude Blocks to Use:
%Note, there is no data for v=500 ft/s and Alti=30,000 due to
%flight envelope restrictions.
number=12 for q=1:3
    for r=1:3
        for p=1:number
% Obtain only non-zero values for given turn:
Index1=find(ML(:,4,p,q,r)>0); v1=length(Index1);
% Obtain only non-zero values for 2nd given turn:
Index2=find(ML(:,4,(p+number),q,r)<0); v2=length(Index2);

%Eliminate non-existance v and altitude flight regime data
        if ((r==3) & (q==1));
            psi1=0;
        else
```

```
            psi1=ML(v1,4,p,q,r);
        end
%
%Compute Rotation Matrix using values from final state, first manuever
 rotation1=[cos(psi1),sin(psi1);-sin(psi1),cos(psi1)];
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
v3=v2+v1; v4=v3; man1(1:v1,:)=(ML(1:v1,1:4,p,q,r));
man2(1:v2,:)=(ML(1:v2,1:4,(p+number),q,r));
man3(1:v1,:)=(ML(1:v1,1:4,p,q,r));
man4(1:v2,:)=(ML(1:v2,1:4,(p+number),q,r));
%Compute Second Turn State Vector
        for i=1:v2;
            man1(v1+i,:)=[((rotation1*(man2(i,1:2))')'+ ML(v1,1:2,p,q,r))
            ((ML(v1,3:4,p,q,r)-ML(1,3:4,p,q,r))+man2(i,3:4)) ];
        end
        if ((r==3) & (q==1));
            man(:,:,p,q,r)=zeros;
        else
        dt=ML(v2,5,p,q,r)+ML(v1,5,p+number,q,r);
        delta_vector=[man1(v3,1);man1(v3,2);(man1(v3,3)-man1(1,3));
        (man1(v3,4)-man1(1,4));dt];
        x_vector=zeros(v4,1);
        maneuver_temp=zeros(v4,5);
        x_vector(1:5,:)=delta_vector;
        maneuver_temp(:,:)=[man1(:,:) x_vector];
 man(1:v3,:,p,q,r)=maneuver_temp(:,:);
        end
clear man1 clear  maneuver_temp psi1 Index1 Index2 delta_vector dt
x_vector man2 v1 v2 v3 v4
        end
    end
end clear ML end toc time=toc/60
```

B.2.2 Computing the Advanced Go-Around Manuever.

```
% Captain Alexander M.G. Walan: GAE-03M-09
% Written January 2003
% Thesis Code: advanced_goaround_man
% This code creates the manuever library for a series of go-around manuevers.
% For basic go around, set segment of straight and level flight equal to zero.
% Library is a array of form: S=[i,x,p,q,r,s] where
% S=[index,X,turn number,vel, altitude,wind condition]
% X=[East,North,Down,psi,Time,delta]
clear tic man(:,:,:,:,:,:)=zeros(1000,5,12,3,3,3);
% Decide Which Basic Turn Type To Use%%%%%%%%%%%%
for tp=1:1
    if tp==1
        load ML_4;
        ML=ML_4;
    elseif tp==2
        load ML_5;
        ML=ML_5;
    elseif tp==3
        load ML_6;
        ML=ML_6;
    end
    load ML_1
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Define # of Turns and Altitude Blocks to Use:
%Note, there is no data for v=500 ft/s and Alti=30,000 due to
%flight envelope restrictions.
number=12 for q=1:3
    for r=2:3
        for p=1:12
% Obtain only non-zero values for given turn:
Index1=find(ML(:,4,p,q,r)>0); v1=length(Index1);
%Eliminate non-existance v and altitude flight regime data
            if ((r==3) & (q==1));
              psi1=0;
```

```
            else
               psi1=ML(v1,4,p,q,r);
             end
%Compute Rotation Matrix using values from final state, first manuever
 rotation1=[cos(psi1),sin(psi1);-sin(psi1),cos(psi1)];
% Obtain only non-zero values for 2nd given turn:
Index2=find(ML(:,4,(p+number),q,r)<0); v2=length(Index2);
% Obtain values for straight flight for 2 nautical miles:
Index3=find(ML_1(:,2,1,q,r)<(6076*1.25)); sv=length(Index3);
sv=200;
% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
v3=v2+v1; v4=v3+sv; v5=v4+v1; v6=v5+v1;
man1(1:v1,:)=(ML(1:v1,1:4,p,q,r));
man2(1:v2,:)=(ML(1:v2,1:4,(p+number),q,r));
man3(1:v1,:)=(ML(1:v1,1:4,p,q,r));
man4(1:v2,:)=(ML(1:v2,1:4,(p+number),q,r));
man5=ML_1(1:sv,1:4,1,q,r);
    %   Compute Rotation Matrix using values from final state, first manuever
            if ((r==3) & (q==1));
                 psi2=0;
                mant(:,:,p,q,r)=zeros;
            else
                psi2=man2(v2,4);
                rotation2=[cos(psi2),sin(psi2);-sin(psi2),cos(psi2)];
            end
   %
            for i=1:v1;
             %Compute 2nd Turn State Vector
                man1(v1+i,:)=[((rotation1*(man2(i,1:2))')'+ ML(v1,1:2,p,q,r))
                    ((ML(v1,3:4,p,q,r)-ML(1,3:4,p,q,r))+man2(i,3:4)) ];
            end
         % Add Straight and Level Flight Portion
             for n=1:sv
                 man1(v3+n,:)=[man1(v3,:)+man5(n,:)];
```

```
            end
          %Compute 3rd Turn State Vector
            for l=1:v1
                 man1(v4+l,:)=[man1(v4,:)+man4(l,:)];
            end
           %Compute 4th Turn State Vector
            for m=1:v1;
                man1(v5+m,:)=[((rotation2*(man1(m,1:2))')'+ man1(v5,1:2))
                    ((man1(v5,3:4)+ man1(m,3:4)))];
            end
        dt=2*ML(v2,5,p,q,r)+2*ML(v1,5,p+number,q,r)+ML_1(sv,5,1,q,r);
        delta_vector=[man1(v6,1);man1(v6,2);(man1(v6,3)-man1(1,3));
            (man1(v6,4)-man1(1,4));dt];
        x_vector=zeros(v6,1);
        maneuver_temp=zeros(v6,5);
        x_vector(1:5,:)=delta_vector;
        maneuver_temp(:,:)=[man1(:,:) x_vector];
mant(1:v6,:,p,q,r)=maneuver_temp(:,:); clear man1  maneuaver_temp
psi1 Index1 Index2 delta_vector dt x_vector man2n psi2 clear
Index3 man5 man2 man3 man4
        man_a(1:v6,:,p,q,r,tp)=mant(1:v6,:,p,q,r);
clear v1 v2 v3 v4 v5 v6
          end
      end
    end
    clear ML mant
end toc time=toc/60
```

Appendix C. Matlab M Files Used To Plot Data and Maneuvers

C.1 Files Written To Plot Basic Maneuvers

C.1.1 Plotting Basic Turns.

```
% Captain Alexander M.G. Walan: GAE-03M-09
% Written January 2003
% Thesis Code: plotting_basic_turns
% This code plots the basic turns generated and stored in the various basic
% manuever librarys.
% Library is a array of form: S=[i,x,p,q,r,s] where
% S=[index,X,turn number,vel, altitude,wind condition]
% X=[East,North,Down,psi,Time,delta]
clear
% Load each Library for each basic turn type.
  load ML_4
  load ML_5
  load ML_6
   ML_t=ML_4;
   ML_t2=ML_5;
   ML_t3=ML_6;
clf figure
% Set number of turns to be plotted as well as flight regime (alt, velocity, wind)
for p=2:1:12;
    for q=2:2;
        for r=2:2;
            for s=1:1;
% Find only non-zero entries to plot:
x=1;y=2;mi=1/6760; index=find(ML_t(:,4,p,q,r)>0);
n=length(index)+1; index2=find(ML_t2(:,4,p,q,r,s)>0);
n2=length(index2)+1; index3=find(ML_t3(:,4,p,q,r,s)>0);
n3=length(index3)+1;
```

```
%Plot Figures
plot(ML_t(1:n,x,p,q,r)*mi,ML_t(1:n,y,p,q,r)*mi,'r'); hold on;axis
equal plot(ML_t2(1:n2,x,p,q,r,s)*mi,ML_t2(1:n2,y,p,q,r,s)*mi)
plot(ML_t3(1:n3,x,p,q,r,s),ML_t3(1:n2,y,p,q,r,s),'+','color',[.3*p*(.4*q)*(.1*r)*(6*s)
.1*p*(.4*q)*(.1*r)*(2*s) .7*p*(.2*q)*(.1*r)*(2*s)])
            end
        end
    end
end
```

C.1.2 Plotting Offset Manuever.

```
% Captain Alexander M.G. Walan: GAE-03M-09
% Written January 2003
% Thesis Code: plotting_offset
% This code plots the offset manuever.
% Library is a array of form: S=[i,x,p,q,r,s] where
% S=[index,X,turn number,vel, altitude,wind condition]
% X=[East,North,Down,psi,Time,delta]
% Load Manuever Library containing Offset Maneuver
load Maneuver_Library_offset; man=Manuever_Library_offset;
%Conversion mi converts values from feet to NM.
mi=1/6076 ; Plot Offsets
   figure
for p=2:2:12;
    for q=2:2
        for r=2:2
index1=find(man(:,1,p,q,r)>0); n1=length(index1); st=((n1/2)-1)/2
 plot(man(1:n1,1,p,q,r)*mi,man(1:n1,2,p,q,r)*mi,'.');hold on;axis equal;
% plot(man(1:st,1,p,q,r)*mi,man(1:st,2,p,q,r)*mi,'.');hold on;axis equal
% plot(man((st+1:st*2),1,p,q,r)*mi,man((st+1:st*2),2,p,q,r)*mi,'r.');
% plot(man((st*2+1:st*3),1,p,q,r)*mi,man((st*2+1:st*3),2,p,q,r)*mi,'g.');
% plot(man((st*3+1:st*4),1,p,q,r)*mi,man((st*3+1:st*4),2,p,q,r)*mi,'y.');
xlabel('Easting (Nautical Miles)');ylabel('Northing (Nautical
Miles)'); title('Off-Set Manuever: 1.5G Level Turn @ 10,000 ft');
```

```
legend('.5 Mach','.75 Mach','1.25 Mach');
        end
    end
end
```

Appendix D. Notional Route

D.1 Mission Planning Process for SEAD Mission

PFPS version 3.2 was used to plan the notional SEAD route of Section 4.3. This route was developed by combining the SEAD route of the SAB UCAV study [21] with pre-existing route information from PFPS. Waypoints were edited using the Falcon View default GUI, speed and altitude were specified using the Combat Flight Planning Software (CFPS) menu. No actual weapon delivery was planned on the target, waypoint 7, so the Combat Weapon Delivery System (CWDS) was not used in this scenario. The following route properties were developed:

Threat Layout: Sariavio Notional Threat Laydown was modified to limit type and number of threats around target area.

Route: Nellis AFB-Based notional route was modified to match distance of SAB SEAD route [21].

Segment 3-4: Segment between waypoints 3 and 4 was purposely made to be 30 NM long, bearing true north to make implementation of MBC simpler.

Figure D.1 shows the notional SEAD route as displayed by the PFPS GUI.

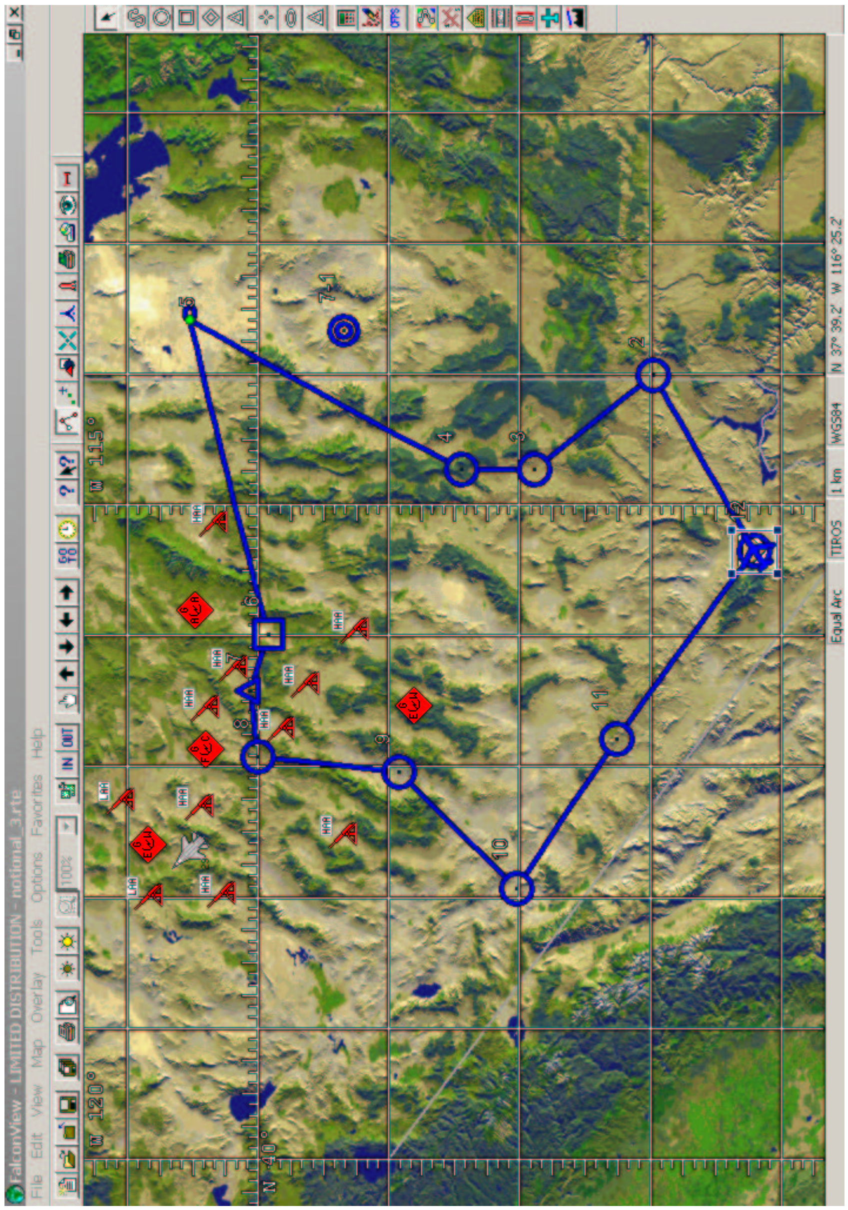

Figure D.1 PFPS Graphical User Interface (GUI)

D.3 *PFPS Output: AF Form 70*

Figures D.2-D.3 contain the AF Form 70 for the notional SEAD route. The AF Form 70, or "kneeboard" charts contain the detailed route information for each waypoint making up the route.

RouteName:C:\PFPS\data\Routes\notional_3.rte Date: 19 FEB 03 NAVDATE: 13 JUN 02

CLEARANCE	TAKE-OFF, CLIMB, CRUISE DATA GENERIC_AIRCRAFT Climb: 10000M Cruise: 445 Wind: Wind: Temp: -5C FF: 1000
FREQUENCIES	FOB: N/C RES: N/C ROUTE AVG WIND: 0

DEP FIELD DATA	TOT DIST	TOT ETE	TOT FUEL
	772.9	02+22+29	10000

TP# DTD# KIND ALT	FIX/PT ID DESCRIPTION (ADD PT ID) (DESCRIPTION) WIND FACTOR	NAV CHAN FREQ	LAT LON VAR ELEV	MH MC (MH) (TH)	DIST LEG TOT	CAS GS TAS IMN	ETE ETA (DVT FF)	FUEL: LEG USED TOT REMG CONT.FUEL (FF)
TP 1 DTD STTO 0M	0		N 36 12.24 W115 20.69 13.4E unk	027 027	0.0 0.0		00+00+00 00:00:00	200 9800 3373 ( )
TP DTD LVLO 10000M	.level off 0		N 36 46.30 W114 21.92 13.3E unk	041 041	58.3 58.3	N/A N/A N/A N/A	00+10+00 00:10:00	167 9633 3206 (1000)
TP 2 DTD TURN 10000M	0		N 36 58.48 W114 00.49 13.2E unk	041 041	21.1 79.4	389 445 445 .70	00+02+50 00:12:50	47 9586 3159 (1000)
TP 3 DTD TURN 10000M	0		N 37 53.22 W114 43.70 13.6E unk	315 315	64.6 144.0	389 445 445 .70	00+08+43 00:21:33	145 9441 3014 (1000)
TP 4 DTD TURN 10000M	0		N 38 26.70 W114 43.70 13.7E unk	346 346	33.4 177.4	389 445 445 .70	00+04+31 00:26:04	75 9366 2939 (1000)
TP 5 DTD 25 ORBT 10000M	0		N 40 31.44 W113 35.12 13.9E unk	009 009	135.5 312.9	389 445 445 .70	00+18+16 00:44:20	304 9062 2635 (1000)
TP 6 DTD IP 10000M	0		N 39 55.44 W115 59.30 14.4E unk	239 239	116.1 429.1	389 445 445 .70	00+45+40 01:30:00	761 8301 1874 (1000)

AF FORM 70 (Modified 11/02/2001, Wind Factors added) - CFPS Ver. 3.2

Figure D.2 AF FORM 70 (Front)

TP# DTD# KIND ALT	FIX/PT ID DESCRIPTION (ADD PT ID) (DESCRIPTION) WIND FACTOR	NAV CHAN FREQ	LAT LON VAR ELEV	MH MC (MH) (TH)	DIST LEG TOT	CAS GS TAS IMN	ETE ETA (DVT FF)	FUEL: LEG USED TOT REMG CONT.FUEL (FF)
TP 7 DTD TGT 10000M	0		N 40 03.57 W116 24.95 14.6E unk	278 278	21.3 450.4	389 445 445 .70	00+02+52 01:32:52	48 8253 1826 (1000)
DVT 20000M	( ) ( ) 0		N 39 20.96 W113 39.79 13.6E unk	(093) (108)	134.4	222 300 300T .49 NM	00+25+12 (1000)	420 7833
TP 8 DTD TURN 10000M	0		N 40 00.33 W116 55.59 14.7E unk	248 248	23.8 474.1	389 445 445 .70	00+03+12 01:36:04	53 8200 1773 (1000)
TP 9 DTD TURN 10000M	0		N 38 55.34 W117 02.21 14.5E unk	170 170	65.1 539.3	389 445 445 .70	00+08+47 01:44:51	146 8054 1627 (1000)
TP 10 DTD TURN 10000M	0		N 38 01.50 W117 55.94 14.4E unk	204 204	68.4 607.6	389 445 445 .70	00+09+13 01:54:04	154 7900 1473 (1000)
TP 11 DTD TURN 10000M	0		N 37 15.47 W116 47.14 14.0E unk	115 115	71.4 679.0	389 445 445 .70	00+09+38 02:03:42	160 7740 1313 (1000)
TP 12 DTD TURN 10000M	0		N 36 12.24 W115 20.69 13.4E unk	118 118	93.9 772.9	260 300 300 .47	00+18+47 02:22:29	313 7427 1000 (1000)

AF FORM 70 (Modified 11/02/2001, Wind Factors added) - CFPS Ver. 3.2

Figure D.3 AF FORM 70 (Back)

Appendix E. Data

E.1 Winds Aloft Data

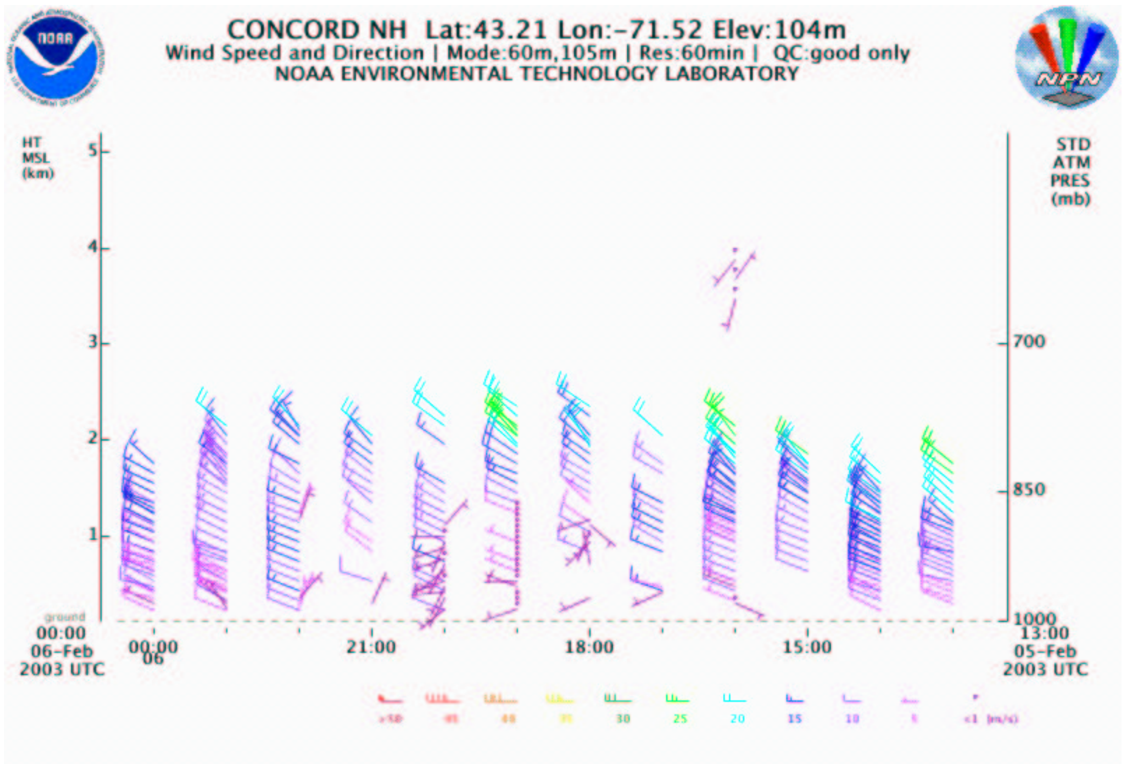

Figure E.1 National Oceanographic Atmospheric Organization Wind Data [19]

E.2 Atmospheric Data

[19]

Altitude Feet	P/Po Pressure Ratio	P/Po Density Ratio	Speed Of Sound Ratio	Speed of Sound	.5 Mach (ft/s)	.75 Mach (ft/s)	1.2 Mach (ft/s)	500 ft/s Mach Number	750 ft/s Mach Number	1250 ft/s Mach Number
0	1.00	1.00	1.00	1063.00	531.50	797.25	1275.60	0.47	0.71	1.20
1000	0.96	0.97	1.00	1059.39	529.69	794.54	1271.26	0.47	0.71	1.20
2000	0.93	0.94	0.99	1055.67	527.83	791.75	1266.80	0.47	0.71	1.21
3000	0.90	0.92	0.99	1051.94	525.97	788.96	1262.33	0.48	0.71	1.21
4000	0.86	0.89	0.99	1048.33	524.17	786.25	1258.00	0.48	0.72	1.22
5000	0.83	0.86	0.98	1044.61	522.31	783.46	1253.53	0.48	0.72	1.22
6000	0.80	0.84	0.98	1040.89	520.44	780.67	1249.07	0.48	0.72	1.22
7000	0.77	0.81	0.98	1037.06	518.53	777.80	1244.48	0.48	0.72	1.23
8000	0.74	0.79	0.97	1033.34	516.67	775.01	1240.01	0.48	0.73	1.23
9000	0.71	0.76	0.97	1029.62	514.81	772.22	1235.55	0.49	0.73	1.24
10000	0.69	0.74	0.97	1025.80	512.90	769.35	1230.95	0.49	0.73	1.24
11000	0.66	0.72	0.96	1021.97	510.98	766.48	1226.36	0.49	0.73	1.25
12000	0.64	0.69	0.96	1018.25	509.12	763.69	1221.90	0.49	0.74	1.25
13000	0.61	0.67	0.95	1014.42	507.21	760.82	1217.31	0.49	0.74	1.26
14000	0.59	0.65	0.95	1010.49	505.24	757.87	1212.59	0.49	0.74	1.26
15000	0.56	0.63	0.94	999.96	499.98	749.97	1199.96	0.50	0.75	1.28
16000	0.54	0.61	0.94	1002.83	501.42	752.13	1203.40	0.50	0.75	1.27
17000	0.52	0.59	0.94	998.90	499.45	749.18	1198.68	0.50	0.75	1.28
18000	0.50	0.57	0.94	995.07	497.54	746.31	1194.09	0.50	0.75	1.28
19000	0.48	0.55	0.93	991.14	495.57	743.36	1189.37	0.50	0.76	1.29
20000	0.46	0.53	0.93	987.21	493.60	740.41	1184.65	0.51	0.76	1.29
21000	0.44	0.52	0.93	983.28	491.64	737.46	1179.93	0.51	0.76	1.30
22000	0.42	0.50	0.92	979.34	489.67	734.51	1175.21	0.51	0.77	1.30
23000	0.40	0.48	0.92	975.30	487.65	731.48	1170.36	0.51	0.77	1.31
24000	0.39	0.46	0.91	971.37	485.68	728.53	1165.64	0.51	0.77	1.31
25000	0.37	0.45	0.91	967.33	483.67	725.50	1160.80	0.52	0.78	1.32
26000	0.36	0.43	0.91	963.29	481.65	722.47	1155.95	0.52	0.78	1.32
27000	0.34	0.42	0.90	959.25	479.63	719.44	1151.10	0.52	0.78	1.33
28000	0.33	0.40	0.90	955.21	477.61	716.41	1146.25	0.52	0.79	1.33
29000	0.31	0.39	0.89	951.17	475.59	713.38	1141.41	0.53	0.79	1.34
30000	0.30	0.37	0.89	947.03	473.51	710.27	1136.43	0.53	0.79	1.35
31000	0.28	0.36	0.89	942.88	471.44	707.16	1131.46	0.53	0.80	1.35
32000	0.27	0.35	0.88	938.84	469.42	704.13	1126.61	0.53	0.80	1.36
33000	0.26	0.33	0.88	934.70	467.35	701.02	1121.64	0.53	0.80	1.36
34000	0.25	0.32	0.88	930.55	465.28	697.91	1116.66	0.54	0.81	1.37
35000	0.24	0.31	0.87	926.30	463.15	694.72	1111.56	0.54	0.81	1.38
36000	0.22	0.30	0.87	922.15	461.08	691.61	1106.58	0.54	0.81	1.38
37000	0.21	0.28	0.87	921.73	460.86	691.30	1106.07	0.54	0.81	1.38
38000	0.20	0.27	0.87	921.73	460.86	691.30	1106.07	0.54	0.81	1.38
39000	0.19	0.26	0.87	921.73	460.86	691.30	1106.07	0.54	0.81	1.38
40000	0.19	0.25	0.87	921.73	460.86	691.30	1106.07	0.54	0.81	1.38

Figure E.2 Atmospheric Data [19]

Vita

Captain Alexander M.G. Walan graduated from Oswego High School in Oswego, Illinois in June 1990. He entered undergraduate studies at the University of Minnesota, Minneapolis-Saint Paul, Minnesota where he graduated with a Bachelor of Science Degree in Aerospace Engineering and Mechanics in December 1994.

His first assignment was as a intelligence analyst engineer at the National Air Intelligence Center (NAIC), Wright-Patterson AFB, Ohio. While assigned to NAIC he served as a Scientific and Technical Intelligence Liaison Officer (STILO) for the Green Flag 96 exercise and as a white cell scripter during Green Flag 97. He authored numerous intelligence studies including “Future Land Attack Cruise Missiles.”

In March 1998 he was re-assigned to the Electronic System Center, Hanscom AFB, Massachusetts. He held several positions while there including: Chief Engineer Low Observable Mission Planning, AOC Blue Two Development Lead, JEFX ’00 Site Lead, Langley AFB. He was awarded a Masters of Business Administration from The University of Massachusetts, Lowell in July 2001.

In August 2001, he entered the Graduate School of Engineering and Management, Air Force Institute of Technology. Upon graduation, he will be assigned to the Air Force Research Lab Sensors Directorate, Wright-Patterson AFB, Ohio.

Bibliography

1. Alan Brockett, Lila Laux and David Dahn. "A Human-Centered Approach to Controlling Multiple Semi-Autonomous Unmanned Vehicles." *AIAA 1st Technical Conference and Workshop on Unmanned Aerospace Vehicles, Systems, Technologies, and Operations*. Number AIAA 2002-3472. 2002.

2. Anderson, John D. *Introduction To Flight*. "McGraw-Hill", "1989".

3. Bryson, Arthur E. *Dynamic Optimization*. ?, 1999.

4. C. Nilsson, C. Hall, S. Heinzen and N. Chokani. "GPS Auto-Navigation Design for Unmanned Air Vehicles." *40th AIAA Aerospace Sciences Meeting and Exhibit*. Number AIAA 2002-0389. 2002.

5. Duus, Lt Col Glen (Ret). "The History Of UAVs: Vehicles Without Pilots." Australian Army Aviation Association Web Site: http://www.fourays.org, January 2003.

6. ESC/ACU. "Recce Mission Planning Status." Electronic Powerpoint Presentation, 2002.

7. Federation Of American Scientists. "US Weapon Systems." FAS Website: http://www.fas.org, January 2003.

8. Flade, Lieutenant Colonel John W. "Teaching A New Dog Old Tricks: Replacing Man With Artificial Intelligence In Combat Aircraft," *Strategy Research Project, USAWC Class of 2000. U.S. Army War College, Carlisle Barracks, PA*

9. Frazzoli, Emilo. "Maneuver-Based Motion Planning and Coordination for Single and Multiple UAV's." *AIAA 1st Technical Conference and Workshop on Unmanned Aerospace Vehicles, Systems, Technologies, and Operations*. Number AIAA 2002-3472. 2002.

10. Glassco, Captain Roy Glen. *Closely Supervised Reactive Control of an Uninhabited Aerial Vehicle*. MS thesis, AFIT/GAE.ENY/99J-01, Graduate School of Engineering, Air Force Institute of Technology (AETC), Wright-Patterson AFB OH, June 1999 (AD-A364819).

11. Howitt, S.L. "Real Time Deep Strike Mission Simulation Using Air-Launched UAV's." *AIAA 1st Technical Conference and Workshop on Unmanned Aerospace Vehicles, Systems, Technologies, and Operations*. Number AIAA 2002-3480. 2002.

12. Jacques, Lieutenant Colonel David. Class handouts distributed in MECH 622, Functional Optimization and Optimum Control. Graduate School of Engineering, Air Force Institute of Technology (AETC), Wright-Patterson AFB OH, January 2003.

13. Lazarski, Lieutenant Colonel Anthony J. "Leagle Issues with UAV's," *Aerospace Power Journal* (April 2002).

14. Liebst, Bradley S. *Nonlinear Dynamical Model of the F-16 Aircraft*. Unpublished Working Paper CRDA 96-AFIT-02, Wright-Patterson AFB, OH: Air Force Institute Of Technology, 2000.

15. Loegering, Greg. "Landing Dispersion Results- Global Hawk Auto-Land System." *AIAA 1st Technical Conference and Workshop on Unmanned Aerospace Vehicles, Systems, Technologies, and Operations*. Number AIAA 2002-3457. 2002.

16. Maj. Chip Thompson, USAF. "F-16 UCAVs: A Bridge to the Future of Air Combat?," *Aerospace Power Journal* (Spring 2000).

17. Miller, Captain Garry D. *Trimmed Flight Control*. AFIT Working Paper, Wright-Patterson AFB, OH: Air Force Institute Of Technology, 2002.

18. Niculescu, Marius. "Lateral Track Control Law for Aersonde UAV." *39th AIAA Aerospace Sciences Meeting and Exhibit*. Number AIAA 2001-0016. 2002.

19. NOAA Forecast Systems Laboratory. "NOAA Forecast Systems Laboratory Site: http://www.profiler.noaa.gov/jsp." Web Site, January 2003.

20. North Atlantic Treat Organization: Research and Technology Organization. *Alternative Control Technologies Human Factors Issues*. NATO RTO Lecture Series 215 DTIC AD-A355911. Neuilly-Sur-Seine Cendex, France: Research and Technology Office, 1998.

21. P Worch, J. Borky, R Gabriel W. Heiser T. Swalm T. Wong. "UAV Technologies and Combat Operations, Volume 1," *Air Force Scientific Advisory Board*, (SAB-TR-96-01) (1996).

22. Pendleton, Captain Ryan R. *Use of Unusual Aircraft Orientations to Generate Low Observable Routes*. MS thesis, AFIT/GAE.ENY/00M-09, Graduate School of Engineering, Air Force Institute of Technology (AETC), Wright-Patterson AFB OH, March 2000.

23. Pope, Alexander. "Essay On Man." Penn State University's Electronic Classics Series Web Site: http://www.hn.psu.edu/faculty/jmanis/pope.htm, Febuary 2003.

24. Prichard, David E.R. *Dynamic Route Replanning and Retasking of Unmanned Aerial Reconnaissance Vehicles*. MS thesis, AFIT/GAE.ENY/00M-10, Graduate School of Engineering, Air Force Institute of Technology (AETC), Wright-Patterson AFB OH, March 2000 (AD-A382039).

25. Sheridan, Thomas B. *Telerobotics, Automation, and Human Supervisory Control*. MIT Press, 1992.

26. Siouris, George M. *Aerospace Avionics Systems, A Modern Synthesis*. San Diego: Academic Press, Inc, 1993.

27. Tobin, Major Keith E. *Piloting the USAF's UAV Fleet: Pilots, Non-Rated Officers, Enlisted, or Contractors?*. MS thesis, School of Advanced Airpower Studies: Air University, Maxwell Air Force Base, Alabama, June 1999 (AD-A391753).

28. U.S. Navy Hydrographic Office. *Air Navigation*. H.O. Pub. 216. Washington D.C.: U.S. Government Printing Office, 1955.

29. Wolper, James S. *Understanding Mathematics For Aircraft Navigation*. McGraw-Hill, 2001.

REPORT DOCUMENTATION PAGE

Form Approved OMB No. 0704–0188

1. REPORT DATE *(DD–MM–YYYY)*
14–03–2003

2. REPORT TYPE
Master's Thesis

3. DATES COVERED *(From — To)*
Sep 2001 — Mar 2003

4. TITLE AND SUBTITLE
APPLICATION OF MANEUVER-BASED CONTROL IN VARIABLE AUTONOMY UNMANNED COMBAT AERIAL VEHICLES

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)
WALAN, ALEXANDER M.G., CAPTAIN, USAF

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Air Force Institute of Technology
Graduate School of Engineering and Management
2950 P Street, Building 640
WPAFB OH 45433-7765

8. PERFORMING ORGANIZATION REPORT NUMBER
AFIT/GAE/ENY/03M-09

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)
DAGSI

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION / AVAILABILITY STATEMENT
Approval for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES

14. ABSTRACT
The rise in the capability and lethality of unmanned combat aerial vehicles (UCAVs) historically has been paralleled by an increase in the complexity in the command and control of these systems. As the missions tasked to these vehicles increase in complexity and lethality, operators will increasingly require the ability to tailor the amount of control exercised over the vehicle. Maneuver Based Control (MBC) offers the potential to give future UCAV operators the ability to vary the autonomy of the vehicle against the amount of control they exercise. The objective of this research is to validate the concept of Maneuver Based Control (MBC). Utilizing an existing non-linear computer model for an F-16 aircraft, maneuvers representative of those performed in a real-world mission are computed and stored. These stored maneuvers are then used to illustrate the application of MBC to in-flight replanning and mission execution by way of a representative mission scenario. Results indicate that MBC provides an effective method of variable control for future UCAVs.

15. SUBJECT TERMS
UAV, UCAV, DRONE, RPV, MANUEVER, MISSION PLANNING, AUTONOMOUS CONTROL, VARIABLE AUTONOMY, UNMANNED AERIAL VEHICLE

16. SECURITY CLASSIFICATION OF: a. REPORT	b. ABSTRACT	c. THIS PAGE	17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
U	U	U	UU	112	Curtis S. Spenny, PhD (ENY)

19b. TELEPHONE NUMBER *(include area code)*
(937) 255-7777, x3296

Standard Form 298 (Rev. 8–98)
Prescribed by ANSI Std. Z39.18