Université d'Ottawa • University of Ottawa

# Université d'Ottawa · University of Ottawa

FACULTÉ DES ÉTUDES SUPÉRIEURES ET POSTDOCTORALES

FACULTY OF GRADUATE AND POSTDOCTORAL STUDIES

CHENG, Zhaoquan

AUTEUR DE LA THÈSE - AUTHOR OF THESIS

M.A.Sc. (Mechanical Engineering)

GRADE - DEGREE

Department of Mechanical Engineering

FACULTÉ, ÉCOLE, DÉPARTEMENT - FACULTY, SCHOOL, DEPARTMENT

TITRE DE LA THÈSE - TITLE OF THE THESIS

Dynamic Inversion and Model Predictive Control for Unmanned Aerial Vehicles

Dan Necsulescu

DIRECTEUR DE LA THÈSE - THESIS SUPERVISOR

EXAMINATEURS DE LA THÈSE - THESIS EXAMINERS

J. Sasiadek

B. Kim

M. Liang

J.-M. De Koninck, Ph.D.

LE DOYEN DE LA FACULTÉ DES ÉTUDES SUPÉRIEURES ET POSTDOCTORALES

SIGNATURE

DEAN OF THE FACULTY OF GRADUATE AND POSTDOCTORAL STUDIES

# Dynamic Inversion and Model Predictive Control for Unmanned Aerial Vehicles

**Zhaoquan Cheng**

Thesis submitted to the
Faculty of Graduate and Postdoctoral Studies
in partial fulfillment of the requirements for the degree of

**Master of Applied Science**

**in**

**Mechanical Engineering**

Department of Mechanical Engineering
Faculty of Engineering
Ottawa-Carleton Institute of Mechanical and Aerospace Engineering
University of Ottawa

**March, 2004**

*Your file Votre référence*
*ISBN: 0-494-01438-5*
*Our file Notre référence*
*ISBN: 0-494-01438-5*

*To my lovely wife Liping*

*and my wonderful parents*

# Abstract

Unmanned Aerial Vehicles (UAVs) have attracted considerable interest in the commercial markets for the military and civilian uses, such as surveillance and reconnaissance, aerial surveys for natural sources, traffic monitoring, and early forest fire detection, etc. Presently, UAVs are being proven as a cost-effective platform for the military and civilian applications because they gather information without endangering the lives of the pilots, increase maneuverability without limitations from human abilities, cost much less than the traditional aircrafts and do not need human-pilot interfaces. Although UAVs present numerous advantages over the manned aircrafts, they face challenges in achieving autonomous control. Model Predictive Control (MPC) is an interesting solution for UAV control to improve the level of autonomous control, but is mostly applicable to a linear or linearized system at this time.

In this thesis, first is presented a complete kinematics and dynamic model of Unmanned Aerial Vehicles which is programmed in Simulink. Second, a control scheme based on a proportional controller for the inner loop and a proportional – integral controller for the outer loop is investigated. Model predictive control is applied to a linearized UAV using dynamic inversion, and simulation results obtained using a nonlinear UAV model are presented and analyzed in view of the suitability for UAV autonomous control.

## Acknowledgment

I would like to express my deeply gratitude to my supervisor, Dr. Dan S. Necsulescu for his time, keen insight, encouragement, support, guidance and patience. I deeply appreciate his sincere enthusiasm in research which will guide me throughout my career.

I am also grateful to my master thesis defence committee members, for their valuable time, guidance, review criticisms, and insightful suggestions, regarding my thesis.

Special thanks to my colleagues for their technical expertise, guidance, and moral support throughout my research.

Most of all, I am grateful to my wife Liping particularly, whose continuous supports, understanding, and patience throughout my master studies have enabled me to produce this thesis.

# TABLE OF CONTENTS

# LIST OF FIGURES

*Number* *Page*

# Nomenclature

## Acronyms and Abbreviations

| | |
|---|---|
| c.g. | Center of Gravity |
| AOA | Angle of Attack |
| DCM | Direct Cosine Matrix |
| DMC | Dynamic Matrix Control |
| GCS | Ground Control Station |
| HARV | High Angle-of-Attack Research Vehicle |
| LPV | Linear Parameter Varying |
| MAC | Mean Aerodynamic Chord |
| MPC | Model Predictive Control |
| UAV | Unmanned Aerial Vehicle |
| VRML | Virtual Reality Modeling Language |
| VTOL | Vertical TakeOff and Landing |

## Symbols

| | |
|---|---|
| $b$ | Wing Span |
| $c$ | Mean Aerodynamic Chord |
| $\rho$ | Dynamic Pressure |
| $S$ | Wing Planform Area |
| $C_d \quad C_y \quad C_l$ | Drag / Side / Lift Force Coefficient |
| $C_L \quad C_M \quad C_N$ | Roll / Pitch / Yaw Moment Coefficient |
| $C_{l0}$ | Lift at Minimum Drag |
| $C_{d0}$ | Minimum Drag coefficient |
| $C_{d\ y\ l}^{\delta_{control\ surface}}$ | Derivative of Drag / Side / Life Force w.r.t Control Deflection |
| $\delta_f \quad \delta_e \quad \delta_a \quad \delta_r$ | Control Deflection of Flap / Elevator / Aileron / Rudder |
| $C_{d\ y\ l}^{\alpha\ \beta}$ | Derivative of Drag / Side / Life Force w.r.t AOA / Sideslip Angle |

| | |
|---|---|
| $C_{L\ M\ N}^{\delta_f\ \ e\ \ a\ \ r}$ | Derivative of Roll / Pitch / Yaw Moment w.r.t Control Deflection |
| $F_T$ | Thrust generated by propulsion system |
| $C_{th}$ | Thrust Coefficient |
| $\delta_{th}$ | Thrust input |
| $g$ | Gravity |
| $I_{x\ \ y\ \ z\ \ xz}$ | Inertia about Aircraft body axis and Cross Product of Inertia |
| $L\quad M\quad N$ | Roll / Pitch / Yaw Moment |
| $p\quad q\quad r$ | Roll / Pitch / Yaw Angle Rate |
| $p_w\quad q_w\quad r_w$ | Roll / Pitch / Yaw Angular Rate in wind axis system |
| $\psi\quad \theta\quad \phi$ | Yaw (Heading) / Pitch / Roll (Bank) Angle |
| $\alpha$ | Angle of Attack |
| $\beta$ | Sideslip Angle |
| $V_T$ | Airspeed |
| $x_b - y_b - z_b$ | Body Axis System |
| $u\quad v\quad w$ | Velocity components in the body axis system |
| $\mathbf{T_{WB}}$ | Transformation matrix from wind axis system to Body axis system |
| $\mathbf{T_{IB}}$ | Transformation matrix from earth fixed frame to Body axis system |
| $x\quad y\quad z$ | UAV position in the earth fixed frame |
| $H_x\quad H_y\quad H_z$ | Moment of momentum along the x, y and z axes |
| $I_{i,\ ij}$ | Inertial Coefficients |
| $\mathbf{K_p}$ | Proportional Gain Matrix |
| $\mathbf{K}_I$ | Integration Gain Matrix |

| | |
|---|---|
| $\mathbf{L_f}$ * | Lie derivative |
| $\mathbf{u_M}(k)$ | Input data going into $M$ steps future starting at $k$ |
| $\mathbf{y_M}(k)$ | Output data going into $M$ steps future starting at $k$ |
| $\mathbf{C_M}$ | Controllability Matrix |
| $\mathbf{O_M}$ | Observability Matrix |
| $\mathbf{T_M}$ | Toeplitz Matrix |
| $\mathbf{W}$ | Future reference |

# Chapter 1

# 1 INTRODUCTION

## 1.1 Motivation

Unmanned Aerial Vehicles (UAVs) have attracted considerable interest in the commercial markets for the military, mainly for surveillance and reconnaissance purposes, and for the civilian uses such as aerial surveys for agriculture, traffic monitoring , pollution control, meteorological data collection, pipeline survey and early forest fire detection, etc. Presently, UAVs are being proven as cost-effective platforms for the military and civilian applications because they gather information without endangering the lives of the pilots, increase maneuverability without limitation due to human capabilities, much lower cost than the traditional aircraft and elimination of on-board human-pilot interfaces. Although UAVs present some advantages over the manned aircrafts, they require autonomous control and have more stringent size and capability constraints than conventional aircrafts. To enhance the safety and security level for performing some tasks, remotely controlled, semi-autonomous and autonomous vehicles are used in practice (Baum M. L. and Passino K. M., 2002). With recent developments in various aspects such as automatic flight control systems, guidance techniques, and navigation instruments, some researches have been done to explore approaches to move the operations from ground stations and to the vehicle and realize complex operational objectives on-board with high reliability in aerial missions, at lower risk and lower cost (Okan, A. et al., 2002). Therefore, complex requirements, especially related to vehicle

dynamics, are imposed for UAV controller design. Although guidance systems have been developed for many years, the UAVs can't achieve every stage of the whole mission under complex environment conditions without some common participation from ground based control station.

In most configurations, UAVs are designed to operate in two distinct modes: remote control from Ground Control Stations (GCS) and autonomous control (Liang F., 2002). In GCS remote control mode, the UAV receives commands from the GCS, and at the same time, the motion and the states of the UAV are transmitted to the GCS. The commands are generated by an experienced pilot or by an expert system. Normally, the ground remote control mode is used to cope with highly uncertain or unpredicted conditions, such as takeoff and landing under complex environments, unpredicted failure of actuators during mission and unanticipated natural disturbances. In the ground remote control mode, UAVs are controlled by experienced pilots. Therefore, the complexity of the controller design and configuration are reduced and the failure of UAVs can be prevented under certain conditions. In the autonomous control mode, UAVs can perform pre-programmed operational missions without participation of pilots, and the states are transmitted to the GCS for monitoring and thus the decision of staying in autonomous control mode or switching to GCS control mode is made.

To deal with different tasks, there is a need for a higher level of autonomous, reliable and performance UAVs and also a more harmonious integration of autonomous UAVs and intelligent pilot system. To enhance the operational capabilities of UAVs, one of the issues to be solved is to make sure that the UAV can follow the trajectory

generated in real-time to finish the mission task taking into consideration the dynamic characteristics of the UAV, and the constraints under which the UAV operates, such as the safe and protected flight envelope, adverse atmospheric conditions, collision avoidance with static ground obstacles and other moving objects in the air to correct the trajectories, efficient fuel consumption for long-distance mission, correction of the predefined trajectories etc. In order to explore the competitive advantages, the UAVs should have a high level performance and maneuverability when operated in a tightly constrained environment. A two-level control system is introduced in Mettler M. B., et al (2002). In this system, the feedback control system is used to achieve various actions, and the on board high-level expert system decides the selections of the combination of actions and path planning to fulfill the mission under different flight conditions. Furthermore, in order to increase the operation efficiency, multiple-UAV systems, as opposed to a single UAV, were developed in recent years to improve the overall mission efficiency and capability. However, multiple-UAV systems require collision avoidance control system between the each UAV in the formation group and with obstacles. The control of each UAV must be based on the flight environment parameters and the states of other members in the formation.

Traditionally, the flight controller is designed in certain low-angle-of-attack flight regimes using linear design methods, which use linearized models in the trim condition. The parameters of these linear controllers are defined as a function of different trim conditions. A linear controller alone can not effectively control the higher performance UAVs in the nonlinear flight regions or with high-angle-of-attack flight. In this thesis, the

dynamics based control approach, dynamic inversion, is used to enhance the piloting ability and performance in the whole flight envelope without modifying controller parameters. Dynamic Inversion permits UAVs to realize the real-time command tracking and Model Predictive Control is used for the inner loop.

## 1.2 Research Goals

There are several areas of research for autonomous UAVs: UAV design and configuration, navigation, motion planning, collision avoidance, target acquisition, automatic flight control and multi-UAV formation hold, etc.

My research focuses on UAV's kinematic and dynamic modeling and dynamic flight control algorithms and model predictive control (MPC) for single UAV. First, this thesis presents a complete six-degree-freedom nonlinear model for the rigid body UAV using the hybrid coordinate systems. Second, the Dynamic Inverse (DI) is accomplished by inner and outer loop control, and the MPC is applied to the inner loop at this stage to investigate the potential advantages to the collision avoidance. Finally, the complete nonlinear UAV model and control algorithms are simulated by Simulink® and the simulation results are compared and analyzed to verify the advantages of dynamic inversion control and the suitability of MPC for UAV autonomous control using dynamic inversion as proposed in this thesis.

# Chapter 2

# 2 LITERATURE REVIEW

In the section, the literature review stresses on the modeling of a single UAV, conventional flight control, dynamic inversion control and model predictive control.

## 2.1 Kinematics and Dynamics Nonlinear Model of UAV

The nonlinear dynamic model of an UAV is based on the Newton-Euler equations of motion for a rigid body (Stevens B. L. and Lewis F. L. (1992), Nelson R. C. (1998)). The equations of motion for UAV, described as a rigid body UAV with six-degree-freedom, driven by thrust, aerodynamic and gravity forces to fly over a flat, non rotating Earth with zero ambient winds are presented by Menon P.K.A. et al (1987) and Looye G. (1999).

In Jang I. J. S. and Tomlin C. J. (2001) and Lee T. and Kim Y. (2001), the motion of a six-degree freedom UAV is described by three translational equations, derived in the wind axes to reduce complexities of describing aerodynamic angles, and three rotational equations, derived in the body axes to avoid a time-varying moment of inertia. Normally, most of UAVs are designed to be axially symmetrical such that the moment generated by the control surface does not affect the direction of flight and the moment equations can be simplified by neglecting effects of the cross-coupling inertial coefficients. In these equations, the aerodynamic forces and moments are defined by the linear functions of aerodynamic angles, angular velocity and control surface deflections. Furthermore, the

position of UAV in the Earth Fixed Frame is defined by three navigation equations and the orientation of UAV is expressed by three kinematics equations with Euler angles in Stevens B. L. and Lewis F. L. (1992).

In Snell S. A. et al (1992) and Bugajski D. J. and Enns D. F. (1992), the full, six-degree-of-freedom (DOF), nonlinear, rigid body dynamics, without considering the dynamics of the sensor, is modeled by 12 first-order differential equations, i.e. 12 states. The states are chosen as body roll rate, body pitch rate, body yaw rate, velocity roll angle, angle of attach sideslip angle velocity, heading angle, flight-path angle and horizontal displacement with regard to the reference frame. The aerodynamic coefficients, needed to calculate the aerodynamic force and moment, are obtained by interpolation from experimental results of a High Angle-of-attack Research Vehicle (HARV). The states are divided into fast and slow dynamics. The fast dynamics are three angular velocities, which are controlled by aileron, canard, rudder, and lateral and normal thrust vectoring control, and they are calculated from the moment equations. The slow dynamics are roll angle about the velocity vector and aerodynamic angles, i.e. angle of attack and sideslip angle. Assuming there is enough time scale separation, the three fast dynamics are calculated from slow dynamics. The attitudes of UAV are controlled by the three angular velocity commands. The aircraft velocity vector is controlled by thrust and attitude commands.

In the Liang F. (2002), Adams R. J. et al (1994), and Aerosim [1], the object orientated design approach is presented. The total nonlinear UAV model consists of many submodels describing the nonlinear motion equations, engines, atmosphere,

aerodynamics, sensors, and actuators. These sub-models are connected one to another in the Simulink through the interconnections according to the way that the objects physically interact. The rigid UAV model is used as the basis while the aerodynamic, elastic parts are defined depending on the accuracy requirement by the applications. Some applications are presented in Looye G. (1999).

In recent year, some advanced UAVs have been proposed in recent years. In the Okan, A. et al. (2002), the Vertical Takeoff and Landing (VTOL) UAV is introduced which combines the vertical flight capability of the helicopter with the superior forward flight performance of the fixed-wing airplane. Due to this combination, these UAVs have their own characteristics and are generally more difficult for pilot to control than either conventional aircrafts or helicopters.

## 2.2 Conventional Control

The first and most important step in the UAV control design is to determine a set of dynamics which can be achieved by the given limited effectors to stabilize UAV under the required moments and forces and to follow the specified trajectory command with considering collision avoidance of unexpected objects.

In the conventional flight control designs, the aircraft dynamics are assumed to be linear and time-invariant about the nominal trimmed flight condition. In Stevens B. L. and Lewis F. L. (1992), Nelson R. C. (1998), the controller design is based on the linearized model to meet handling quality criteria. With the introduction of the highly maneuverable aircraft, which has the capability of flight at high angle of attack, the

Multi-Input and Multi-Output (MIMO) control design methods are required. In some cases, the excellent performance is achieved in a local region where linearity can be assumed, but robustness and stability outside the trimmed flight conditions are not guaranteed. To make the design for the entire operational envelope design, the controller parameters are scheduled as functions of different trimmed flight conditions to cover the entire flight envelop. However, in the extreme flight conditions, the unmodelled effects of nonlinearities lead to deteriorate the performance of these systems, such as for significant high angles of attack or high angular rates pointed (Lane S. H. and Stengel R. F. 1988). Furthermore, gain scheduling of MIMO controller is quite laborious and complicated, and the fidelity of the system based on the linearized model without considering the nonlinear nature of aircraft in the whole flight envelope is very low. Therefore, an alternative design method, which considers the nonlinearity of aircraft, has to be used.

## 2.3 Dynamic Inverse

Over the last decade, nonlinear controller for the UAVs controller design has gained considerable popularity for the designs of the supermaneuverable flight aircrafts which is to execute demanding missions in extreme operational regimes with high angle of attack and high acceleration turns. The dynamic model of aircraft is significantly nonlinear. Compared with conventional designs using linearizing assumptions, nonlinear controllers, in particular dynamics inversion based controllers, gained a great attention due to their potential to improve the levels of performance, maneuverability, robustness, stability and fidelity of system over conventional flight controller designs and have been

used for the different types of modern aircrafts such as F-16, F-18, F-117, X-38 (Ito D. et al, 2001). The idea of dynamic inversion applied to highly maneuverable aircraft operating system is proposed in Lane S. H. and Stengel R. F. (1988). Inverse Dynamics controller is synthesized through nonlinear feedback to eliminate the undesirable dynamics to get specified desired dynamic characteristics. This more general methodology, applicable to input-output system, is Feedback Linearization. After feedback linearization, the dynamics between the desired inputs and controlled outputs is linear over the entire flight domain and the controller's parameters don't need to be adjusted once they are designed for any specified point in the flight envelope. Therefore, the complexity of gain scheduling is reduced or even avoided. To apply the dynamic inversion, Reiner J. et al (1995) pointed out that the key equations to describe the system are known precisely and the states of system have to be measurable or estimated accurately.

In actual applications of dynamic inversion, the equations of the aircraft are never known exactly due to different reasons such as the uncertainty of aerodynamic parameters, unsteady aerodynamic effect at high angle of attach, unmodeled dynamics and changes of dynamics under different formation structures, actuator failure or unpredicted damages. The dynamic inversion controllers will lose their beneficial effects regarding the maneuverability and expected performance if the nonlinear dynamics used for the feedback are not complete and accurate. To deal with the nonlinearity and uncertainty of aerodynamic coefficients, lookup tables are established to simplify the design process (Bugajski D. J. and Enns D. F., 1992). Schumacher C. (1999), combined

with adaptive neural networks with dynamic inversion to compensate the errors of parameters, and applied it to a tailless fighter aircraft. In Singh S. N. and Steinberg M. (1996), derived an adaptive control law on the basis of state feedback linearization model to operate the aircraft under large uncertainty in aerodynamic parameters. In Ito D. et al (2001), a Linear Quadratic Gaussian (LQG) controller is used in the outer loop to improve the robustness when the inner loop presents some uncertainties. In Lee T. and Kim Y. (2001), backstepping controller is used to stabilize the states simultaneously and the adaptive controller based on neural networks is used to compensate the effects of uncertainty in the aerodynamic modeling. In Reiner J. et al (1995), a linear outer-loop controller designed using $\mu$ synthesis improves the robustness to uncertainties in the system modeling and the measurements.

With the developments of the flexible aircrafts, dynamic inversion has become as a valuable candidate to handle the flexibility in the aircraft design. In the paper of Gregory I. M. (1999), the modified dynamic inversion controller has been used in the large flexible aircraft and obtained encouraging results.

## 2.4 Model Predictive Control

Model Predictive Control (MPC), also known as receding horizon control, is not a recent method of control design. In 1976, Richalet first promoted model predictive control for the process control. Model predictive control has been studied in several survey papers. In particular, the survey by Mayne D. Q. et al (2000) is used as a primary source for this part. The standard optimal control problem of model predictive control is

solved in finite horizon and the current state of the plant is determined by solving the optimal control problem on-line, which is different from the conventional control using a pre-computed control law. The major advantage of model predictive control is being an easy and effective design method and being able to yield excellent discrete time tracking. An important characteristic of model predictive control is the ability to consider state and control limits as the constraint during the optimization.

In the paper of Steinbery M. L. (2001) and Heise S. A. and Maciejowski J.M. (1996), the feedback control applied to the plant is obtained by solving an online open-loop optimal control problem to predict future system behavior based on future input changes. An on-line sequence of control commands is obtained based on a optimal quadratic cost function, using the current states as the initial states, over a finite optimization horizon. The difference between the specified setting points in a prediction horizon and system future outputs is minimized taking into account control power penalties and system hard or soft constraints over the control horizon. Normally, the optimization horizon and control horizon are the same and only the first control value of command sequence is implemented. At the next sample time, a new entire set of command sequence is re-computed.

A large amount of time is needed to get the solution of the optimization problem recast as a set of difference equations. Successful applications of model predictive control were confined so far mostly to the area of chemical process control. With the development of faster and cheaper computers and the development of efficient optimization solving numerical algorithms, the model predictive control has become very

popular recently. However, the stability of model predictive control can't be proved for the nonlinear system theoretically, and is obtained only by restricting the stable systems to a finite domain and by choosing a larger horizon compared with setting time of systems (Mayne D. Q. et al, 2000). The stability and successful applications for the linear systems are presented in Soeterboek, R. (1992), Clarke D. (1994) and Maciejowski J. M. (2002).

In Jung Y. C. and Hess R. A. (1991), single-input /single-output generalized predictive control is applied for a rotorcraft to control the longitudinal / vertical terrain-following flight. In the paper of Berlin F. and Frank P.M. (1994), the proposed predictive control algorithm is applied to MIMO linear system which is a 3-tank system with two inputs and two outputs.

The application of predictive control for the MIMO nonlinear system gains a lot of research attention in resent years. Since the optimal solution for the nonlinear system is not guaranteed, the model predictive control is used in combination with other control methods. In the Mehra R.K. et al (2001), the model predictive controller is applied to the XV-15 to act as a real-time simulator and determine model predictive control for an autopilot using Linear Parameter Varying (LPV) approximation for the nonlinear dynamics. For the linear parameter varying method, the changing of nonlinear system states are related to the time-varying scheduling variable which is defined by the flight condition and thus the whole flight envelope is defined by these states. To get the scheduling variable, firstly, the flight envelope is divided into several regions, and then the trim states, control settings and constraints are obtained from trimmed aircraft

equations at each region and at last, the parameters of linear equations around the trim point are described by the functions of scheduling variable at specified regions. In the Bhattacharya R. et al (2002), the model predictive control is applied for the nonlinear F-16 aircraft with the LPV.

In Botto M. A. and Costa J. S. (1998), the nonlinear system is linearized by the input-output feedback linearization, and then the predictive control is used as the outer loop to handle the constraints for actuator saturation on the basis of linearized system.

In this thesis, inner / outer loop dynamic inversion permits the UAV follow the command trajectory in the whole flight envelope and the combination of model predictive control with inner loop dynamic inversion provides a solution to improve higher level of autonomous control of an UAV.

# Chapter 3

# 3 UAV DYNAMIC MODEL

## 3.1 Brief Introduction of Aircraft Description and Modeling

The main components of a fixed wing aircraft are shown in Figure 3.1. The engine provides the thrust to push the aircraft forward and the wing generates most of the lift to hold the aircraft in the air. The air resistance to the motion results in the aerodynamic drag force. These forces applied to the aircraft are shown in Figure 3.2.

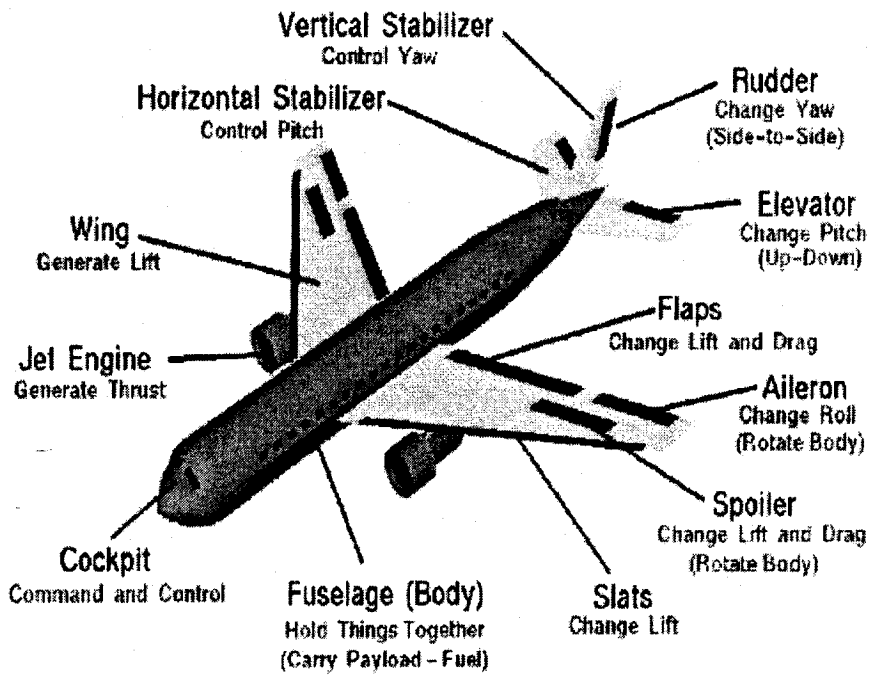



Figure 3.1 Aircraft Parts Definition (Credits – NASA)

(Form: http://www.grc.nasa.gov/WWW/K-12/airplane/airplane.html)

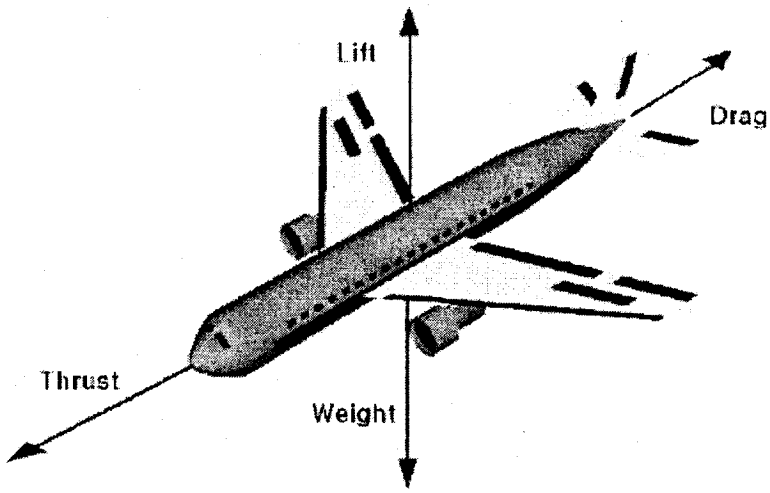



Figure 3.2 Aircraft Forces (Credits – NASA)

(From: http://www.grc.nasa.gov/WWW/K-12/airplane/forces.html)

The motion of aircraft is controlled by the deflections of control surfaces (elevator, aileron, rudder, and so on), which lead to the changes of aerodynamic forces and moments, and the thrust force generated by propeller system. Actually, the most important aspect of aircraft control is to specify the achievable motions by control inputs applied simultaneously.

In this thesis, the UAV is modeled on the basis of references (Stevens B. L. and Lewis F. L. (1992); McLean D. (1990); Nelson R. C. (1998) and AeroSim blockset user guide [1].)

## 3.2 Introduction of Frames

In order to derive mechanical equations, at first, the inertial frame has to be defined. Afterwards, relative frames allow the simplification of the writing of fundamental relationships. To describe the kinematics, dynamics and navigation of UAVs control system, an appropriate frame of reference needs to express the equations of

motion of an UAV. A convenient inertial reference frame is a nonrotating and nonaccelerating Cartesian coordinate, as for example with the origin is fixed at the center of the Earth named Earth Axis Frame which is used to express the gravitational effects, altitude, horizontal distance and the orientation of the UAV.

The UAV itself must have a suitable axis system to describe the forces and moments. The choice of axis system governs the form of the equations of motion. To a great extent, three special axis systems are introduced in the Aircraft Control System.

- Body Axis System

In the body-axis system, the origin of Cartesian axis system is fixed in the center of gravity of UAV. The $x_b$ axis points out of the nose of the UAV and is coincident with the longitudinal axis of the aircraft. The $y_b$ axis is directed out of the right wing of the UAV and the $z_b$ axis is perpendicular to both the $x_b$ and $y_b$ axes and is directed downward. The angular displacement terms roll, pitch and yaw. In the body axis frame, the velocities are expressed as ( $u$, $v$, $w$ ) along the $x_b$, $y_b$, $z_b$ axis respectively. The definition of forces ( $X$, $Y$, $Z$ ), and moments ( $L$, $M$, $N$ ) and angular rates ( $p$, $q$, $r$ ) components are shown in Figure 3.3.

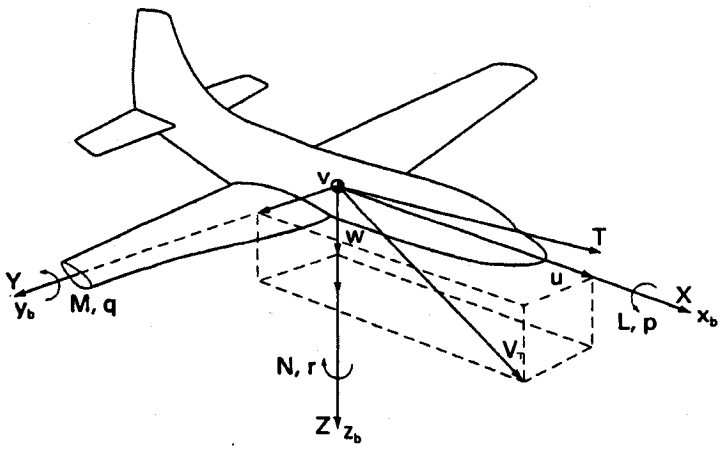



Figure 3.3 Definition of forces, moments, and velocity components in body axis system (From: Nelson R. C. 1998, Figure 1.10)

- Stability Axis System

The $x$ axis is chosen to coincide with the projection of velocity vector $V_P$ into $x_b z_b$ plane. The angle between the $x$ axis of the stability axis system and $x$ axis of the body axis system is named as Angle of Attack ($\alpha$) shown in Figure 3.1.

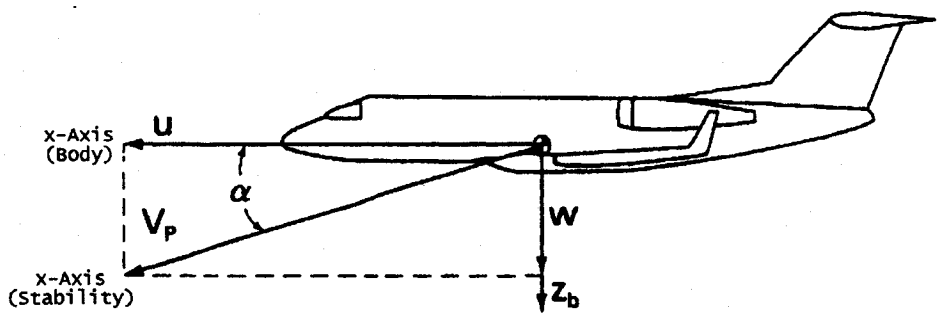



Figure 3.4 Definition of Angle of Attack (AOA)

(From: Nelson R. C. 1998, Figure1.11)

- Wind Axis System

In the wind-axis system, the origin of the rectangular Cartesian system is fixed at the center of gravity of the UAV. The $x$ axis of the wind axis system, which is shown in Figure 3.5, points into the direction of the air velocity vector $V_T$. The angle between the $x$ axis of the stability axis system showing in Figure 3.4 and the $x$ axis of the wind axis system is named as Sideslip angle ($\beta$). The definition of sideslip angle is shown in Figure 3.5.

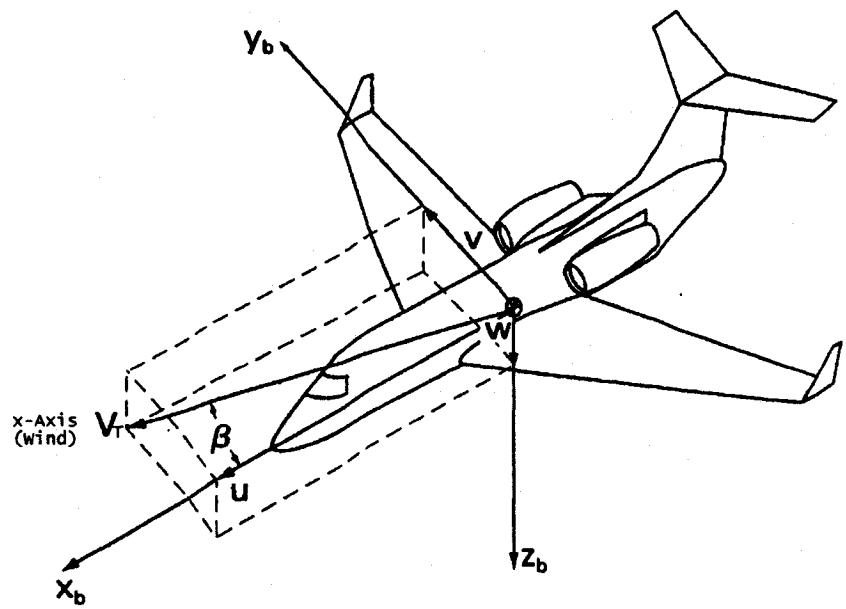



Figure 3.5 Definition of Sideslip Angle

(From: Nelson R. C. 1998, Figure 1.11)

where

$$\alpha = \tan^{-1}\frac{w}{u}$$

$$\beta = \tan^{-1}\frac{v}{V_T}$$

$$V_T^2 = u^2 + v^2 + w^2$$

The transformation from one frame to another is obtained by a transformation matrix. The matrix associated with the rotation around an axis passing through the origin of the orthogonal right-handed frame is a real orthogonal matrix. This matrix has two remarkable characteristics, its transpose is equal to its inverse and its determinant is equal to one. The body frame, stability frame and wind frame are always fixed to the UAV i.e. they move and rotate with UAV body frame with regard to an inertial frame. The rotation matrixes between two different axis systems are expressed as follows:

From Body axis frame to stability axis frame

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix}_S = \begin{bmatrix} \cos\alpha & 0 & \sin\alpha \\ 0 & 1 & 0 \\ -\sin\alpha & 0 & \cos\alpha \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix}_B \tag{3.1}$$

From Stability axis frame to Wind axis frame

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix}_W = \begin{bmatrix} \cos\beta & \sin\beta & 0 \\ -\sin\beta & \cos\beta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix}_S \tag{3.2}$$

From Body axis frame to Wind axis frame

$$\mathbf{T_{WB}} = \begin{bmatrix} x \\ y \\ z \end{bmatrix}_W = \begin{bmatrix} \cos\alpha\cos\beta & \sin\beta & \sin\alpha\cos\beta \\ -\cos\alpha\sin\beta & \cos\beta & -\sin\alpha\sin\beta \\ -\sin\alpha & 0 & \cos\alpha \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix}_B \qquad (3.3)$$

## 3.3 Orientation and Position of UAV

The orientation of the UAV can be defined relative to a fixed frame of reference. Normally, the earth fixed frame is chosen as the reference. At time $t = 0$, the fixed reference frame and body axis frame coincide.

The orientation of UAV is described by three consecutive angular rotations, which are named as Euler Angles. Starting from the reference frame:

1. Rotate about the $z$ axis, ***nose right*** to generate positive yaw angle $\psi$
2. Rotate about the new $y$ axis, ***nose up*** to generate positive pitch angle $\theta$
3. Rotate about the new $x$ axis, ***right wing down*** to generate positive roll (or bank) angle $\phi$

The definition of Euler angle is shown in Figure 3.6.

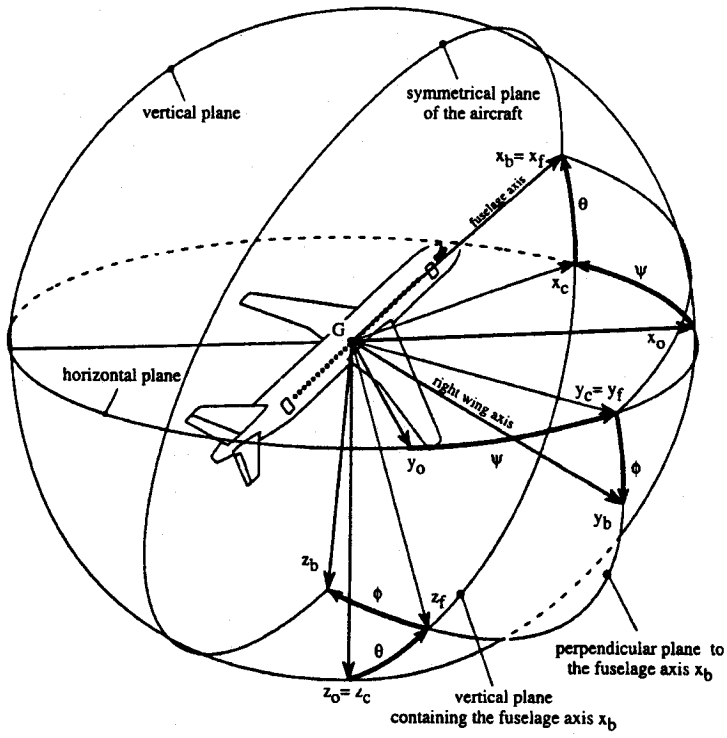



Figure 3.6 Definition of Euler Angle

(From: Boiffier J. L. 1998, figure2.10)

Each rotation can be written as a $3\times3$ rotational transformation matrix.

$$\mathbf{T}_{\psi} = \begin{bmatrix} \cos\psi & \sin\psi & 0 \\ -\sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3.4}$$

$$\mathbf{T}_{\theta} = \begin{bmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{bmatrix} \tag{3.5}$$

$$\mathbf{T}_{\phi} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & \sin\phi \\ 0 & -\sin\phi & \cos\phi \end{bmatrix} \tag{3.6}$$

Since each of rotational transformation matrices are orthogonal and the determinant of each matrix is equal to one, the inverse of the each transformation matrix is equal to its transpose.

The complete transformation sequence from the reference inertial frame to the body axis frame is shown in Figure 3.7 and is given in equation(3.7).

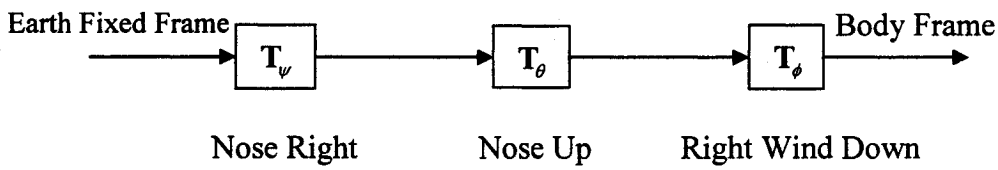



Figure 3.7 Earth Fixed Frame to Body Frame Transformation Sequence

$$\mathbf{T_{IB}} = \begin{bmatrix} \cos\theta\cos\psi & \cos\theta\sin\psi & -\sin\theta \\ \sin\phi\sin\theta\cos\psi - \cos\phi\sin\psi & \sin\phi\sin\theta\sin\psi + \cos\phi\cos\psi & \sin\phi\cos\theta \\ \cos\phi\sin\theta\cos\psi + \sin\phi\sin\psi & \cos\phi\sin\theta\sin\psi - \sin\phi\cos\psi & \cos\phi\cos\theta \end{bmatrix} \quad (3.7)$$

The derivation of all transformation matrixes can be computed by Maple, and the programme is attached in Appendix 3.

The angular velocities in the body axis system, $p, q$ and $r$, are shown in Figure 3.3.

The relationship between the angular velocities and the Euler angular velocities ($\dot{\psi}$ $\dot{\theta}$ $\dot{\phi}$) can be obtained from the kinematic equations for rotations.

$$\begin{bmatrix} \dot{\psi} \\ \dot{\theta} \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} 0 & \frac{\sin\phi}{\cos\theta} & \frac{\cos\phi}{\cos\theta} \\ 0 & \cos\phi & -\sin\phi \\ 1 & \sin\phi\tan\theta & \cos\phi\tan\theta \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix} \tag{3.8}$$

The velocities, in the earth fixed system, can be expressed as

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} = (\mathbf{T}_{\mathbf{IB}})^{-1} \begin{bmatrix} u \\ v \\ w \end{bmatrix} \tag{3.9}$$

$$= \begin{bmatrix} \cos\psi\cos\theta & \cos\psi\sin\theta\sin\phi - \sin\psi\cos\phi & \cos\psi\sin\theta\cos\phi + \sin\psi\sin\phi \\ \sin\psi\cos\theta & \sin\psi\sin\theta\sin\phi + \cos\psi\cos\phi & \sin\psi\sin\theta\cos\phi - \cos\psi\sin\phi \\ -\sin\theta & \cos\theta\sin\phi & \cos\theta\cos\phi \end{bmatrix} \begin{bmatrix} u \\ v \\ w \end{bmatrix}$$

where $x$, $y$, $z$ describe the position of UAV in the earth fixed frame. The equation is named as Navigation Equation for translations.

Using the transformation matrix, the velocity components in body axis system can be expressed with regard to the airspeed $V_T$ as

$$\begin{bmatrix} u \\ v \\ w \end{bmatrix} = V_T \begin{bmatrix} \cos\beta\cos\alpha \\ \sin\beta \\ \cos\beta\sin\alpha \end{bmatrix} \tag{3.10}$$

## 3.4 Definition of Forces and Moments

The forces and moments at the UAV's center of gravity have components due to aerodynamic effects, engine thrust and gravitational force; these components will be denoted respectively, by the subscripts $A$ , $T$ and $G$ . The aerodynamic effects on an UAV

are produced by the relative motion with respect to the air and the orientation with respect to the airflow. In a uniform airflow, two orientation angles are needed to specify the aerodynamic forces and moments. These two angles, which are known as aerodynamics angles, are the Angle Of Attack ($\alpha$) and the Sideslip Angle ($\beta$). In the wind axis system, an $x$ component of the velocity vector at c.g. is the true airspeed $V_T$. The component of thrust is aligned $x$ axis in the body axis system and passed through the c.g., i.e. for UAV modeling in this thesis, the engine thrust doesn't generate the moment to UAV.

The aerodynamic forces and moments are expressed by the dimensionless aerodynamic coefficients. Unfortunately, there is not a formal standard accepted universally to build up aerodynamic coefficients. The nondimenional system (Etkin, B and Reid, L.D 1996) presented the aerodynamic forces and moments are shown in Table 3-1. where $b$, $c$ and $S$ are geometric parameters of wing.

| **Aerodynamic Force/Moment** | **Divisor** | **Dimensionless Coefficients** |
|---|---|---|
| $d$ | $\frac{1}{2}\rho V_T^2 S$ | $C_d$ |
| $y$ | | $C_y$ |
| $l$ | | $C_l$ |
| $M$ | $\frac{1}{2}\rho V_T^2 Sc$ | $C_M$ |
| $L$ | $\frac{1}{2}\rho V_T^2 Sb$ | $C_L$ |
| $N$ | | $C_N$ |

Table 3-1 Aerodynamic Force and Moment Dimensionless System

For instance, the drag force is computed by the drag coefficient.

$$d = \frac{1}{2}\rho V_T^2 S C_d$$

These dimensionless coefficients are not only primarily dependent on the aerodynamic angles, $\alpha$ and $\beta$, but also are dependent on the changing rates of the aerodynamic angles and control surface deflections. Furthermore, the aerodynamic coefficients are also dependent on other factors, such as engine power level, Mach number and so on. To simplify the computation and analysis, the dimensionless coefficients are expressed by the Taylor series approximation for various variables. For example, the effect of drag force coefficient with respect to the changing of the elevator deflection is expressed as the product of the partial derivatives of aerodynamic coefficient and manipulating variable.

$$\frac{\partial C_d}{\partial \delta_e}\delta_e = C_d^{\delta_e}\delta_e$$

In the modeling of aerodynamic forces and moments, the effect of Mach number is ignored for subsonic speeds and control surfaces are limited in aileron, elevator and rudder. The aerodynamic forces acting on the complete UAV in the body frame are defined in following form.

$$\begin{aligned}
&\text{Drag:} && d = \frac{1}{2}\rho V_T^2 S\left[C_{d0} + \frac{(C_L - C_{L0})^2}{\pi e AR} + C_d^{\delta_e}\delta_e + C_d^{\delta_a}\delta_a + C_d^{\delta_r}\delta_r\right] \\
&\text{Sideforce:} && y = \frac{1}{2}\rho V_T^2 S\left[C_y^{\beta}\beta + C_y^{\delta_a}\delta_a + C_y^{\delta_r}\delta_r + \frac{b}{2V_T}\left(C_y^{p}p + C_y^{r}r\right)\right] \\
&\text{Lift:} && l = \frac{1}{2}\rho V_T^2 S\left[C_{l0} + C_l^{\alpha}a + C_l^{\delta_e}\delta_e + \frac{c}{2V_T}\left(C_l^{\dot{\alpha}}\dot{\alpha} + C_l^{q}q\right)\right]
\end{aligned} \tag{3.11}$$

The moments are expressed in the same manner.

$$\text{Rolling Moment:} \quad L=\frac{1}{2}\rho V_T^2 Sb\left[C_L^{\beta}\beta+C_L^{\delta_a}\delta_a+C_L^{\delta_r}\delta_r+\frac{b}{2V_T}\left(C_L^{p}p+C_L^{r}r\right)\right]$$

$$\text{Pitch Moment:} \quad M=\frac{1}{2}\rho V_T^2 Sc\left[C_{M0}+C_M^{\alpha}\alpha+C_M^{\delta_e}\delta_e+\frac{c}{2V_T}\left(C_M^{q}q+C_M^{\dot{\alpha}}\dot{\alpha}\right)\right] \quad (3.12)$$

$$\text{Yawing Moment:} \quad N=\frac{1}{2}\rho V_T^2 Sb\left[C_N^{\beta}\beta+C_N^{\delta_a}\delta_a+C_N^{\delta_r}\delta_r+\frac{b}{2V_T}\left(C_N^{p}p+C_N^{r}r\right)\right]$$

The meanings of dimensionless coefficients are defined in the symbol part. $\delta_e$, $\delta_a$, $\delta_r$ are the deflections of elevator, aileron and rudder. With different setting of these values, the status of UAV is changed accordingly.

The other two forces, propulsive and gravitational forces, are described in the next section.

## 3.5 Derivation of Motion Equations

First, it is assumed that the UAV is a rigid-body, i.e. the distances between any points on the UAV do not change in flight. If the coordinate origin is chosen at the center of mass, the equation of motion for a rigid aircraft can be decoupled into rotation and translation equations. By applying Newton's Second Law to that rigid body, the equations of motion can be established in terms of the translation and angular accelerations which occur as a consequence of some forces and moments being applied to the UAV.

The Newton's second law is used to derive the rigid body equations of motion. i.e. the summation of all external forces acting on a body is equal to the time rate of change

of linear momentum of the body; and the summation of the external moments acting on the body is equal to the time rate of change of angular momentum. The linear and angular momentums are referred to an absolute to the inertial reference frame. The earth fixed frame is considered as the inertial reference frame for many problems in the aircraft dynamics. The vector equation of Newton's second law can be expressed as:

$$\begin{aligned} \sum \mathbf{F} &= \frac{d}{dt}(m\mathbf{v}) \\ \sum \mathbf{M} &= \frac{d}{dt}\mathbf{H} \end{aligned} \tag{3.13}$$

where $\mathbf{v}$ is the absolute velocity vector of the UAV c.g. with respect to the inertial reference system.

Because the reference inertial frame, earth fixed coordinate, is nonrotating, the moments and products of inertia will vary with time. The body fixed axis frame is possible to avoid this difficulty. Therefore, the derivatives of the vectors $\mathbf{v}$ and $\mathbf{H}$ referred to the rotating body fixed frame to the reference frame must be determined. The equation (3.13) can be written as

$$\begin{aligned} \mathbf{F} &= m\frac{d\mathbf{v}_T}{dt}\bigg|_B + m(\boldsymbol{\omega} \times \mathbf{v}_T) \\ \mathbf{M} &= \frac{d\mathbf{H}}{dt}\bigg|_B + \boldsymbol{\omega} \times \mathbf{H} \end{aligned} \tag{3.14}$$

where body frame is rotating with an angular velocity $\boldsymbol{\omega} = p\mathbf{i} + q\mathbf{j} + r\mathbf{k}$ . The subscripts $B$ refer to the body frame of reference.

The scalar force equation is

$$\begin{bmatrix} F_x \\ F_y \\ F_z \end{bmatrix} = m \begin{bmatrix} \dot{u} + qw - rv \\ \dot{v} + ru - pw \\ \dot{w} + pv - qu \end{bmatrix} \tag{3.15}$$

The force vector consists of the three components of the force $F_x$ , $F_y$ , $F_z$ and velocity $u$ ,$v$ ,$w$ along $x, y,$ and $z$ axis in the body axis frame, respectively. The force components are built up by the effects caused by aerodynamic ( $A$ ), propulsive ( $T$ ) and gravitational forces ( $G$ ) acting on the UAV. Assuming that the mass of aircraft is a constant, the external force equation can be rewritten as:

$$\begin{bmatrix} F_x \\ F_y \\ F_z \end{bmatrix} = \begin{bmatrix} F_{x\_A} \\ F_{y\_A} \\ F_{z\_A} \end{bmatrix} + \begin{bmatrix} F_{x\_T} \\ F_{y\_T} \\ F_{z\_T} \end{bmatrix} + \begin{bmatrix} F_{x\_G} \\ F_{y\_G} \\ F_{z\_G} \end{bmatrix} \tag{3.16}$$

The gravitational force will not produce any moment because the origin of body axis system is fixed to the c.g. It will produce only force components along body axes. With the help of the transformation matrix equation(3.7), the gravitational force components can be written as

$$\begin{bmatrix} g_x \\ g_y \\ g_z \end{bmatrix} = \mathbf{T_{IB}} \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix} = \begin{bmatrix} -g\sin\theta \\ g\sin\phi\cos\theta \\ g\cos\phi\cos\theta \end{bmatrix} \tag{3.17}$$

Assuming that the propulsive force generated by the thrust is along $x$ axis in the body axis system, the propulsive part is

$$\begin{bmatrix} F_{x\_T} \\ F_{y\_T} \\ F_{z\_T} \end{bmatrix} = \begin{bmatrix} F_T \\ 0 \\ 0 \end{bmatrix} \tag{3.18}$$

where $F_T = C_{th}\delta_{th}$ is the propulsive force.

The aerodynamic forces are described in equation(3.11) such that the complete equation in the body frame can be written as

$$\begin{bmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{bmatrix} = \frac{1}{m}\left\{ \begin{bmatrix} F_{x\_A} \\ F_{y\_A} \\ F_{z\_A} \end{bmatrix} + \begin{bmatrix} F_T \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} g_x \\ g_y \\ g_z \end{bmatrix} \right\} + \begin{bmatrix} rv - qw \\ pw - ru \\ qu - pv \end{bmatrix} \tag{3.19}$$

and the moment equation

$$\begin{bmatrix} L \\ M \\ N \end{bmatrix} = \begin{bmatrix} \dot{H}_x + qH_z - rH_y \\ \dot{H}_y + rH_x - pH_z \\ \dot{H}_z + pH_y - qH_x \end{bmatrix} \tag{3.20}$$

where $L, M, N$ are the components of the moment which is shown in equation (3.12) and $H_x, H_y, H_z$ are moment of momentum along the $x$, $y$, and $z$ axes, respectively.

$$\begin{aligned} H_x &= pI_x - qI_{xy} - rI_{xz} \\ H_y &= -pI_{xy} + qI_y - rI_{yz} \\ H_z &= -pI_{xz} - qI_{yz} + rI_z \end{aligned} \tag{3.21}$$

Normally, the UAV is symmetric with regard to the $xz$ plane which leads to the products of inertia $I_{yz} = I_{xy} = 0$. The moment equations can be written as

$$\begin{aligned} L &= I_x \dot{p} - I_{xz} \dot{r} + qr\left(I_z - I_y\right) - I_{xz} pq \\ M &= I_y \dot{q} + rp\left(I_x - I_z\right) + I_{xz}\left(p^2 - r^2\right) \\ N &= -I_{xz} \dot{p} + I_z \dot{r} + pq\left(I_y - I_z\right) + I_{xz} qr \end{aligned} \tag{3.22}$$

The force and moment equations are described in the body fixed frame. Furthermore, these equations can be written in the wind axis system as well.

Force equation (Wind Axis System)

$$\begin{bmatrix} \dot{V}_T \\ \dot{\beta} \\ \dot{\alpha} \end{bmatrix} = \frac{1}{m}\left\{\begin{bmatrix} F_T \cos\alpha\cos\beta \\ \dfrac{-F_T \cos\alpha \sin\beta}{V_T} \\ \dfrac{-F_T \sin\alpha}{V_T \cos\beta} \end{bmatrix} + \begin{bmatrix} -d \\ \dfrac{y}{V_T} \\ \dfrac{-l}{V_T \cos\beta} \end{bmatrix}\right\} + \begin{bmatrix} 0 \\ -r_w \\ \dfrac{q_w}{\cos\beta} \end{bmatrix} + \begin{bmatrix} g_1 \\ \dfrac{g_2}{V_T} \\ \dfrac{g_3}{V_T \cos\beta} \end{bmatrix} \tag{3.23}$$

The subscript $w$ is used to denote wind-axes quantities. The angular velocity in the wind axis system is

$$\begin{bmatrix} p_w \\ q_w \\ r_w \end{bmatrix} = \mathbf{T_{BW}} \begin{bmatrix} p \\ q \\ r \end{bmatrix} = \begin{bmatrix} p\cos\alpha\cos\beta + q\sin\beta + r\sin\alpha\cos\beta \\ -p\cos\alpha\sin\beta + q\cos\beta - r\sin\alpha\sin\beta \\ -p\sin\alpha + r\cos\alpha \end{bmatrix} \tag{3.24}$$

and the gravitational force in the wind-axes frame is written as

$$\begin{bmatrix} g_1 \\ g_2 \\ g_3 \end{bmatrix} = \mathbf{T}_{\mathbf{BW}} \begin{bmatrix} g_x \\ g_y \\ g_z \end{bmatrix} = \mathbf{T}_{\mathbf{BW}} \begin{bmatrix} -g\sin\theta \\ g\sin\phi\cos\theta \\ g\cos\phi\cos\theta \end{bmatrix} \tag{3.25}$$
$$= \begin{bmatrix} g(-\cos\alpha\cos\beta\sin\theta + \sin\beta\sin\phi\cos\theta + \sin\alpha\cos\beta\cos\phi\cos\theta) \\ g(\cos\alpha\sin\beta\sin\theta + \cos\beta\sin\phi\cos\theta - \sin\alpha\sin\beta\cos\phi\cos\theta) \\ g(\sin\alpha\sin\theta + \cos\alpha\cos\phi\cos\theta) \end{bmatrix}$$

The wind-axes frame moment equation is more complex than the original body-axes equations. Therefore, a nonlinear model will typically use the body-axes moment equations combined with force equations in either in wind or body axes.

## 3.6 Nonlinear Model of UAV in State Space Format

The force equation in the wind frame, moment equation in body frame and kinematics equations are used for the UAV control design and flight simulation. This 3D rigid body model ignores part of actual UAV dynamics, for example, control surface dynamics, flexibility of wings etc. The elements of the state vector are composed of velocity vector in wind axis system, angular velocities in body axis system and Euler angles.

$$\mathbf{x} = [V_T \quad \alpha \quad \beta \quad p \quad q \quad r \quad \phi \quad \theta \quad \psi] \tag{3.26}$$

The three equation are expressed as follow

1. Force equation (Wind-Axes System):

$$\begin{bmatrix} \dot{V}_T \\ \dot{\beta} \\ \dot{\alpha} \end{bmatrix} = \frac{1}{m}\left\{\begin{bmatrix} F_T\cos\alpha\cos\beta \\ \dfrac{-F_T\cos\alpha\sin\beta}{V_T} \\ \dfrac{-F_T\sin\alpha}{V_T\cos\beta} \end{bmatrix} + \begin{bmatrix} -d \\ \dfrac{y}{V_T} \\ \dfrac{-l}{V_T\cos\beta} \end{bmatrix}\right\} + \begin{bmatrix} 0 \\ -r_w \\ \dfrac{q_w}{\cos\beta} \end{bmatrix} + \begin{bmatrix} g_1 \\ \dfrac{g_2}{V_T} \\ \dfrac{g_3}{V_T\cos\beta} \end{bmatrix} \quad (3.27)$$

The aerodynamic forces, $d$, $y$, $l$, are given in equation (3.11) and the

2. Moment Equation (Body Axis System):

$$\begin{aligned} \dot{p} &= (c_1 r + c_2 p)q + c_3 L + c_4 N \\ \dot{q} &= c_5 pr - c_6\left(p^2 - r^2\right) + c_7 M \\ \dot{r} &= (c_8 p - c_2 r)q + c_4 L + c_9 N \end{aligned} \quad (3.28)$$

where the inertial coefficients are defined as follow:

$$c_1 = \frac{(I_y - I_z)I_z - I_{xz}^2}{\Gamma} \qquad c_2 = \frac{\left(I_x - I_y + I_z\right)I_{xz}}{\Gamma}$$

$$c_3 = \frac{I_z}{\Gamma} \qquad c_4 = \frac{I_{xz}}{\Gamma}$$

$$c_5 = \frac{I_z - I_x}{I_y} \qquad c_6 = \frac{I_{xz}}{I_y}$$

$$c_7 = \frac{1}{I_y} \qquad c_8 = \frac{\left(I_x - I_y\right)I_x + I_{xz}^2}{\Gamma}$$

$$c_9 = \frac{I_x}{\Gamma} \qquad \text{where } \Gamma = I_x I_z - I_{xz}^2$$

To simplify the inertial coefficients, all cross-coupling terms are neglected.

$$c_1 = \frac{I_y - I_z}{I_x} \qquad c_2 = 0 \qquad c_3 = \frac{1}{I_x}$$

$$c_4 = 0 \qquad c_5 = \frac{I_z - I_x}{I_y} \qquad c_6 = 0$$

$$c_7 = \frac{1}{I_y} \qquad c_8 = \frac{I_x - I_y}{I_z} \qquad c_9 = \frac{1}{I_z}$$

Simplified Moment Equations

$$\begin{aligned} \dot{p} &= c_1 rq + c_3 L \\ \dot{q} &= c_5 pr + c_7 M \\ \dot{r} &= c_8 pq + c_9 N \end{aligned} \tag{3.29}$$

To avoid extensive computation, the simplified moment equations are used for the simulation in this thesis. The aerodynamic moments, $L$, $M$, $N$, are given in equation (3.12).

3. Kinematic Equations for rotation

$$\begin{aligned} \dot{\phi} &= p + \tan\theta\left(q\sin\phi + r\cos\phi\right) \\ &= p + \dot{\psi}\sin\theta \\ \dot{\theta} &= q\cos\phi - r\sin\phi \\ \dot{\psi} &= \frac{q\sin\phi + r\cos\phi}{\cos\theta} \end{aligned} \tag{3.30}$$

4. Navigation Equations

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} = \begin{bmatrix} \cos\psi\cos\theta & \cos\psi\sin\theta\sin\phi - \sin\psi\cos\phi & \cos\psi\sin\theta\cos\phi + \sin\psi\sin\phi \\ \sin\psi\cos\theta & \sin\psi\sin\theta\sin\phi + \cos\psi\cos\phi & \sin\psi\sin\theta\cos\phi - \cos\psi\sin\phi \\ -\sin\theta & \cos\theta\sin\phi & \cos\theta\cos\phi \end{bmatrix} \begin{bmatrix} u \\ v \\ w \end{bmatrix} \tag{3.31}$$

The complete six-degree-freedom nonlinear model for rigid body aircraft consists of force and moment equations (3.27) and (3.29), kinematics equation (3.30) to describe the attitude of aircraft and navigation equation (3.31) to express the position in the earth fixed system.

# Chapter 4

# 4 UAV FLIGHT CONTROLLER DESIGN

In this section, some classical autopilot controllers are introduced and inner / outer loop dynamic inversion and a Model Predictive Control (MPC) are presented.

## 4.1 Review of Classical Linear Controller for Aircraft Control Design

In the past, most autopilot controllers were designed based on the PID feedback control system using a linearized model in the specified trim conditions. The block diagrams for the attitude, altitude hold and speed are shown in this section.

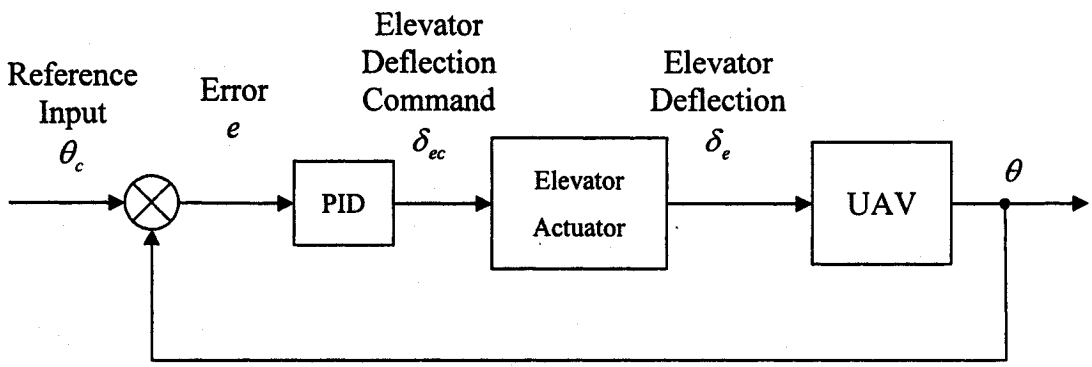



Figure 4.1 Pitch Attitude Autopilot Control with PID Controller

In Figure 4.1, an example for the pitch attitude control with a PID controller is shown. To follow the desired pitch angle, the current pitch angle is measured by vertical gyro or obtained by the integration of rate gyro output. The error is multiplied by $k_p$ , $k_I$ and $k_D$ , then three signals are summed up as the elevator control surface deflection command. At last, the output of elevator system is applied for the UAV and

deflection of elevator causes the UAV to achieve a desired pitch orientation. The parameters of controller are obtained using a linearized UAV model or by trail and error approach. The actuator is subject to only the local controller.

The block diagram for yaw and roll orientation control with rudder and aileron are shown in Figure 4.2 and Figure 4.3.

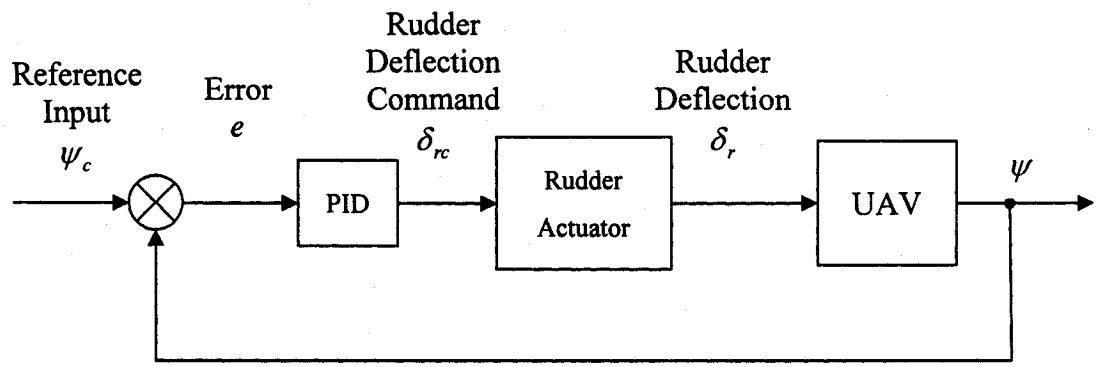



Figure 4.2 Yaw Attitude Autopilot Control with PID Controller

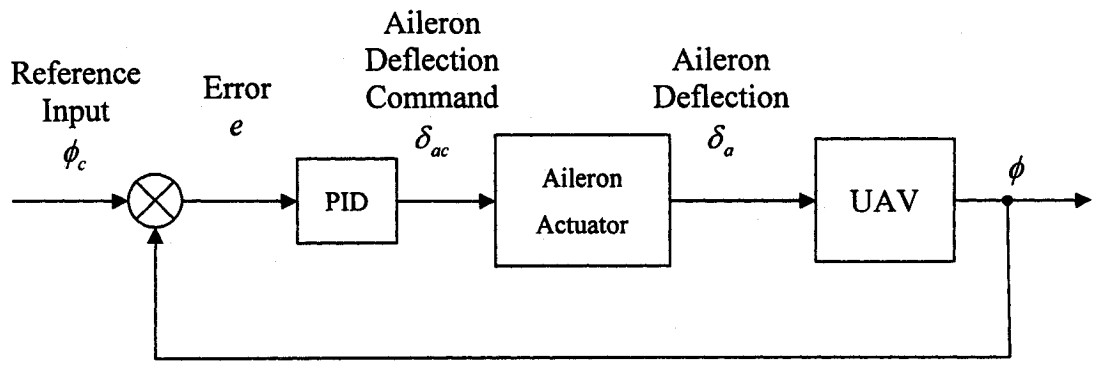



Figure 4.3 Roll Attitude Autopilot Control with PID Controller

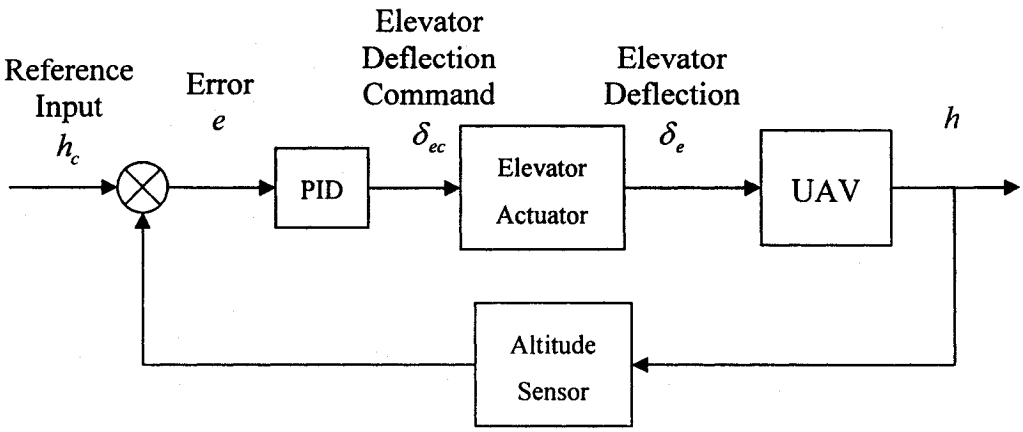



Figure 4.4 Altitude Hold Control System

The simplified the altitude hold system is shown in Figure 4.4. The difference between actual altitude and desired altitude is obtained by changing the pitch angle through the elevator.

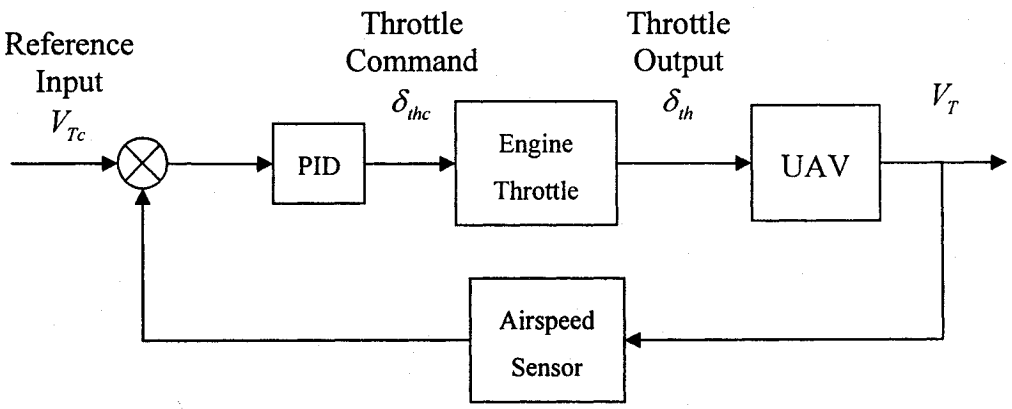



Figure 4.5 Airspeed Control System using Engine Thrust

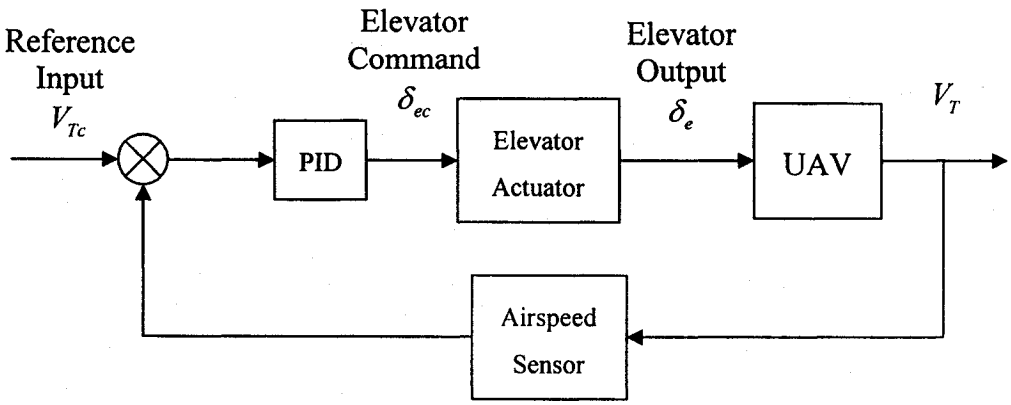



Figure 4.6 Airspeed Control System by Elevator

There are two ways to control airspeed. In Figure 4.5, the airspeed of UAV is controlled by changing the engine throttle setting to increase or decrease the thrust. If the airspeed is still too low under a certain throttle setting, UAV can execute a dive operation with deflection of elevator, shown in Figure 4.6, until it obtains the desired airspeed. On the other hand, the UAV will climb up in order to reduce the airspeed.

We observe that the different desired variables are controlled by the same control surface. This may lead to concurrent command under some operation. For example, if the UAV is requested to climb to higher altitude level without decreasing airspeed, this causes a concurrent command. From control strategy shown in Figure 4.4, a positive pitch is needed to climb to the higher level, while the climbing operation leads to lower airspeed. To achieve the specified manoeuvre and avoid concurrent control command, the high level control system is needed to coordinate the selection of different control strategies. For this example, it can be carried out by the altitude control through the elevator deflection and constant speed by increasing throttle setting.

The above mentioned technology is very common for current commercial autopilot systems because of their high reliability and easy implementation, but the main drawback is that only one control input influences one state of UAV and the cross coupling of different states is not considered. With the help of modern control theory, the system is expressed by the state space form to explore the dynamic characteristic of a Multi-Input Multi-Output (MIMO) system. Furthermore, the parameter of the controller is defined by the linearized model under the certain trim conditions, and there are no guaranteed performances in the whole flight envelope.

From chapter 3, the UAV model is described by the nonlinear functions. To meet the requirements of high-performance UAV, several different nonlinear control laws are proposed in the last decade. These control law approaches, which include fuzzy logic control, adaptive control, neural network control, variable structure control and model predictive control and dynamic inversion in addition to the conventional gain scheduled control law, are compared in the reference (Steinbery M. L. 2001). As the control laws based on the Dynamic Inversion controller provide improved performance is fairly robust over conventional flight control designs, especially for high angles of attack and high angular rate, and is valid over the entire flight envelope. Compared with some conventional control approaches, the dynamic inversion controller is considered as the baseline for the other control approaches. In recent years, the dynamic inversion based controller has been successfully used for some complex nonlinear aircraft simulations, such as X-38 (Ito D. et al 2001). The dynamic inversion controller is considered as the basic part for the UAV control in this thesis.

## 4.2 Introduction of Dynamic Inverse Control

The basic concept behind dynamic inverse control is to linearize the nonlinear system by canceling the natural dynamics through active compensation using negative feedback control. Through the dynamic inversion control, a linearized model is generated from the nonlinear system and the conventional design approaches are applied for the further design to it. Since dynamic inversion is a model based design approach, a fundamental assumption for dynamic inverse approach is that the UAV dynamic model is perfectly known and all states for computation of negative feedback loop are observable at all time.

Normally, a nonlinear system for dynamic inversion is put in the form of

$$\begin{aligned} \dot{\mathbf{x}} &= \mathbf{f}(\mathbf{x}) + \mathbf{g}(\mathbf{x})\mathbf{u} \\ \dot{\mathbf{y}} &= \mathbf{c}\mathbf{x} \end{aligned} \tag{4.1}$$

where $\mathbf{x}$ is the vector of a states used for dynamic inversion, $\mathbf{u}$ is a control inputs vector and $\mathbf{f}(\mathbf{x})$ and $\mathbf{g}(\mathbf{x})$ are nonlinear functions. The outputs consists of states, and the output matrix, $\mathbf{c}$., is therefore a constant matrix. Furthermore, the number control inputs is taken the same as the number of states so that $\mathbf{g}(\mathbf{x})$ is a square matrix and it is invertible within the envelope of states. The control inputs to the nonlinear system can be defined as

$$\mathbf{u} = \mathbf{g}^{-1}(\mathbf{x})(\mathbf{v} - \mathbf{f}(\mathbf{x})) \tag{4.2}$$

where $\mathbf{v}$ is a new vector to be constructed. To guarantee the validation of control inputs $\mathbf{u}$, $\mathbf{g}(\mathbf{x})$ must be invertible for all values of $\mathbf{x}$.

Substituting equation (4.2) into (4.1), results in

$$\dot{\mathbf{x}} = \mathbf{v} \tag{4.3}$$

This is a linear system as the nonlinearities in $\mathbf{f}(\mathbf{x})$ and $\mathbf{g}(\mathbf{x})$ are cancelled.

Several control laws can be implemented to construct the new input vector $\mathbf{v}$. Two of them are applied for the UAV controller design in this thesis.

The first is proportional controller, such that

$$\mathbf{v} = \mathbf{K}_p(\mathbf{x}_c - \mathbf{x}) + \dot{\mathbf{x}}_c \tag{4.4}$$

where $\mathbf{K}_p$ is $n \times n$ diagonal matrix of control gains.

The second is proportional-integral controller, with

$$\mathbf{v} = \dot{\mathbf{x}}_c + \mathbf{K}_p(\mathbf{x}_c - \mathbf{x}) + \mathbf{K}_i \int (\mathbf{x}_c - \mathbf{x})\,dt \tag{4.5}$$

where $\mathbf{K}_p$ and $\mathbf{K}_i$ are $n \times n$ diagonal matrix of control gains. The block diagram of dynamic inversion control is shown in Figure 4.7.

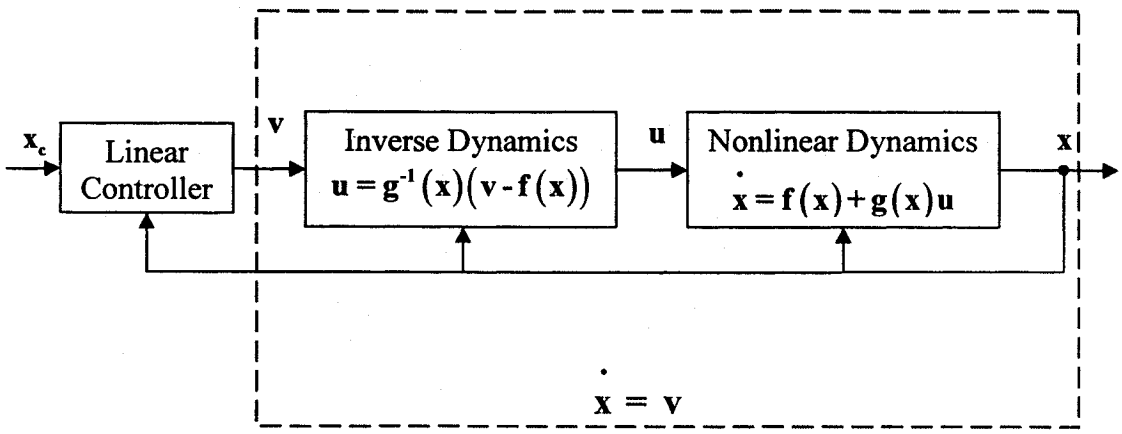



Figure 4.7 Nonlinear Dynamic Inversion Control

## 4.3 Dynamic Inversion Control of UAV

From chapter 3, the nonlinear UAV is modeled by 9 equations and three extra equations for the navigation. Since the number of control inputs is less than the number of states, the straightforward dynamic inversion can't be applied directly. Therefore, two time-scale separation design approach is used. First, the states of nonlinear UAV are separated into fast and slow variables (Menon P.K.A. et al 1987) in the two time scales. The fast variable are three angular rate, $p$, $q$, $r$, which is related directly to the UAV three control inputs from the aileron, elevator, and rudder, and the slow variables are chosen as one of following triplets ($\alpha$, $\beta$, $\phi$), ($\phi$, $\theta$, $\beta$) and ($p$, $\alpha$, $\beta$) which are used to generate the command input ($p_c$, $q_c$, $r_c$) for the fast variables (Azam M. and Singh S. N. (1994) and Kato O. and Sugiura I. (1992))

After the time-scale separation, the number of fast states (three angular rates $p,\ q,\ r$ ) is equal to the number of inputs (three control surface deflections $\delta_a,\ \delta_e,\ \delta_r$ ) so that dynamic inversion can be applied. Inner / outer loop control approach is applied to the fast and slow variables respectively. The fast dynamics is designed in the inner loop and one set of slow variables in this case, $\phi,\ \theta,\ \beta$, is selected in the outer loop design in this thesis. The overall inner/ outer loop control are shown in Figure 4.8.

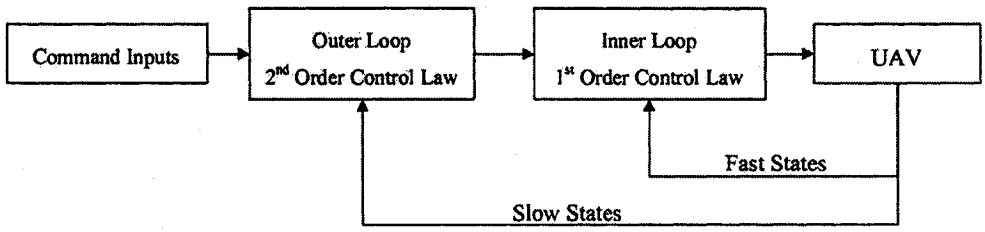



Figure 4.8 Inner /Outer Loop Control

### *4.3.1 Inner Loop Design*

From the simplified moment equation (3.12) and (3.29), the fast states can be written as follow in the matrix form

$$\begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} f_p \\ f_q \\ f_r \end{bmatrix} + \mathbf{A(x)} \begin{bmatrix} \delta_a \\ \delta_r \\ \delta_e \end{bmatrix} \tag{4.6}$$

where

$$f_p = \frac{I_y - I_z}{I_x} rq + \frac{\frac{1}{2}\rho V_T^2 Sb}{I_x}\left[C_L^\beta \beta + \frac{b}{2V_T}\left(C_L^p p + C_L^r r\right)\right]$$

$$f_q = \frac{I_z - I_x}{I_y} pr + \frac{\frac{1}{2}\rho V_T^2 Sc}{I_y}\left[C_{M0} + C_M^\alpha \alpha + \frac{c}{2V_T}\left(C_M^q q + C_M^{\dot\alpha}\dot\alpha\right)\right] \quad (4.7)$$

$$f_r = \frac{I_x - I_y}{I_z} pq + \frac{\frac{1}{2}\rho V_T^2 Sb}{I_z}\left[C_N^\beta \beta + \frac{b}{2V_T}\left(C_N^p p + C_N^r r\right)\right]$$

and

$$\mathbf{A}(\mathbf{x}) = \begin{bmatrix} \frac{\frac{1}{2}\rho V_T^2 Sb}{I_x} C_L^{\delta_a} & \frac{\frac{1}{2}\rho V_T^2 Sb}{I_x} C_L^{\delta_r} & 0 \\ 0 & 0 & \frac{\frac{1}{2}\rho V_T^2 Sc}{I_y} C_M^{\delta_e} \\ \frac{\frac{1}{2}\rho V_T^2 Sb}{I_z} C_N^{\delta_a} & \frac{\frac{1}{2}\rho V_T^2 Sb}{I_z} C_N^{\delta_r} & 0 \end{bmatrix} \quad (4.8)$$

Denoting $\mathbf{y} = [p \quad q \quad r]^T$ and $\mathbf{u} = [\delta_a \quad \delta_r \quad \delta_e]^T$. Then the equations (4.6) have the form of an affine system

$$\dot{\mathbf{y}} = \mathbf{f}(\mathbf{x}) + \mathbf{A}(\mathbf{x})\mathbf{u} \quad (4.9)$$

To cancel the nonlinear part for a new control input $\mathbf{v}$ to be defined later, let

$$\mathbf{u} = \mathbf{A}^{-1}(\mathbf{x})(\mathbf{v} - \mathbf{f}(\mathbf{x})) \quad (4.10)$$

Since the determine of matrix $\mathbf{A}(\mathbf{x})$ can be expressed as

$$\det\left[\mathbf{A}(\mathbf{x})\right] = -\frac{1}{8}\frac{\rho^2 V_T^6 S^3 b^2 c C_M^{\delta_e}\left(C_L^{\delta_a} C_N^{\delta_r} - C_N^{\delta_a} C_L^{\delta_r}\right)}{I_x I_y I_z} \tag{4.11}$$

$\mathbf{A}(\mathbf{x})$ is nonsingular on the condition that

$$C_L^{\delta_a} C_N^{\delta_r} \neq C_N^{\delta_a} C_L^{\delta_r}$$

If the matrix $\mathbf{A}(\mathbf{x})$ is invertible and all states are available at all time to compute $\mathbf{A}(\mathbf{x})^{-1}$, the first order proportional control law for new input $\mathbf{v}$ can be chosen as

And equation (4.10) gives

$$\mathbf{v} = \dot{\mathbf{y}}_c - \mathbf{K}_1(\mathbf{y} - \mathbf{y}_c) \tag{4.12}$$

$$\mathbf{u} = \mathbf{A}(\mathbf{x})^{-1}\left[-\mathbf{f}(\mathbf{x}) + \dot{\mathbf{y}}_c - \mathbf{K}_1(\mathbf{y} - \mathbf{y}_c)\right] \tag{4.13}$$

where $\mathbf{y}_c = \begin{bmatrix} p_c & q_c & r_c \end{bmatrix}^T$ is the command trajectory and $\mathbf{K}_1$ is a $3\times3$ diagonal matrix.

Substituting the first order control law equation(4.13) in equation(4.9), gives

$$\left(\dot{\mathbf{y}} - \dot{\mathbf{y}}_c\right) + \mathbf{K}_1(\mathbf{y} - \mathbf{y}_c) = \mathbf{0} \tag{4.14}$$

If the diagonal values of matrix $\mathbf{K}_1$ are greater than zero, the trajectory error $(\mathbf{y}-\mathbf{y}_c)$ converges to zero exponentially as $t \to \infty$. In this case, the UAV will follow the command trajectory exactly.

The block diagram of the inner loop first order proportional controller based on equations (4.12) and (4.13) is shown in Figure 4.9.

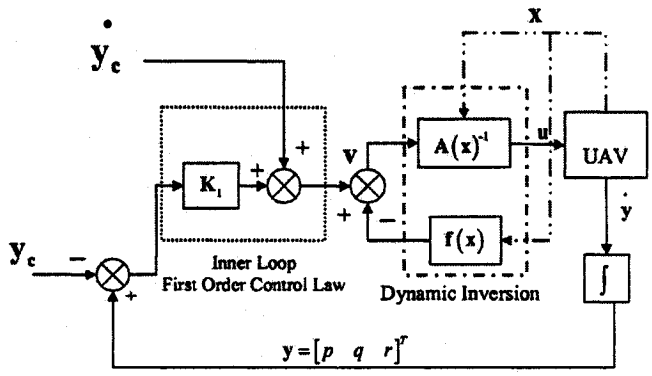



Figure 4.9 Inner Loop Controller

After canceling the nonlinearities of UAV dynamic model, the linearized system can be written as

$$\dot{\mathbf{y}} = \mathbf{v} \tag{4.15}$$

Nonlinear system linearization was defined for the case that all states are observable and the system is affine. Otherwise, more complex input-output linearization approaches have to be applied (Slotine J. E. and Li W. 1991).

### 4.3.2 *Outer Loop Design*

The outer loop receive the command trajectory of the slow states ($\phi_c$, $\theta_c$, $\beta_c$), which is produced by the navigation level and generates the reference inputs ($p_c$, $q_c$, $r_c$) for the inner loop.

Since the aileron, rudder, and elevator produce small forces, we assume $C_y^{\delta_a} = C_y^{\delta_r} = 0$ and engine thrust force $F_T$ is assumed constant.

From force equation (3.11), (3.27) and kinematics equation (3.30), we get

$$\dot{\mathbf{z}} = \begin{bmatrix} 0 \\ 0 \\ h_\beta \end{bmatrix} + \mathbf{G}(\mathbf{x}) \begin{bmatrix} p \\ q \\ r \end{bmatrix} \tag{4.16}$$

where $\dot{z} = \begin{bmatrix} \dot{\phi} & \dot{\theta} & \dot{\beta} \end{bmatrix}^T$

$$h_\beta = \frac{-F_T \cos\alpha \sin\beta + \frac{1}{2}\rho V_T^2 S C_y^\beta \beta}{mV_T} + \frac{g_2}{V_T} \tag{4.17}$$

and

$$\mathbf{G}(\mathbf{x}) = \begin{bmatrix} 1 & \sin\phi\tan\theta & \cos\phi\tan\theta \\ 0 & \cos\phi & -\sin\phi \\ \frac{\rho Sb}{4m} C_y^p + \sin\alpha & 0 & \frac{\rho Sb}{4m} C_y^r - \cos\alpha \end{bmatrix} \tag{4.18}$$

The determinant of matrix $\mathbf{G}(\mathbf{x})$ is

$$\det(\mathbf{G}(\mathbf{x})) = \cos\phi\left(\frac{\rho Sb}{4m}C_y^r - \cos\alpha\right) - \tan\theta\left(\frac{\rho Sb}{4m}C_y^p + \sin\alpha\right) \tag{4.19}$$

The matrix $\mathbf{G}(\mathbf{x})$ is non-singular when

$$\mathbf{x} \in \left\{\cos\phi\left(\frac{\rho Sb}{4m}C_y^r - \cos\alpha\right) \neq \tan\theta\left(\frac{\rho Sb}{4m}C_y^p + \sin\alpha\right)\right\}$$

Then the equation (4.16) can be written as

$$\dot{\mathbf{z}} = \mathbf{H}(\mathbf{x}) + \mathbf{G}(\mathbf{x})\mathbf{y} \tag{4.20}$$

From inner loop controller design, the angular rates $\mathbf{y}$ follow the command trajectory $\mathbf{y}_c$, i.e. $\mathbf{y} \to \mathbf{y}_c$ as $t \to \infty$. If settling time is fast enough, the $\mathbf{y}_c$ can be used instead of $\mathbf{y}$. Therefore, the equation (4.20) gives

$$\dot{\mathbf{z}} = \mathbf{H}(\mathbf{x}) + \mathbf{G}(\mathbf{x})\mathbf{y}_c \tag{4.21}$$

To cancel nonlinearities, $\mathbf{y}_c$ is chosen as

$$\mathbf{y}_c = \mathbf{G}^{-1}(\mathbf{x})\left[-\mathbf{H}(\mathbf{x}) + \boldsymbol{\eta}\right] \tag{4.22}$$

where $\boldsymbol{\eta}$ are new inputs to be constructed.

A second order proportional plus integral controller is applied for the outer loop design where the integral feedback is used to obtain robustness to the uncertainty of parameters in the system model and to ensure zero steady state error.

The second order PI control law generates the new input $\boldsymbol{\eta}$.

$$\boldsymbol{\eta} = \dot{\mathbf{z}}_c - \mathbf{K}_p(\mathbf{z} - \mathbf{z}_c) - \mathbf{K}_i \int (\mathbf{z} - \mathbf{z}_c)\,dt \qquad (4.23)$$

Where $\mathbf{z}_c = [\phi_c \quad \theta_c \quad \beta_c]^T$ and $\mathbf{K}_p, \mathbf{K}_i$ are diagonal gain matrix.

Substituting equations (4.23) in(4.22), we get

$$\mathbf{y}_c = \mathbf{G}(\mathbf{x})^{-1}\left[-\mathbf{H}(\mathbf{x}) + \dot{\mathbf{z}}_c - \mathbf{K}_p(\mathbf{z} - \mathbf{z}_c) - \mathbf{K}_i \int (\mathbf{z} - \mathbf{z}_c)\,dt\right] \qquad (4.24)$$

From equation (4.21)

$$\left(\dot{\mathbf{z}} - \dot{\mathbf{z}}_c\right) + \mathbf{K}_p(\mathbf{z} - \mathbf{z}_c) + \mathbf{K}_i \int (\mathbf{z} - \mathbf{z}_c)\,dt = \mathbf{0} \qquad (4.25)$$

With the suitable selection of gain matrixes $\mathbf{K}_p$ and $\mathbf{K}_i$, the error $(\mathbf{z} - \mathbf{z}_c)$ is made to converge to zero as $t \to \infty$, i.e. the output loop follows the reference inputs $\mathbf{z}_c = [\phi_c \quad \theta_c \quad \beta_c]^T$.

Based on equation (4.21) and (4.24), the block diagram of outer loop control law is shown in Figure 4.10.

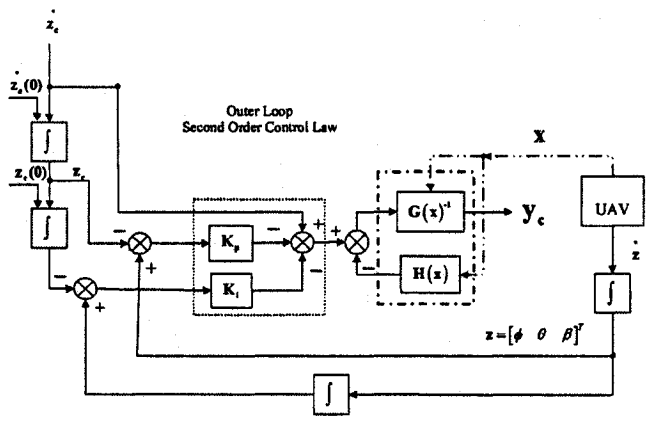



Figure 4.10 Outer Loop Controller

For equation(4.13), two inputs, $\mathbf{y}_c$ and $\dot{\mathbf{y}}_c$ are needed to construct control law for inner loop. $\mathbf{y}_c$ is given in equation (4.24) and $\dot{\mathbf{y}}_c$ is obtained.

by differentiating equation(4.24),

$$\begin{aligned}\dot{\mathbf{y}}_c = &-\mathbf{G}(\mathbf{x})^{-1}\left[\mathbf{L}_f\mathbf{G}(\mathbf{x})\right]\mathbf{G}(\mathbf{x})^{-1}\left[-\mathbf{H}(\mathbf{x})+\dot{\mathbf{z}}_c-\mathbf{K}_p(\mathbf{z}-\mathbf{z}_c)-\mathbf{K}_i\int(\mathbf{z}-\mathbf{z}_c)\,dt\right] \\ &+\mathbf{G}(\mathbf{x})^{-1}\left[-\mathbf{L}_f\mathbf{H}(\mathbf{x})+\ddot{\mathbf{z}}_c-\mathbf{K}_p\left(\dot{\mathbf{z}}-\dot{\mathbf{z}}_c\right)-\mathbf{K}_i(\mathbf{z}-\mathbf{z}_c)\right]\end{aligned} \tag{4.26}$$

where $\mathbf{L}_f$ * is Lie derivative is used in this equation, and $\mathbf{K}_p$ and $\mathbf{K}_i$ are diagonal gain matrices.

Based on equation(4.26), the block diagram for the computation of $\dot{\mathbf{y}}_c$ is shown in Figure 4.11.

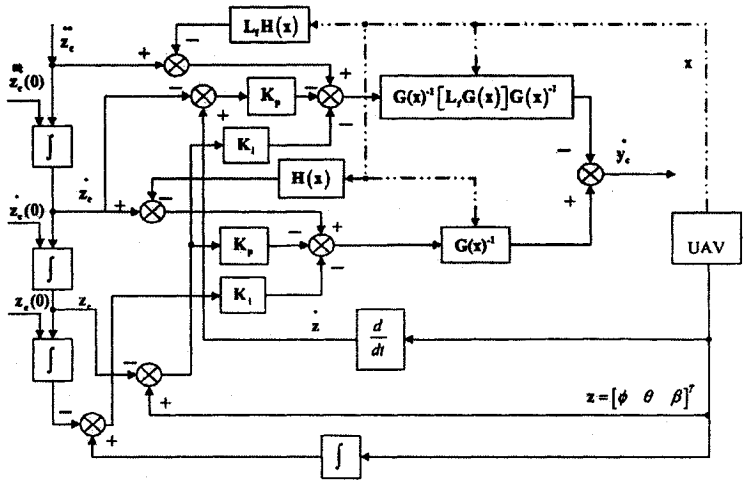



Figure 4.11 Computation of $\dot{\mathbf{y}}_c$

Lie Derivative of $\mathbf{G}(\mathbf{x})$ is

$$\mathbf{L_f G(x)} = \begin{bmatrix} 0 & \dot{\phi}\cos\phi\tan\theta + \dot{\theta}\frac{\sin\phi}{\cos^2\theta} & -\dot{\phi}\sin\phi\tan\theta + \dot{\theta}\frac{\cos\phi}{\cos^2\theta} \\ 0 & -\dot{\phi}\sin\phi & -\dot{\phi}\cos\phi \\ \dot{\alpha}\cos\alpha & 0 & \dot{\alpha}\sin\alpha \end{bmatrix} \tag{4.27}$$

where

$$\begin{aligned}
\dot{\alpha} &= \frac{-F_T\sin\alpha - \frac{1}{2}\rho V_T^2\left[C_{L0} + C_L^{\alpha}\alpha + \frac{c}{2V_T}C_L^{q}q\right]}{mV_T\cos\beta} + \frac{-p\cos\alpha\sin\beta + q\cos\beta - r\sin\alpha\sin\beta}{\cos\beta} \\
&\quad + \frac{g(\sin\alpha\sin\theta + \cos\alpha\cos\phi\cos\theta)}{V_T\cos\beta} \\
\dot{\beta} &= \frac{-F_T\cos\alpha\sin\beta + \frac{1}{2}\rho V_T^2 S\left[C_y^{\beta}\beta + \frac{b}{2V_T}\left(C_y^{p}p + C_y^{r}r\right)\right]}{mV_T} - (-p\sin\alpha + r\cos\alpha) \\
&\quad + \frac{g(\cos\alpha\sin\beta\sin\theta + \cos\beta\sin\phi\cos\theta - \sin\alpha\sin\beta\cos\phi\cos\theta)}{V_T} \\
\dot{\theta} &= q\cos\phi - r\sin\phi \\
\dot{\phi} &= p + \tan\theta(q\sin\phi + r\cos\phi)
\end{aligned} \tag{4.28}$$

and

$$\mathbf{L_f H(x)} = \begin{bmatrix} 0 \\ 0 \\ L_f h_\beta \end{bmatrix} \tag{4.29}$$

where

$$\begin{aligned} L_f h_\beta = \dot{\beta} & \left[ \frac{-F_T \cos\alpha\cos\beta + \frac{1}{2}\rho V_T^2 S C_y^\beta}{mV_T} + \frac{(\cos\beta\cos\alpha\sin\theta - \sin\beta\sin\phi\cos\theta - \cos\beta\sin\alpha\cos\phi\cos\theta)g}{V_T} \right] \\ & + \dot{\phi}\frac{(\cos\beta\cos\phi\cos\theta + \sin\beta\sin\alpha\sin\phi\cos\theta)g}{V_T} \\ & + \dot{\theta}\frac{(\sin\beta\cos\alpha\cos\theta - \cos\beta\sin\phi\sin\theta + \sin\beta\sin\alpha\cos\phi\sin\theta)g}{V_T} \end{aligned} \tag{4.30}$$

From equation(4.12), (4.24) and(4.26), the new input **v** is

$$\begin{aligned} \mathbf{v} = & -\mathbf{G(x)}^{-1}\left[\mathbf{L_f G(x)}\right]\mathbf{G(x)}^{-1}\left[-\mathbf{H(x)} + \dot{\mathbf{z}}_c - \mathbf{K_p(z - z_c)} - \mathbf{K_i}\int(\mathbf{z - z_c})\mathbf{dt}\right] \\ & + \mathbf{G(x)}^{-1}\left[-\mathbf{L_f^2 H(x)} + \ddot{\mathbf{z}}_c - \mathbf{K_p}\left(\dot{\mathbf{z}} - \dot{\mathbf{z}}_c\right) - \mathbf{K_i(z - z_c)}\right] \\ & - \mathbf{K_i}\left\{\mathbf{y} - \mathbf{G(x)}^{-1}\left[-\mathbf{H(x)} + \dot{\mathbf{z}}_c - \mathbf{K_p(z - z_c)} - \mathbf{K_i}\int(\mathbf{z - z_c})\mathbf{dt}\right]\right\} \end{aligned} \tag{4.31}$$

The complete inner and outer loop design for the nonlinear aircraft is shown in Figure 4.12.

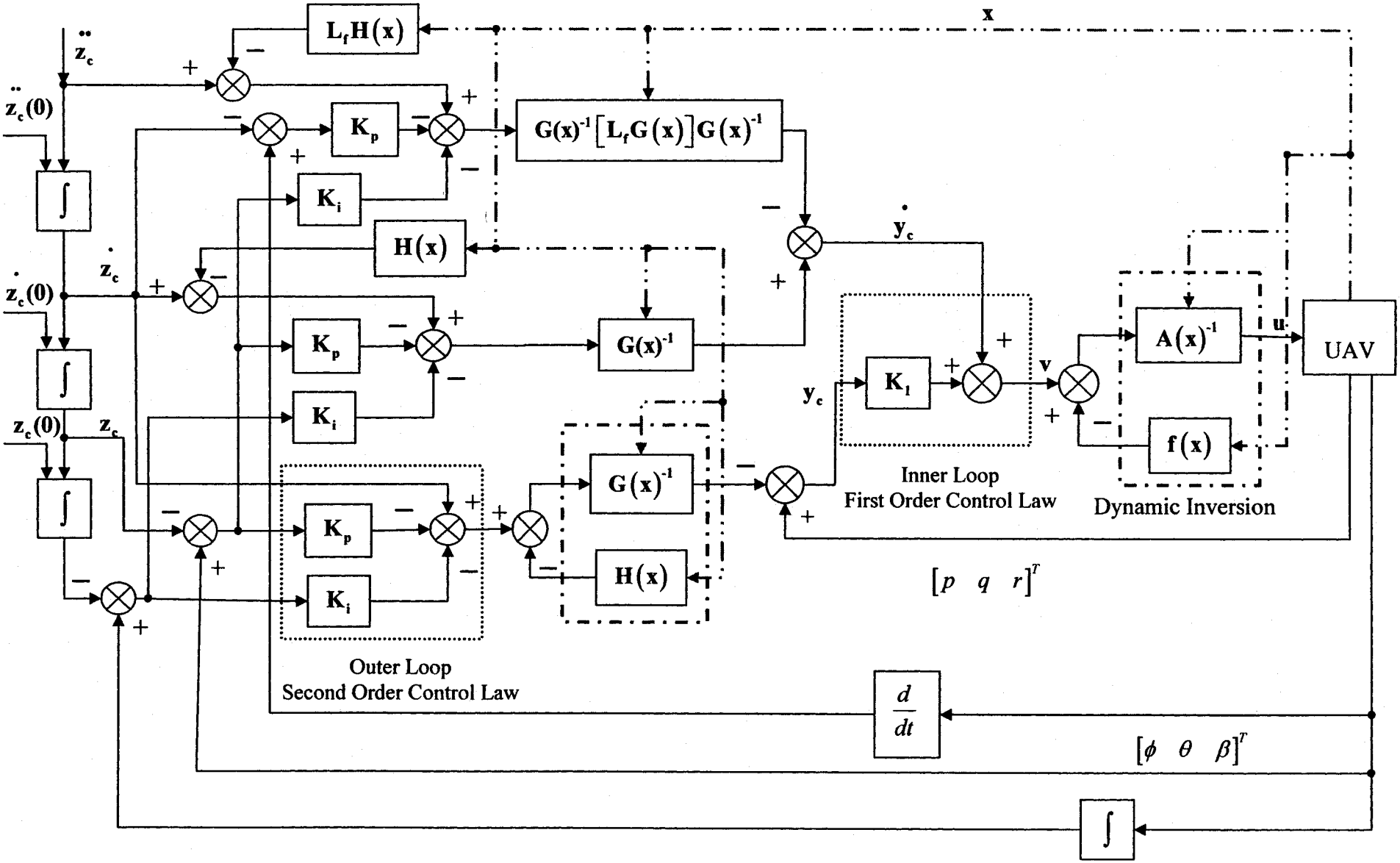



Figure 4.12 Inner / Outer Loop Dynamic Inversion Control

## 4.4 Model Predictive Control

Model Predictive Control (MPC) means that the current control decisions are made based on a prediction of the system controlled response for several time steps into the future. The current control input is computed as a linear combination of past input and past output measurements up to a finite number of time steps. The current control action is derived by minimizing the cost function of the difference between predictions of the system (controlled) output and the desired output for several time steps in the future.

### *4.4.1 System Response*

Consider an $n$-th order, $r$-input, $m$-output discrete-time model of a system in state space form

$$\begin{aligned} \mathbf{x}(k+1) &= \mathbf{A}\mathbf{x}(k) + \mathbf{B}\mathbf{u}(k) \\ \mathbf{y}(k) &= \mathbf{C}\mathbf{x}(k) + \mathbf{D}\mathbf{u}(k) \end{aligned} \tag{4.32}$$

where the dimension of $\mathbf{A} \in \mathbf{R}^{n \times n}, \mathbf{B} \in \mathbf{R}^{n \times r}, \mathbf{C} \in \mathbf{R}^{m \times n}$ and $\mathbf{D} \in \mathbf{R}^{m \times r}$.

After $M$ steps, the discrete time state space equation, solved iteratively, gives

$$\begin{aligned} \mathbf{x}(k+M+1) &= \mathbf{A}^{M+1}\mathbf{x}(k) + \mathbf{C}_{\mathbf{M}}\mathbf{u}_{\mathbf{M}}(k) \\ \mathbf{y}_{\mathbf{M}}(k) &= \mathbf{O}_{\mathbf{M}}\mathbf{x}(k) + \mathbf{T}_{\mathbf{M}}\mathbf{u}_{\mathbf{M}}(k) \end{aligned} \tag{4.33}$$

where $\mathbf{u}_{\mathbf{M}}(k)$ and $\mathbf{y}_{\mathbf{M}}(k)$ are vectors of input and output data after $M$ steps in the future starting at $k$.

$$\mathbf{U} = \mathbf{u}_{\mathbf{M}}(k) = \begin{bmatrix} \mathbf{u}(k) \\ \mathbf{u}(k+1) \\ \vdots \\ \mathbf{u}(k+M) \end{bmatrix}, \text{ and } \mathbf{Y} = \mathbf{y}_{\mathbf{M}}(k) = \begin{bmatrix} \mathbf{y}(k+1) \\ \mathbf{y}(k+2) \\ \vdots \\ \mathbf{y}(k+M) \end{bmatrix}$$

$\mathbf{C}_{\mathbf{M}}$ is an *Controllability matrix*, $\mathbf{O}_{\mathbf{M}}$ is an *Observability matrix* and $\mathbf{T}_{\mathbf{M}}$ is an *Toeplitz matrix* of the system Markov parameters.

$$\mathbf{C}_{\mathbf{M}} = \left[\mathbf{A}^{M}\mathbf{B}, \cdots, \mathbf{AB}, \mathbf{B}\right], \ \mathbf{O}_{\mathbf{M}} = \begin{bmatrix} \mathbf{CA} \\ \vdots \\ \mathbf{CA}^{M} \end{bmatrix}, \ \mathbf{T}_{\mathbf{M}} = \begin{bmatrix} \mathbf{CB} & \mathbf{D} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{CAB} & \mathbf{CB} & \mathbf{D} & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \mathbf{0} \\ \mathbf{CA}^{M-1}\mathbf{B} & \cdots & \mathbf{CAB} & \mathbf{CB} & \mathbf{D} \end{bmatrix}$$

From equation(4.33), the future output can be calculated as the function of the actual state vector $\mathbf{x}(k)$ and the future control input signals $\mathbf{u}_{\mathbf{M}}(k)$.

### *4.4.2 Model Predictive Control*

The strategy of predictive control is based on a minimization of a quadratic cost function which involves a prediction of system and future control signals.

$$\begin{aligned} J_k &= \sum_{i=1}^{M} (\mathbf{w}_{k+i} - \mathbf{y}_{k+i})^T Q_i (\mathbf{w}_{k+i} - \mathbf{y}_{k+i}) + \sum_{i=0}^{N} \mathbf{u}_{k+i}{}^T R_i \mathbf{u}_{k+i} \\ &= (\mathbf{W} - \mathbf{Y})^T \hat{\mathbf{Q}} (\mathbf{W} - \mathbf{Y}) + \mathbf{U}^T \hat{\mathbf{R}} \mathbf{U} \end{aligned} \tag{4.34}$$

Where

$$\hat{\mathbf{Q}} = \begin{bmatrix} \mathbf{Q}_1 & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{Q}_2 & & \vdots \\ \vdots & & \mathbf{Q}_i & \mathbf{0} \\ \mathbf{0} & \cdots & \mathbf{0} & \mathbf{Q}_M \end{bmatrix} \qquad \hat{\mathbf{R}} = \begin{bmatrix} \mathbf{R}_0 & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{R}_1 & & \vdots \\ \vdots & & \mathbf{R}_i & \mathbf{0} \\ \mathbf{0} & \cdots & \mathbf{0} & \mathbf{R}_N \end{bmatrix}$$

$$\mathbf{W} = \mathbf{w}_{\mathbf{M}}(k) = \begin{bmatrix} \mathbf{w}(k+1) \\ \mathbf{w}(k+2) \\ \vdots \\ \mathbf{w}(k+M) \end{bmatrix} \qquad \mathbf{Y} = \mathbf{y}_{\mathbf{M}}(k) = \begin{bmatrix} \mathbf{y}(k+1) \\ \mathbf{y}(k+2) \\ \vdots \\ \mathbf{y}(k+M) \end{bmatrix}$$

$\mathbf{W}$ is a future reference trajectory vector and $\hat{\mathbf{Q}}$ and $\hat{\mathbf{R}}$ are nonnegative diagonal weighting matrices in the optimizing horizon $M$ and control horizon $N$.

Assuming that a constant control signal is applied within the optimizing horizon and that there are no direct input for the output (i.e. matrix $\mathbf{D} = [\mathbf{0}]$, the system response can be expressed as follow. (Berlin F. and Frank P.M. 1994)

$$\begin{aligned} \mathbf{y}_{\mathbf{M}}(k) &= \mathbf{T}\mathbf{x}(k) + \mathbf{S}\mathbf{u}_{\mathbf{M}}(k) \\ \mathbf{Y} &= \mathbf{T}\mathbf{x}(k) + \mathbf{S}\mathbf{U} \end{aligned} \tag{4.35}$$

where

$$\mathbf{T} = \begin{bmatrix} \mathbf{CA} \\ \mathbf{CA}^2 \\ \vdots \\ \mathbf{CA}^M \end{bmatrix} \qquad \mathbf{S} = \begin{bmatrix} \mathbf{CB} \\ \mathbf{CAB} + \mathbf{CB} \\ \vdots \\ \mathbf{CA}^{M-1}\mathbf{B} + \mathbf{CA}^{M-2}\mathbf{B} + \cdots + \mathbf{CB} \end{bmatrix}$$

The cost function is constructed by two parts current state vector and future control input vector. The optimal control solution of equation (4.34) is obtained for

$$\frac{\partial J}{\partial u} = 0 \tag{4.36}$$

and the optimal control vector

$$\mathbf{U} = (\hat{\mathbf{R}} + \mathbf{S}^T \hat{\mathbf{Q}} \mathbf{S})^{-1} \mathbf{S}^T \hat{\mathbf{Q}} (\mathbf{W} - \mathbf{T}\mathbf{x}_k) \tag{4.37}$$

Only the first control vector is applied to the system,

$$\mathbf{u}_k = \begin{bmatrix} \mathbf{I}_{r\times r} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & & \vdots \\ \vdots & & \ddots & \mathbf{0} \\ \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} \end{bmatrix} (\hat{\mathbf{R}} + \mathbf{S}^T \hat{\mathbf{Q}} \mathbf{S})^{-1} \mathbf{S}^T \hat{\mathbf{Q}} (\mathbf{W} - \mathbf{T}\mathbf{x}_k) \tag{4.38}$$

According the approach of modified Dynamic Matrix Control (DMC) design method, the optimizing horizon and the control horizon have same length and the weighting control matrix is neglected in the quadratic cost function. The optimal control vector

$$\begin{aligned} \mathbf{u}_k &= (\mathbf{S}^T \hat{\mathbf{Q}} \mathbf{S})^{-1} \mathbf{S}^T \hat{\mathbf{Q}} (\mathbf{W} - \mathbf{T}\mathbf{x}_k) \\ &= \underbrace{(\mathbf{S}^T \hat{\mathbf{Q}} \mathbf{S})^{-1} \mathbf{S}^T \hat{\mathbf{Q}}}_{G} \mathbf{W} - \underbrace{(\mathbf{S}^T \hat{\mathbf{Q}} \mathbf{S})^{-1} \mathbf{S}^T \hat{\mathbf{Q}} \mathbf{T}}_{GT} \mathbf{x}_k \end{aligned} \tag{4.39}$$

The DMC control law consists of a state feedback part and feedforward part which is shown in Figure 4.13.

### *4.4.3 MPC for Inner Loop*

From equation(4.15), the inner loop model after applying feedback linearization can be expressed in the continuous state space form as follows

$$\begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix} + \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} \tag{4.40}$$

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix}$$

This system can be expressed in the discrete time with different sampling time and the optimal control vector can be calculated on the basis of equation (4.37) and(4.39).

The MPC block diagram for inner loop control is shown in Figure 4.13.

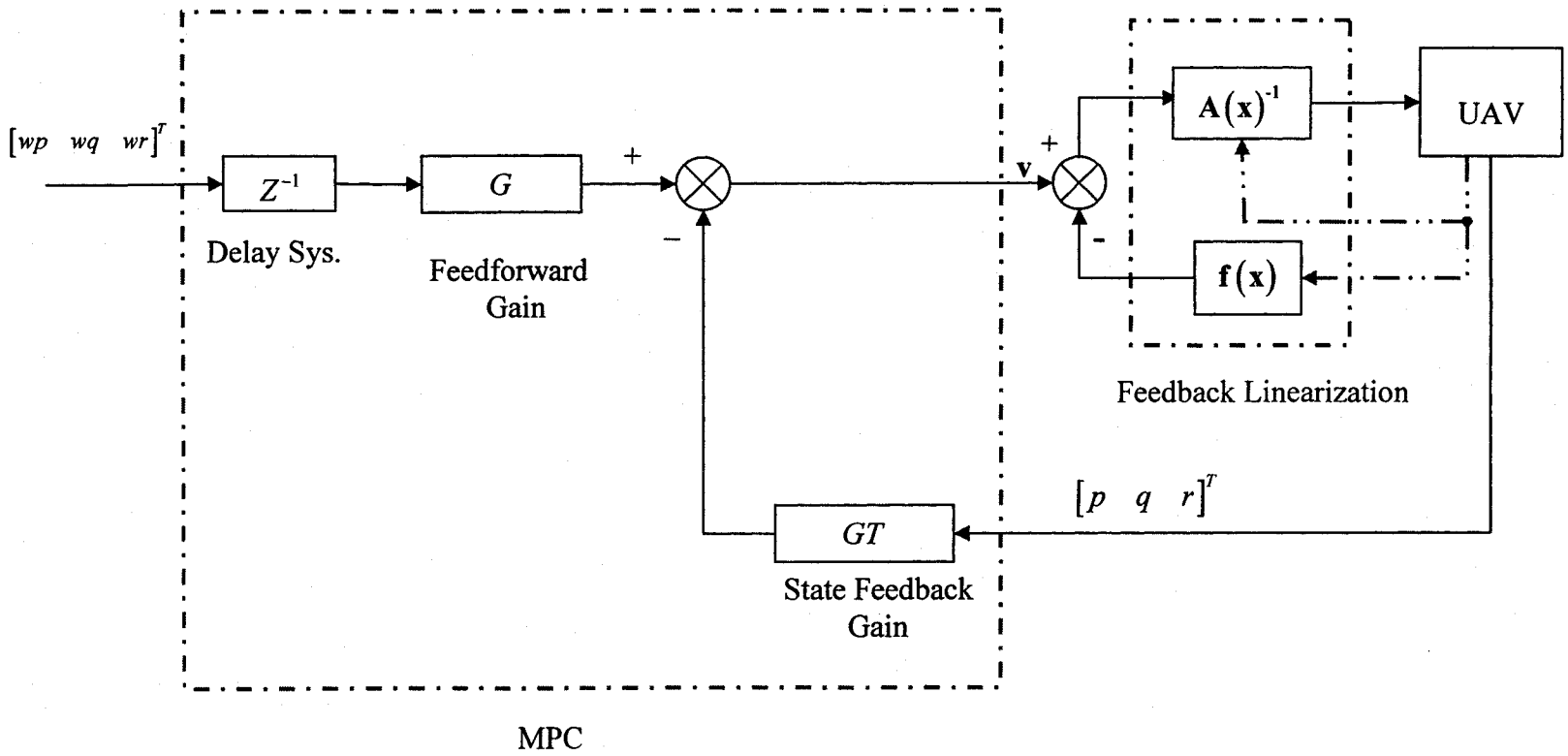



Figure 4.13 Model Predictive Control for Inner Loop

# Chapter 5

# 5 SIMULATION PROGRAM AND RESULTS

## 5.1 Simulator Programming

The simulator is programmed using Simulink® and the motion of UAV is displayed with the help of the Virtual Reality Toolbox in the 3D environment.

The simulation program is divided into several subsystems which are connected through relationships derived in chapter 3 and 4. These main subsystems are introduced as follows:

1. UAV nonlinear model

Following the structure used in the Aerosim product, the equations of nonlinear UAV model are modified according to the chapter 3, which includes force equations, moment equations, kinematic equations and navigation equation. The inputs of nonlinear UAV subsystem are three control surface deflections and the thrust and the outputs are twelve states and $\dot{\alpha}$ which are used for inner and outer loop dynamic inversion control synthesis. The parameters of UAV are generated by the configuration file. The necessary simulation data are saved in the data file and the variables in the workspace.

2. Inner and Outer Dynamic Inversion Control Synthesis

The synthesis of inner and outer loop is programmed as S functions. In order to reuse these programs for future research and for real time simulation, all the S functions are written in C which can be converted to the real time executable code by the Real-time Workshop directly. The configuration of real time simulation is proposed in appendix B. The program of the S function is based on the equations derived in chapter 4.

3. Model Predictive Control

The linearized system is discretized by the Matlab program with specified sampling period, and all parameters, such as feedforward gain, delaying system, and statefeedback gain, are computed in this M file. Before running the Simulink program of the model predictive control, the optimization horizon and sampling period have to be defined.

4. Virtual Reality

It is very easy to validate the operation of UAV in the virtual form during the simulation. During the simulation, the orientation and position of UAV is displayed in the 3D environment, which is programmed by the Virtual Reality Modeling Language (VRML). The orientation and position data are transferred to the 3D environment by the Virtual Reality Toolbox in real time. The 3D environment is modified from vrtkoff example.

The above-mentioned subsystem is interconnected and shown in figure Figure 5.1

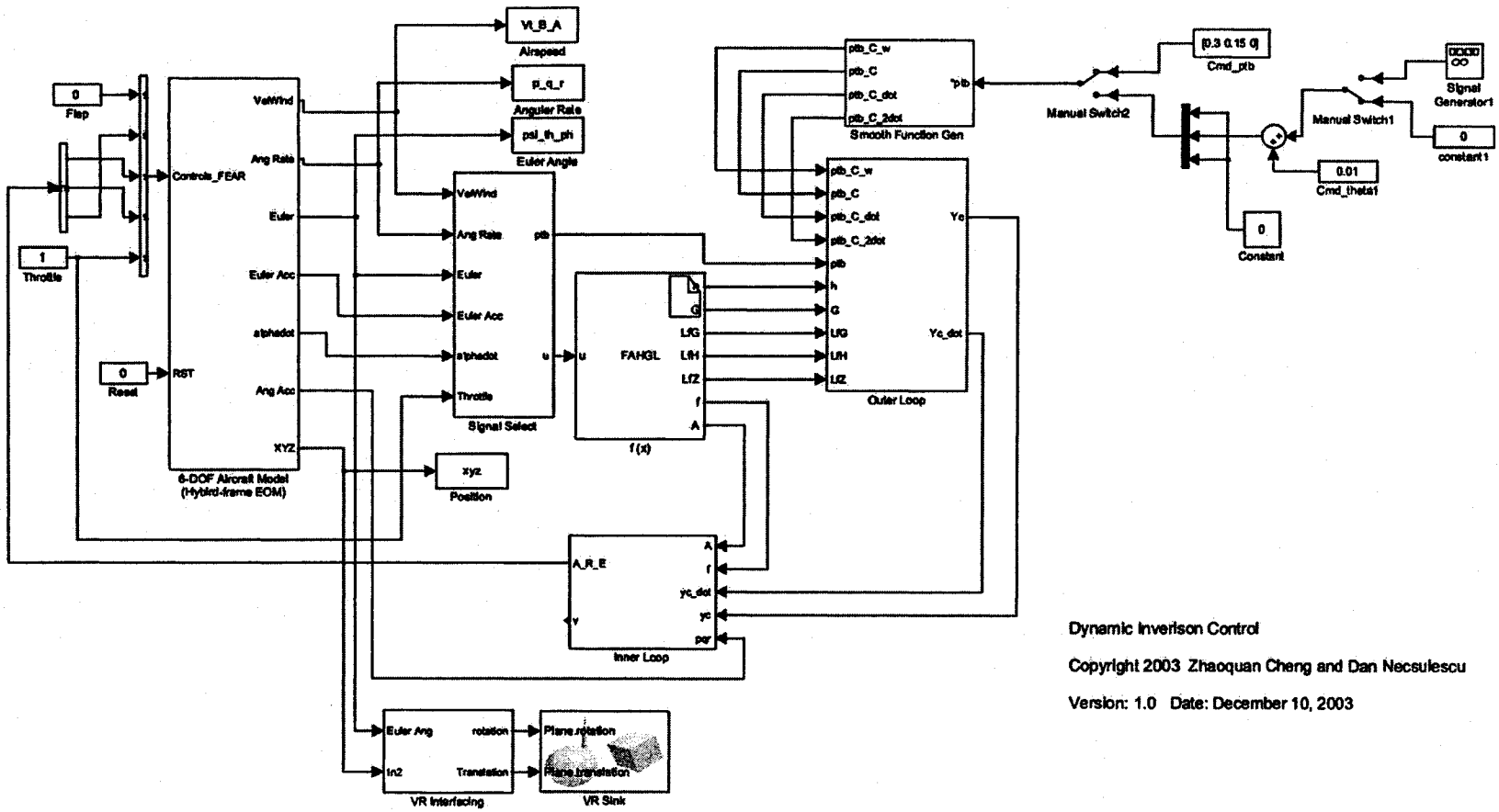



Figure 5.1Inner / Outer Dynamic Inversion Simulink Progarm

## 5.2 UAV Open Loop Response

To test the natural characteristics, the UAV model is first tested in the open loop form. The three control surface deflections are set as zero and the thrust is a constant. All the initial conditions of each state are configured as follows:

| | | | |
|---|---|---|---|
| $V_T \quad \beta \quad \alpha$ | 25 0 0 | $p \quad q \quad r$ | 0 0 0 |
| $\psi \quad \theta \quad \phi$ | 0 0 0 | $x \quad y \quad z$ | 0 0 0 |

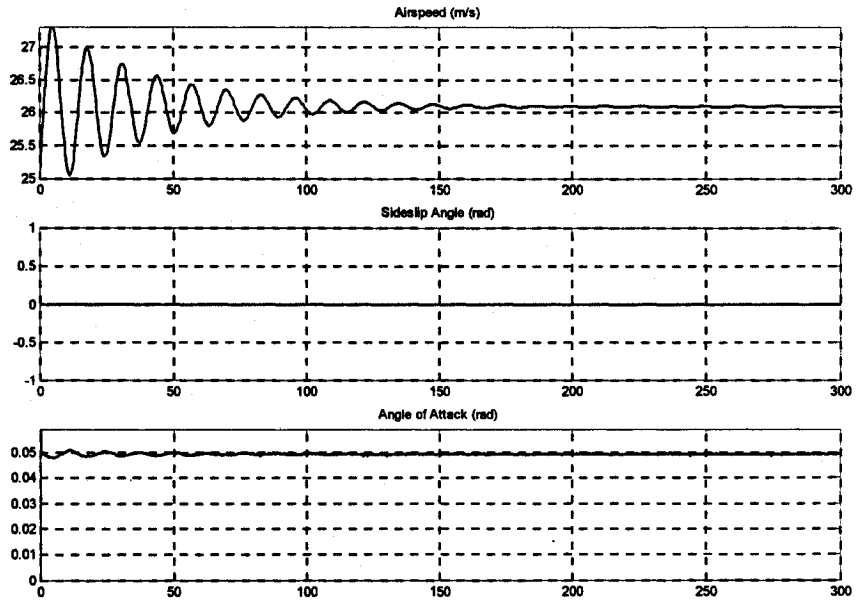



Figure 5.2 Airspeed and Aerodynamic Angles Response (Open Loop)

In Figure 5.2, the airspeed oscillation is damped out at 150 second, the sideslip angle is zero, and the angle of attack is a nonzero constant and causes a certain aerodynamic force to hold UAV in the steady state.

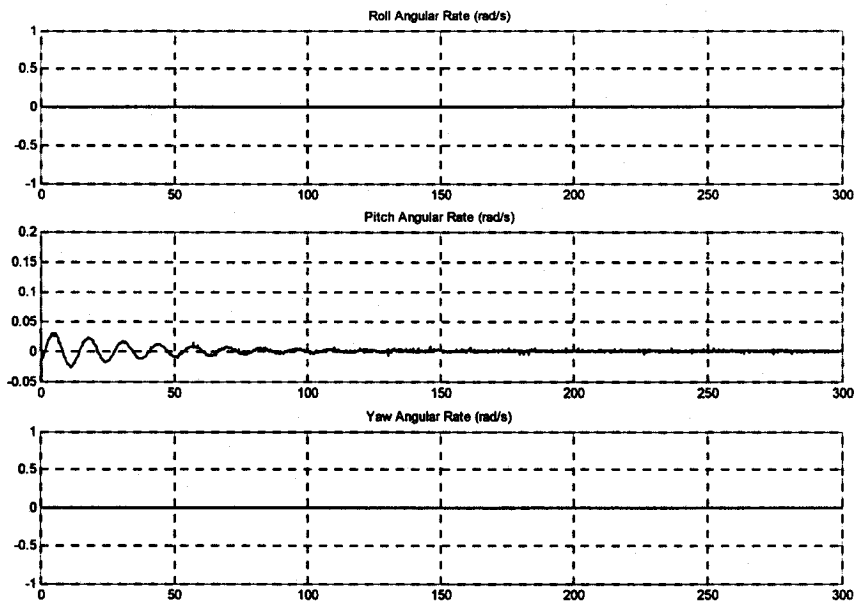



Figure 5.3 Angular Rate Response (Open Loop)

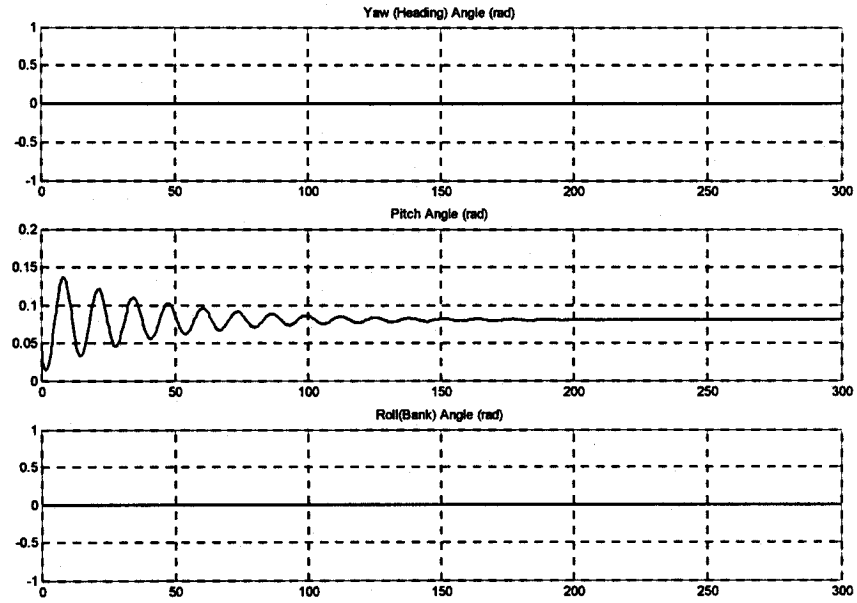



Figure 5.4 Euler Angles Response (Open Loop)

The oscillation of the pitch angle is damped out along with the airspeed, the steady state value of pitch angle is about 0.08 rad and all angular rates are settled at zero.

In this case, the UAV is stable for the open loop form and climbs at constant airspeed, 26 m/s, with constant pitch angle in the steady state. The 3D trajectory of UAV in the fixed earth frame is shown in Figure 5.5 and permits to verify the climbing operation.

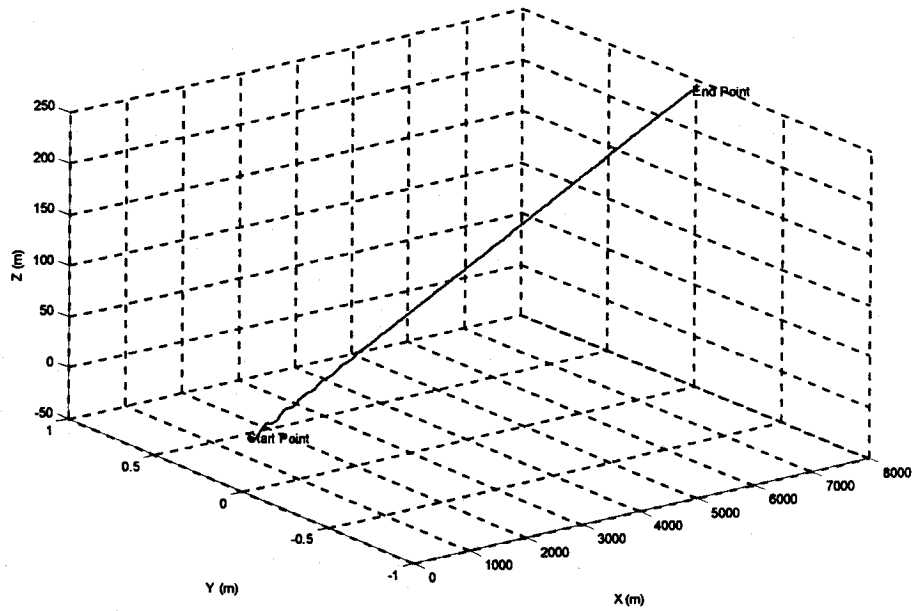



Figure 5.5 UAV 3D Trajectory in Earth Fixed Frame (Open Loop)

The results confirm the expected performance of the open loop UAV model.

## 5.3 UAV Autopilot Control with Classical Controller

From the open loop response, the UAV can't achieve satisfiactory maneuver due to long settling times. Classical controller configurations were presented in chapter 4. The pitch angle is controlled by the deflection of elevator with a PID controller and the parameters of the PID control are obtained by trial and error approach.

To analyse the performance of classical control, all parameters of UAV's configuration and initial conditions of states are set the same as for open loop control.

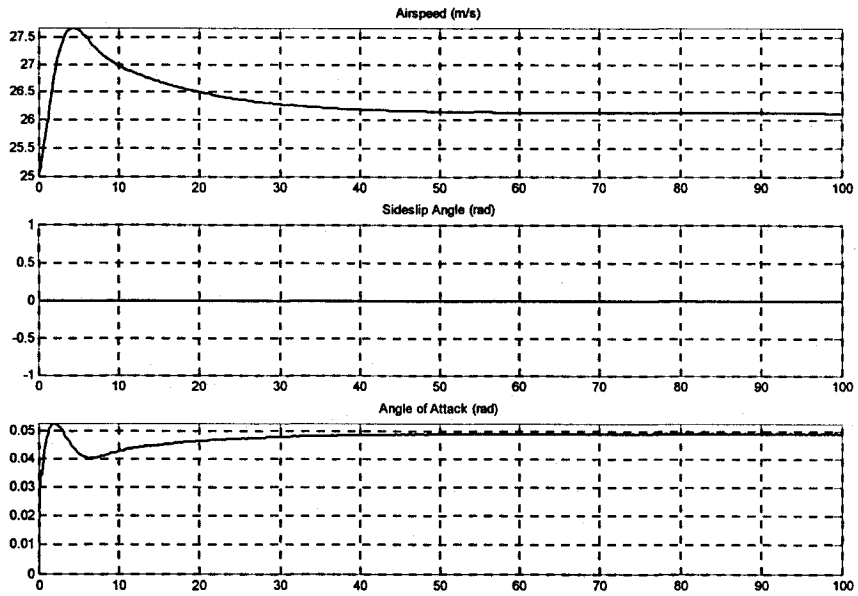



Figure 5.6 Airspeed and Aerodynamic Angles Response (Classical Controller)

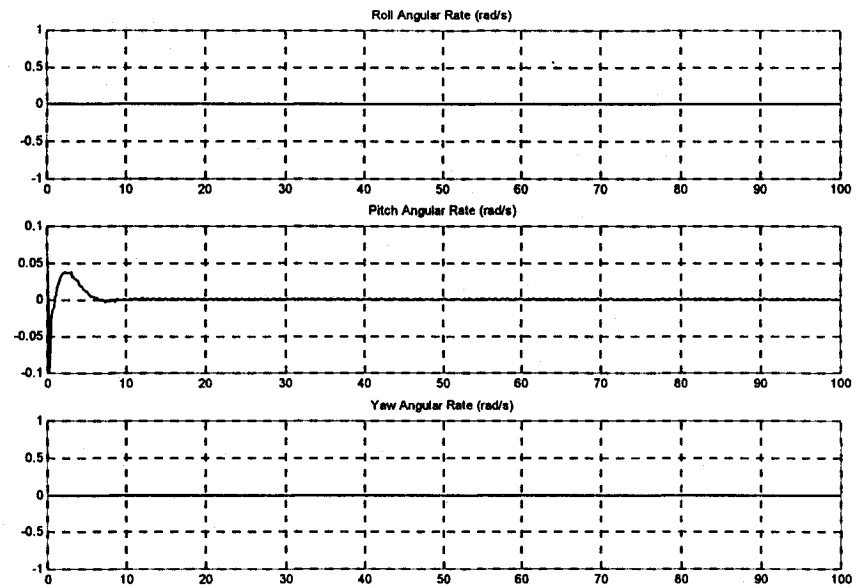



Figure 5.7 Angular Rate Response (Classical Controller)

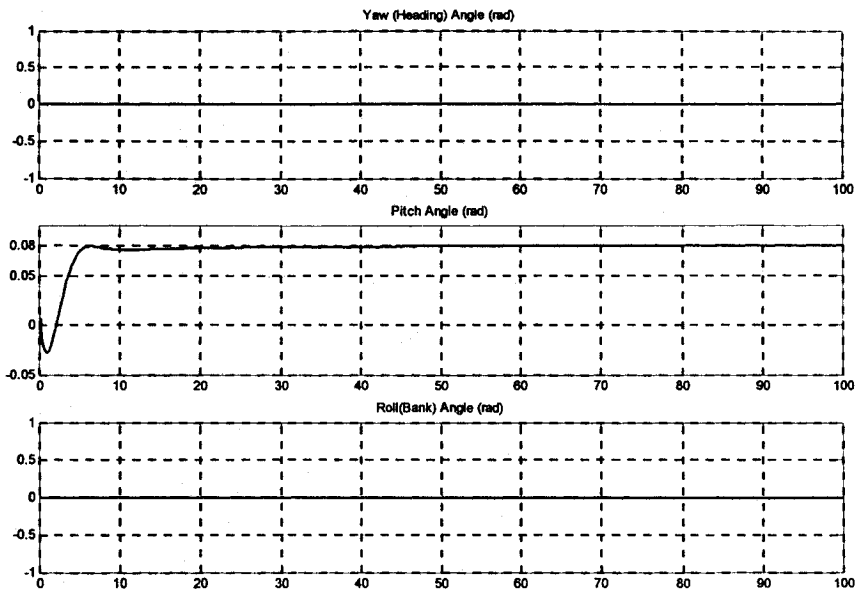



Figure 5.8 Euler Angles Response (Classical Controller)

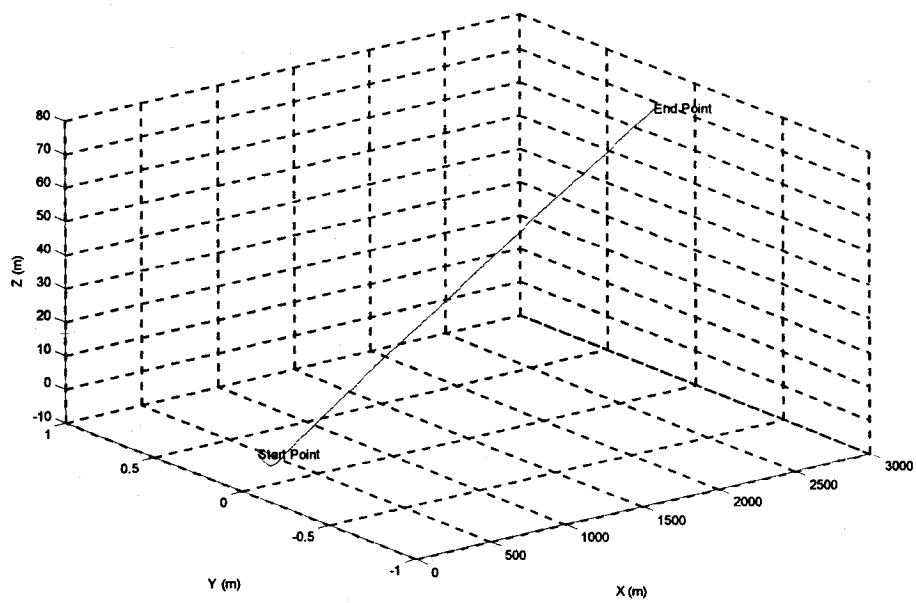



Figure 5.9 UAV 3D Trajectory in Earth Fixed Frame (Classical Controller)

Compared with open loop response, the classical control gives a faster settling time (25 sec) which differs significantly from the longer open loop transients. The airspeed is 26 m/s in the steady state. The parameters of PID controller are chosen to achieve the

desired pitch angle without overshoot and no steady state error. The trajectory of UAV is shown in Figure 5.9

To complete various missions, UAV operate at different pitch angles. With the same PID parameters' setting, the desired pitch angle is reset as $\theta = 0.16$ rad. The responses are shown in Figure 5.10, 5.11 and 5.12.

Since UAV climbs with a large pitch angle, the steady airspeed is reduced to 21 m/s and the angle of attack settles at 0.1 rad to get enough aerodynamic force. From the pitch angle's response, it is found that UAV takes 40sec to obtain the desired values and that the transient response is different from previous setting.

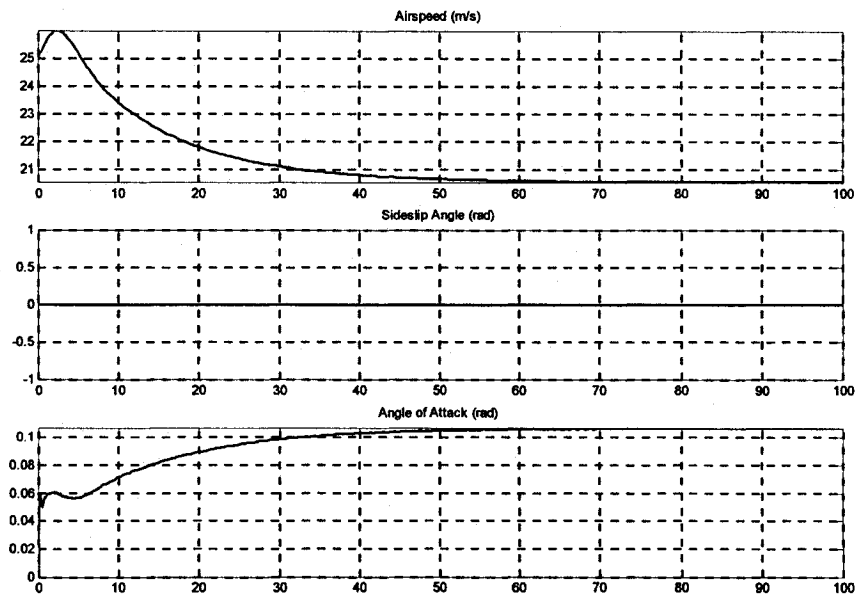



Figure 5.10 Airspeed and Aerodynamic Angles Response
(Classical Controller $\theta = 0.16$ rad)

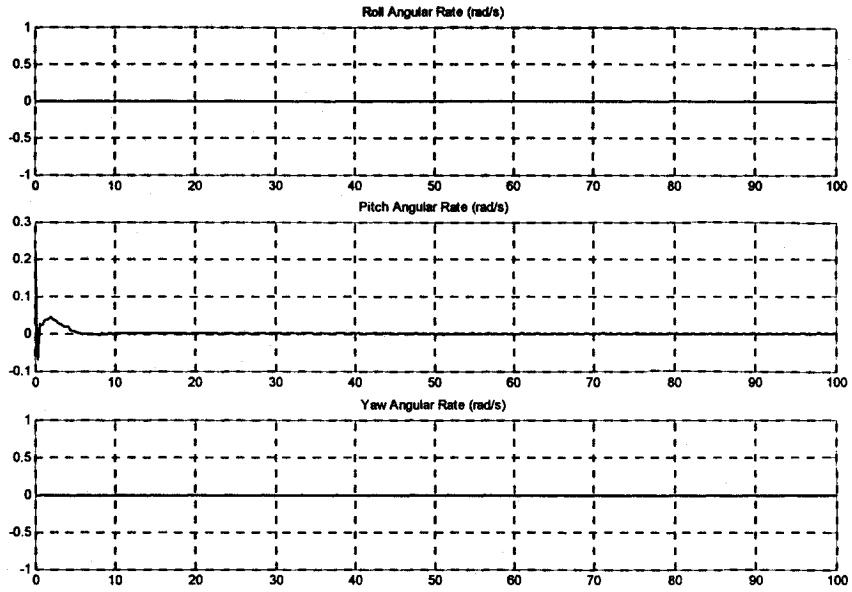



Figure 5.11 Angular Rate Response (Classical Controller $\theta = 0.16$ rad)

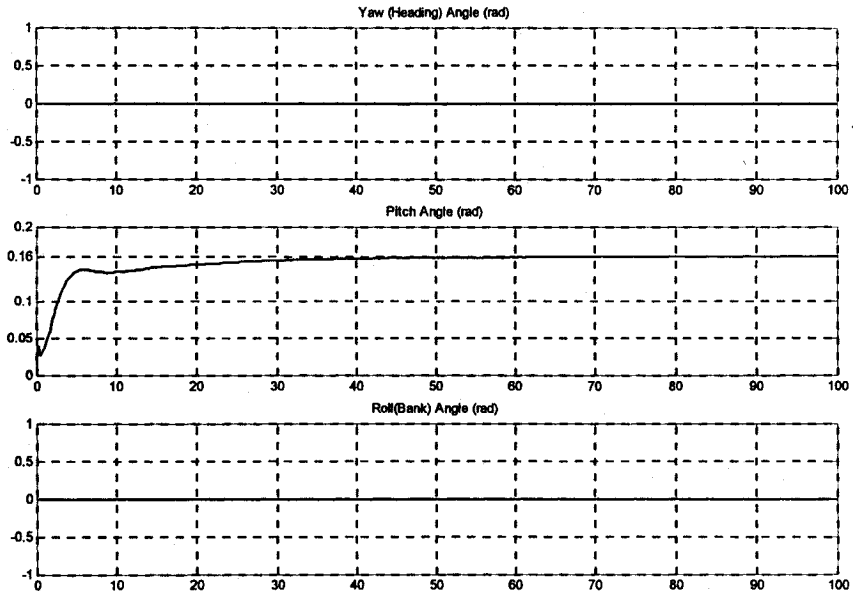



Figure 5.12 Euler Angles Response (Classical Controller $\theta = 0.16$ rad)

Furthermore, the UAV flies at the smaller pitch angle, $\theta = 0.01$ rad. The simulation results are shown in Figure 5.13, 5.14 and 5.15.

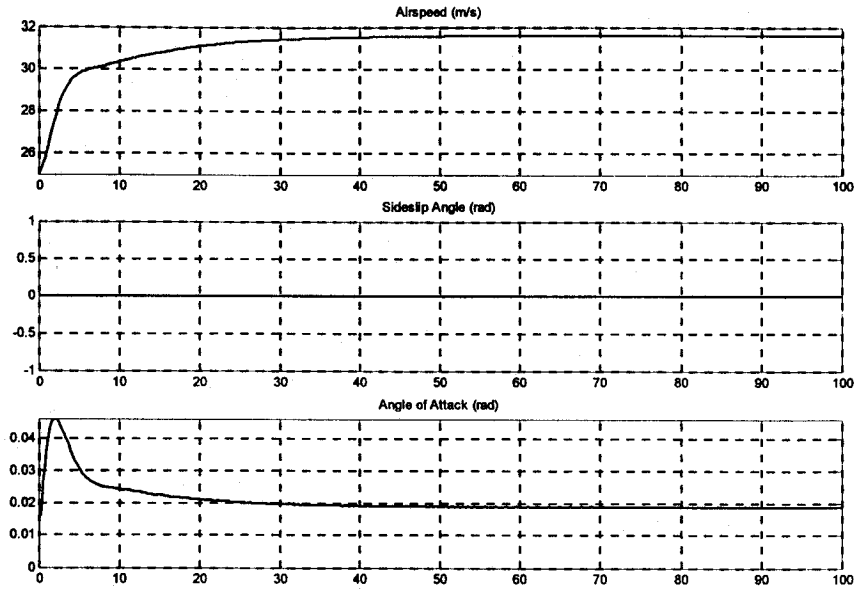



Figure 5.13 Airspeed and Aerodynamic Angles Response (Classical Controller $\theta = 0.01\,\text{rad}$) .

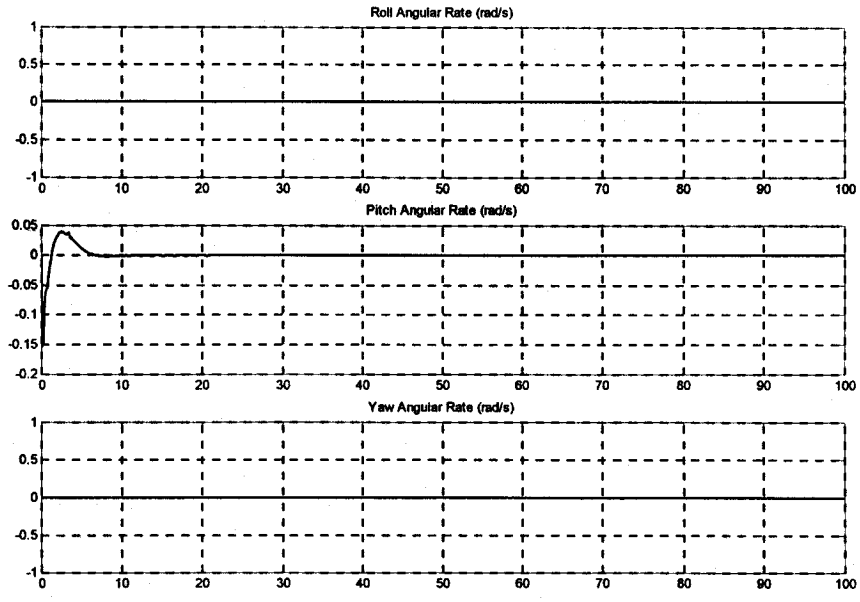



Figure 5.14 Angular Rate Response (Classical Controller $\theta = 0.01\,\text{rad}$)

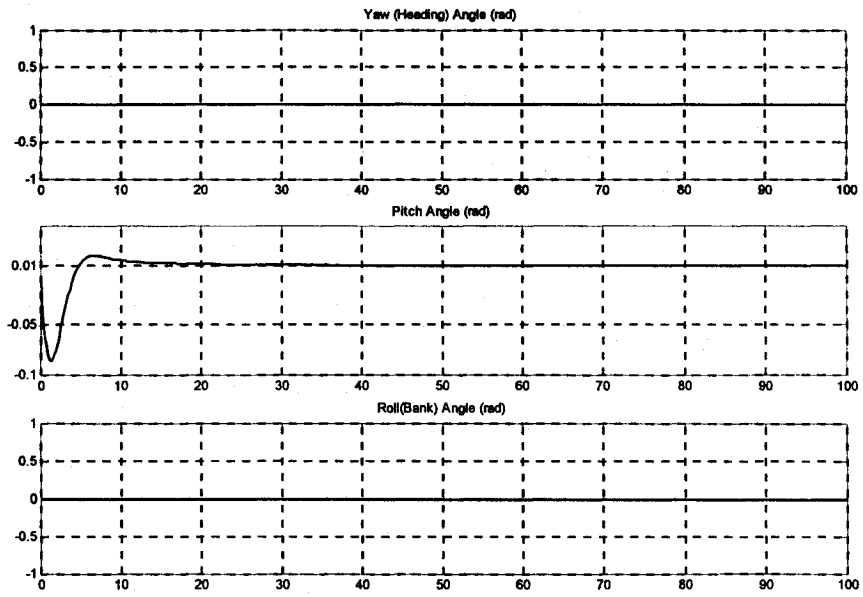



Figure 5.15 Euler Angles Response (Classical Controller $\theta = 0.01$ rad)

The airspeed settles at 32 m/s due to the smaller climbing angle and the steady angle of attack becomes smaller as well. From pitch angle response, although the UAV achieves a short settling time, overshot is presented in the transit response. The maximum pitch angle is about 0.02 rad.

It is observed that the transient response is different when the desired value is away from the controller design point. The unexpected transient response is not acceptable for the high performance UAV and for collision avoidance system design. To meet the performance requirements, the controllers have to be designed under different trim control conditions.

This results show the transient performance limitations when a nonlinear UAV is controller by linear controllers and justify the investigation of nonlinear controller.

## 5.4 Inner and Outer Loop Dynamic Inversion Controller

The inner and outer loop Dynamic Inversion controller, derived in chapter 4, is simulated in this section. The configuration of the UAV is same as for conventional controller design and the simulation results of three different desired pitch angles are shown as follows.

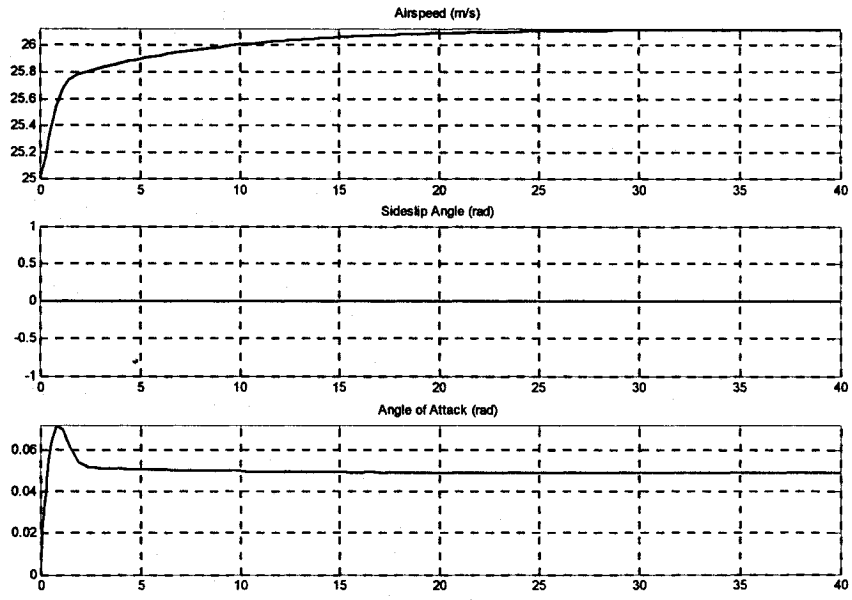



Figure 5.16 Airspeed and Aerodynamic Angles Response
(Inner and Outer Loop Dynamic Inversion Controller)

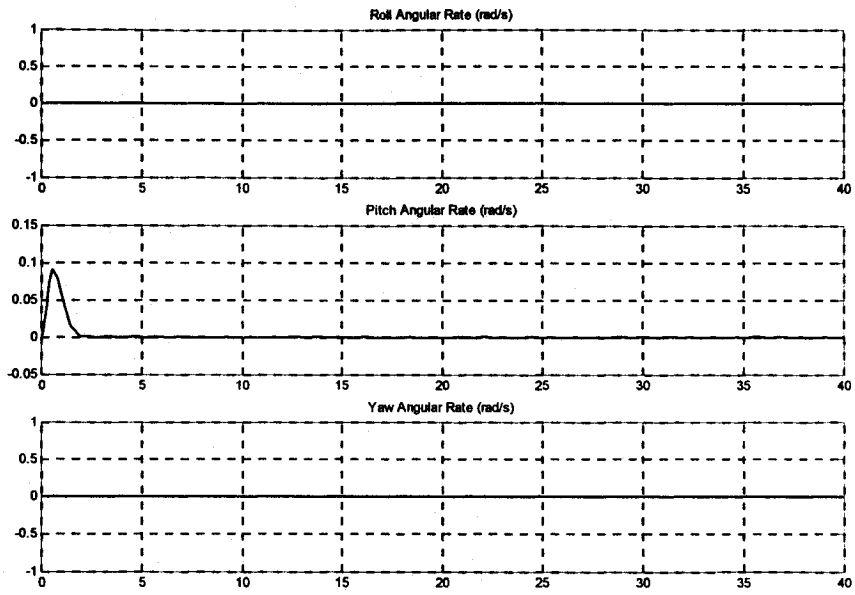



Figure 5.17 Angular Rate Response
(Inner and Outer Loop Dynamic Inversion Controller)

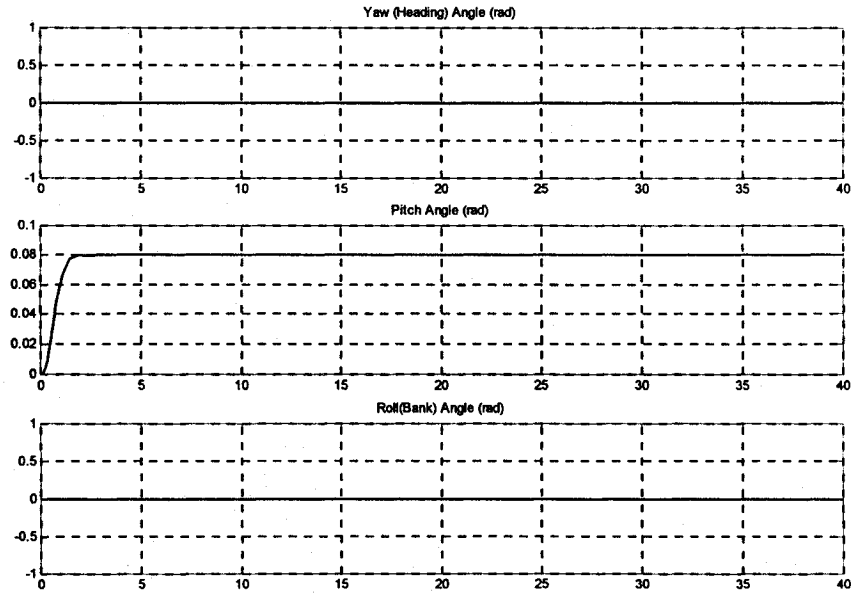



Figure 5.18 Euler Angle Response (Inner and Outer Loop Dynamic Inversion Controller)

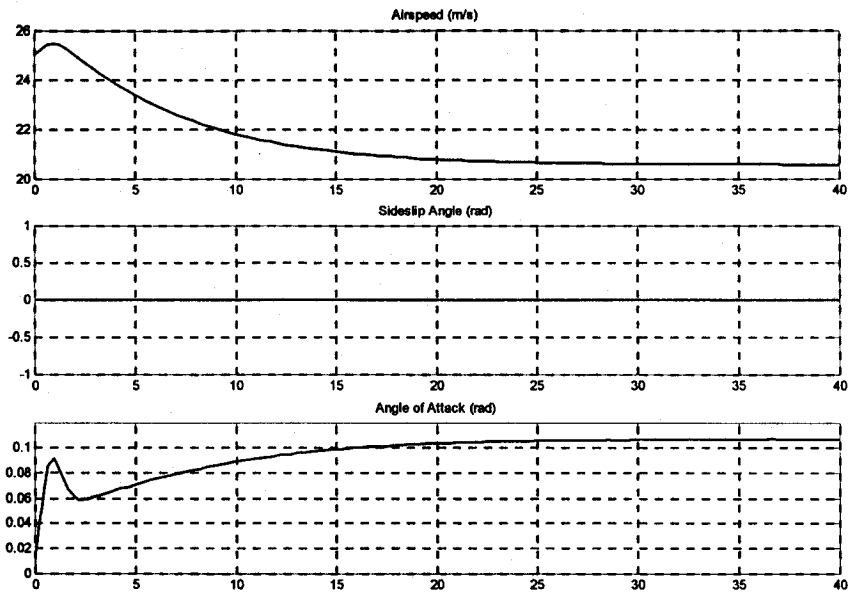



Figure 5.19 Airspeed and Aerodynamic Response
(Inner and Outer Loop Dynamic Inversion Controller $\theta = 0.16$ rad)

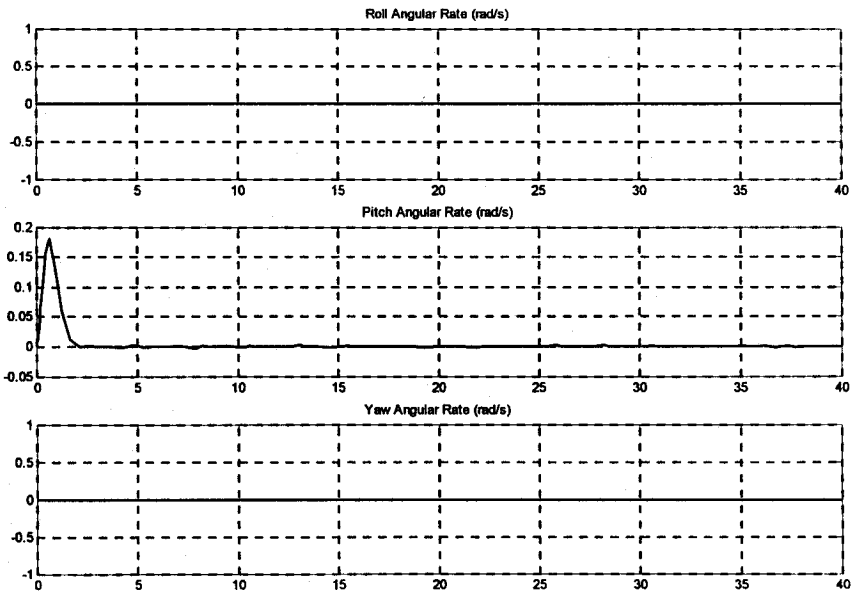



Figure 5.20 Angular Rate Response
(Inner and Outer Loop Dynamic Inversion Controller $\theta = 0.16$ rad)

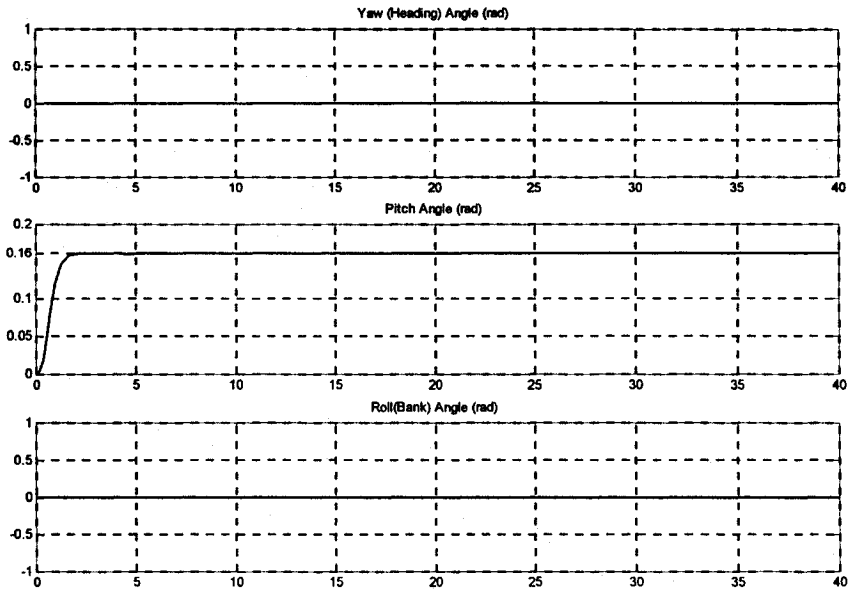



Figure 5.21 Euler Angles Response
(Inner and Outer Loop Dynamic Inversion Controller $\theta = 0.16$ rad)

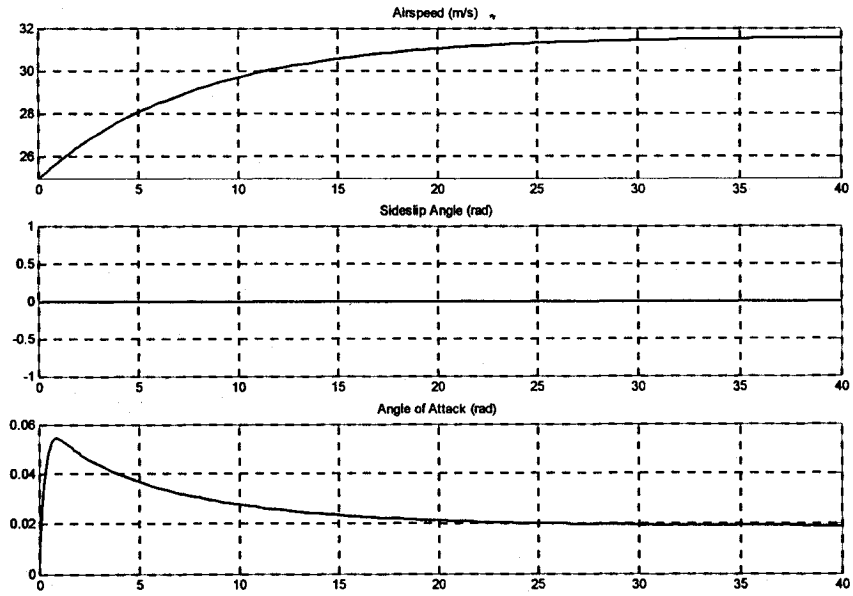



Figure 5.22 Airspeed and Aerodynamic Response
(Inner and Outer Loop Dynamic Inversion Controller $\theta = 0.01$ rad)

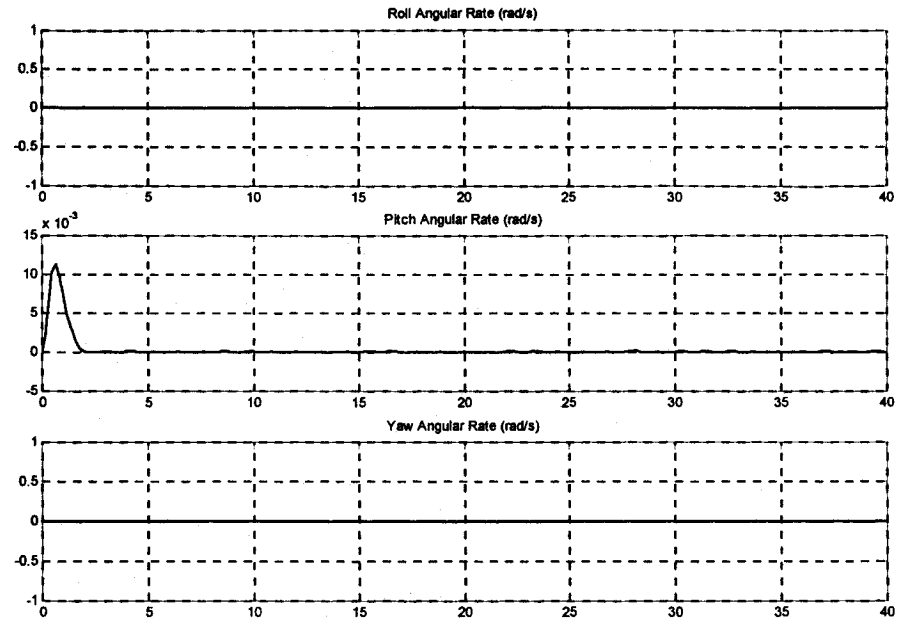



Figure 5.23 Angular Rate Response
(Inner and Outer Loop Dynamic Inversion Controller $\theta = 0.01\,\text{rad}$)

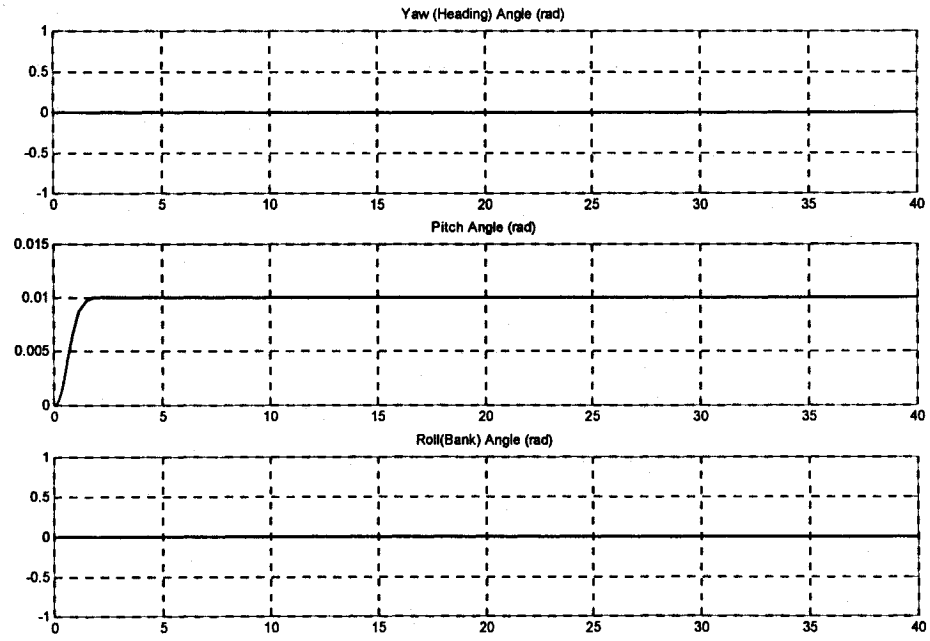



Figure 5.24 Euler Angles Response
(Inner and Outer Loop Dynamic Inversion Controller $\theta = 0.01\,\text{rad}$)

For inner and outer loop dynamic inversion controller, UAV converges to the desired value with fast settling time no overshot and no steady state error, which was not the case with classical control. For the same controller parameters, the transient response of different desired values is same. Therefore, it not necessary to redesign the controller under different trim conditions and the controller design work is significantly reduced.

Furthermore, compared with classical controller, the inner and outer dynamic inversion controller can be used to control multiple variables simultaneously. The desired values are set as $\phi = 0.2 \quad \theta = 0.1 \quad \beta = 0$ rad. The results are shown as follows.

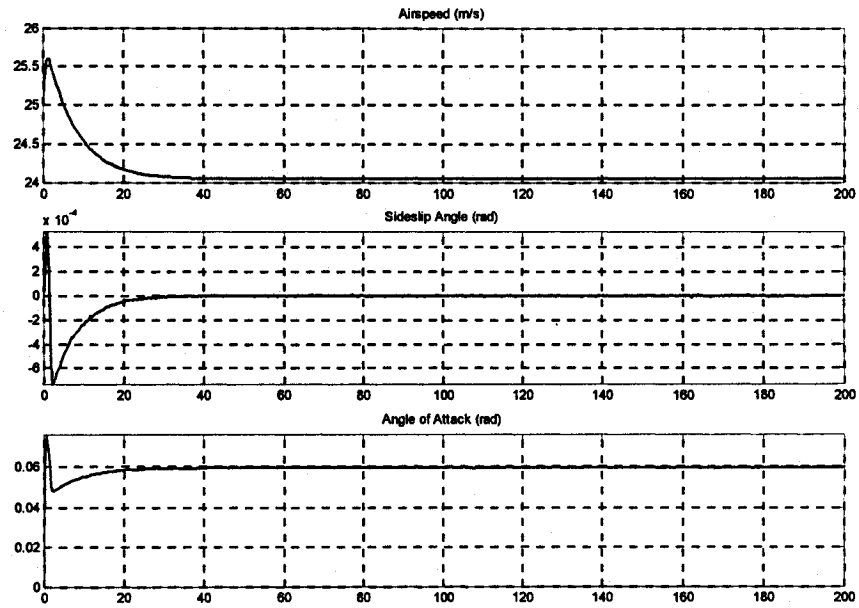



Figure 5.25 Airspeed and Aerodynamic Angles
(Inner and Outer Loop Dynamic Inversion Controller $\phi = 0.2 \quad \theta = 0.1 \quad \beta = 0$)

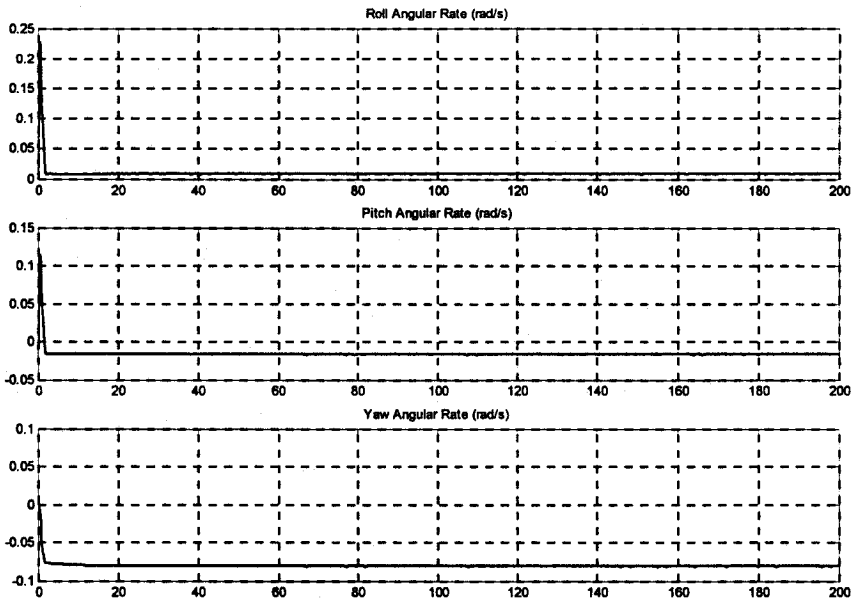



Figure 5.26 Angular Rate Response
(Inner and Outer Loop Dynamic Inversion Controller $\phi = 0.2 \quad \theta = 0.1 \quad \beta = 0$)

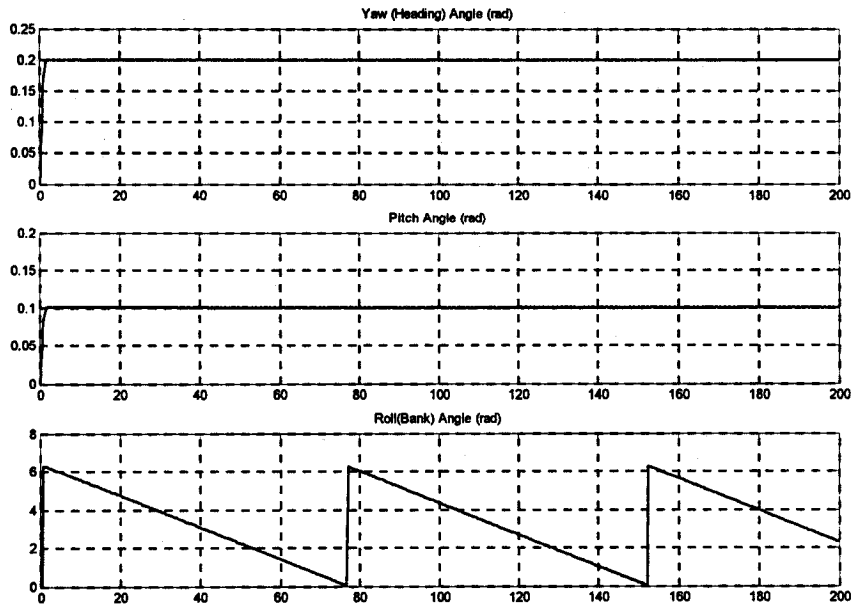



Figure 5.27 Euler Angles Response
(Inner and Outer Loop Dynamic Inversion Controller $\phi = 0.2 \quad \theta = 0.1 \quad \beta = 0$)

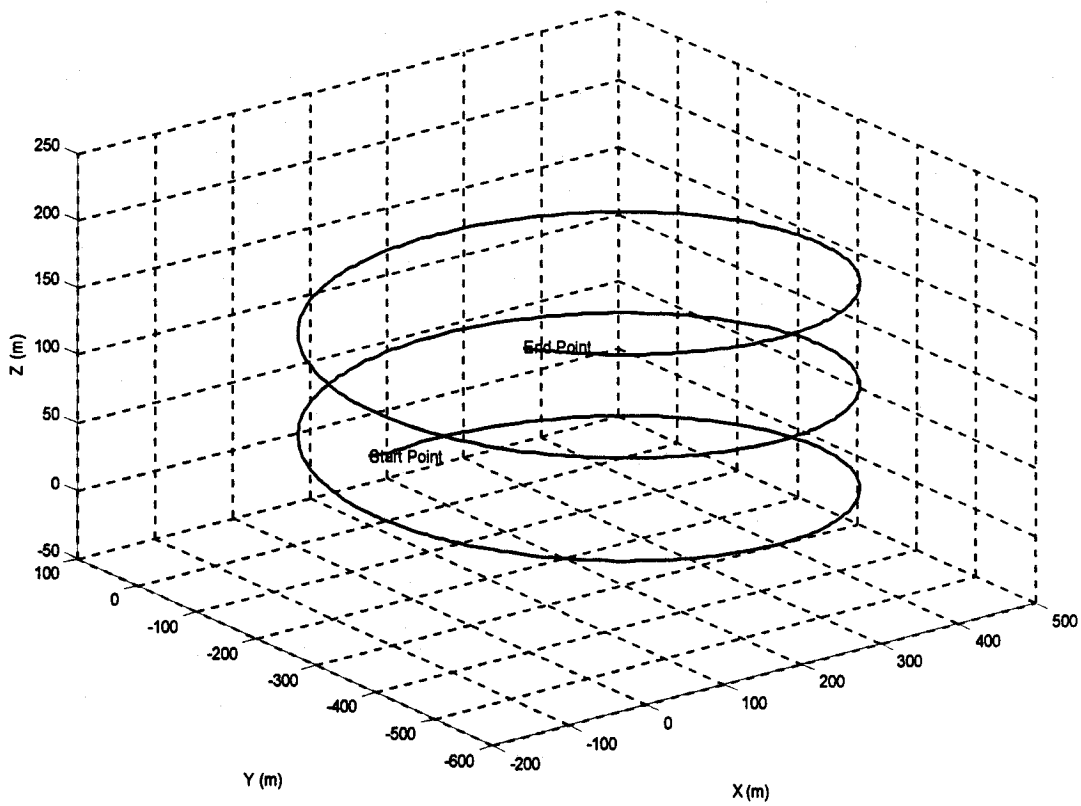



Figure 5.28 UAV 3D Trajectory in Earth Fixed Frame
(Inner and Outer Loop Dynamic Inversion Controller $\phi = 0.2 \quad \theta = 0.1 \quad \beta = 0$)

It is observed that the roll angle ($\phi$) tends towards to the command control value 0.2 rad, pitch angle ($\theta$) to 0.1 rad and yaw angle changed in the constant speed in the steady state from 0 rad to $2\pi$. The steady state value of siderslip angle ($\beta$) converged to the desired setting value 0 rad. Through the response to the desired variable, it is proved that the Inner and Outer loop control synthesis is valid for simultaneous multiple variables control.

Under the command input for the outer loop, the UAV trajectory in the fixed coordinate system is shown in Figure 5.28.

From the response of UAV trajectory, the UAV flies in the hover model which is used at take-off to get the certain height. The radius of flying trajectory is determined by the yaw angle. Furthermore, we observe that the radius is constant which confirms that the yaw angle is changing at constant speed from Figure 5.16 is valid.

## 5.5 Model Predictive Control

The simulation of Model Predictive Control for the inner loop is programmed in a separated file attached in appendix C. To simplify the presentation, the diagonal weighting matrixes, $\mathbf{Q}$ and $\mathbf{R}$, are defined as $\mathbf{I}$ and $r\mathbf{I}$. The simulation results are shown in this section.

The initial conditions are set as follows:

Throttle: 2
Initial angular rates: [0 0 0]
Initial Euler angle: [0 0 0]
Initial Velocities: [25 0 0]
Initial position:[100 0 0]

The command inputs for the inner loop are defined as different signal wave forms to check the response of the systems under various command conditions. The desired angular rates are defined in the following:

Roll ($p$): square wave form
Pitch ($q$): square wave form
Yaw ($r$): sawtooth wave form

Without considering the dynamics of the actuator, the command trajectory of control inputs and system response are shown in the Figure 5.29 to 5.34 for different

horizon values, where solid line designates the desired outputs in the future and dotted line designates to the future response of system.

Ts = 0.005 nh = 50

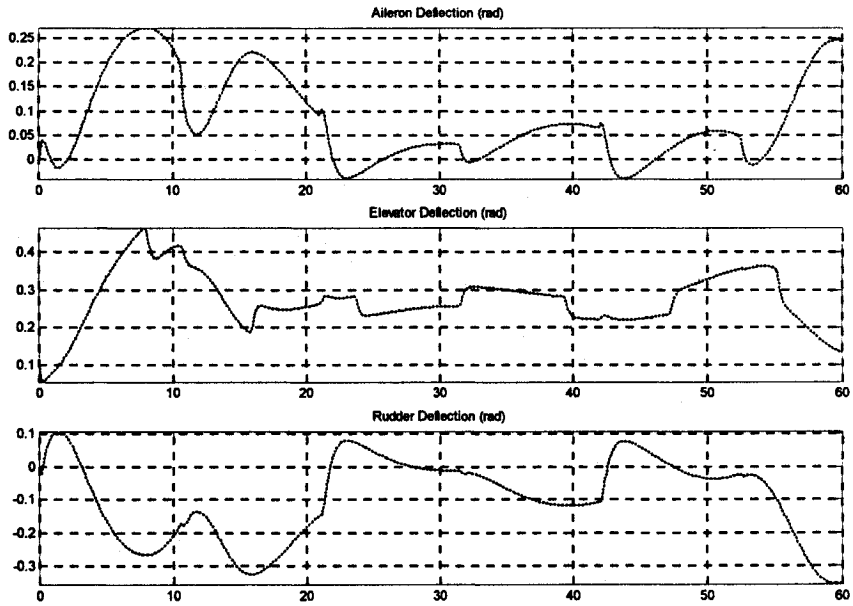



Figure 5.29 Input Command Trajectory (Ts = 0.005 nh = 50)

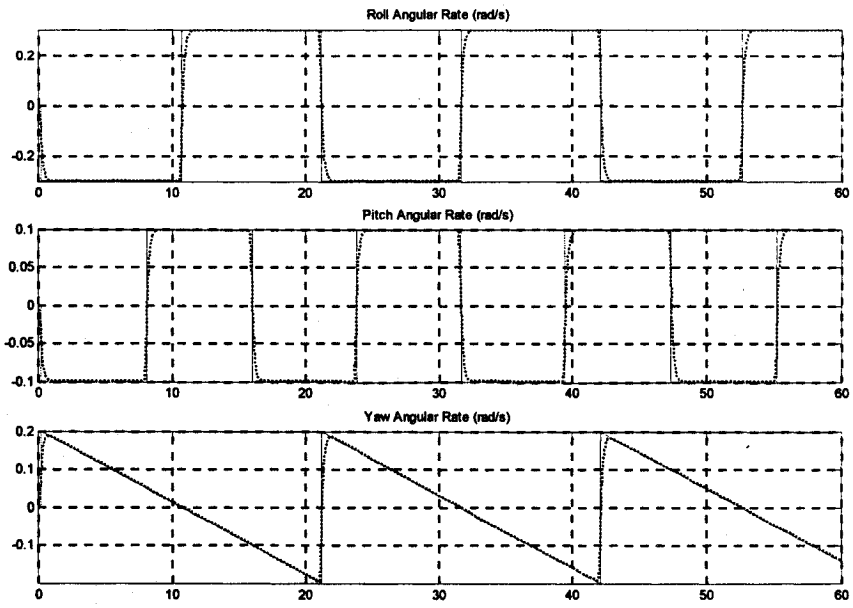



Figure 5.30 Model Predictive Control for Inner Loop (Ts = 0.005 nh = 50)

Ts = 0.005 nh = 100

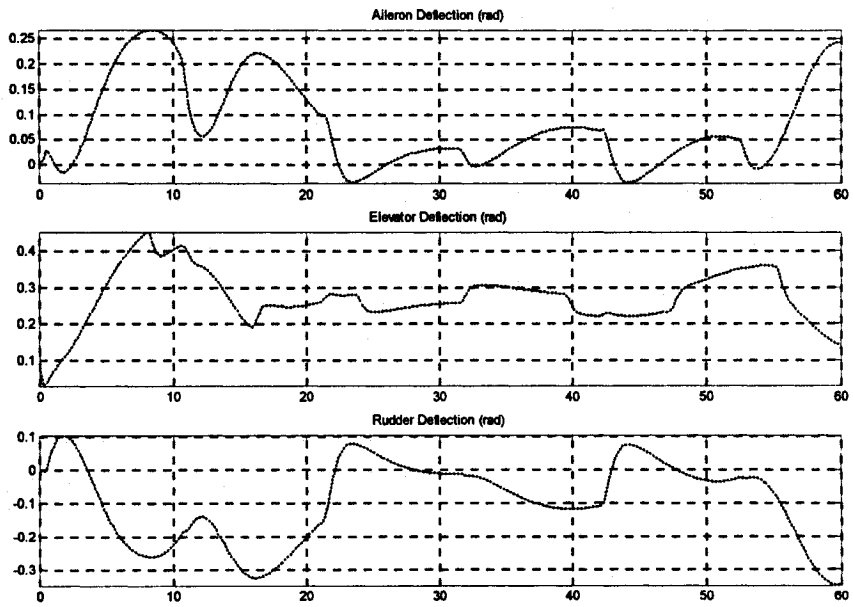



Figure 5.31 Input Command Trajectory (Ts = 0.005 nh = 100)

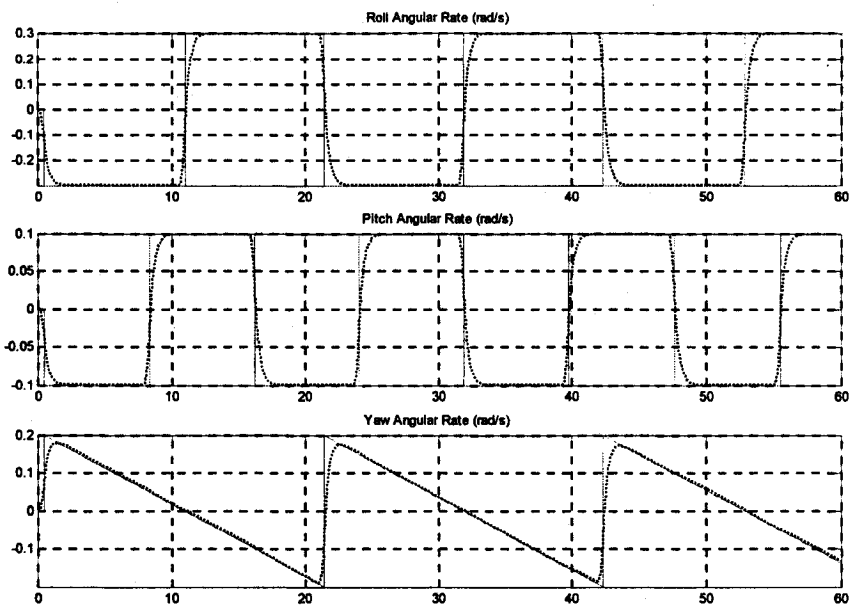



Figure 5.32 Model Predictive Control for Inner Loop (Ts = 0.005 nh = 100)

Ts = 0.005 nh = 200

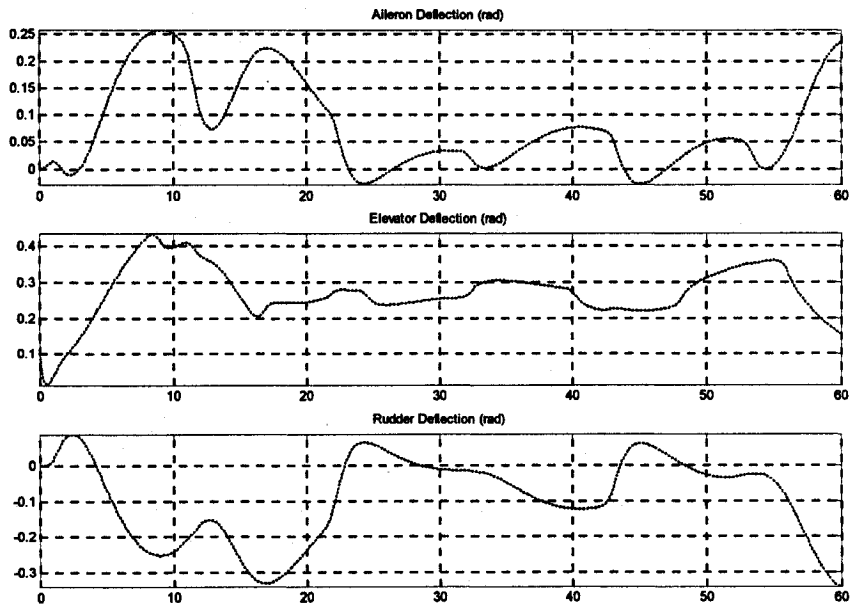



Figure 5.33 Input Command Trajectory (Ts = 0.005 nh = 200)

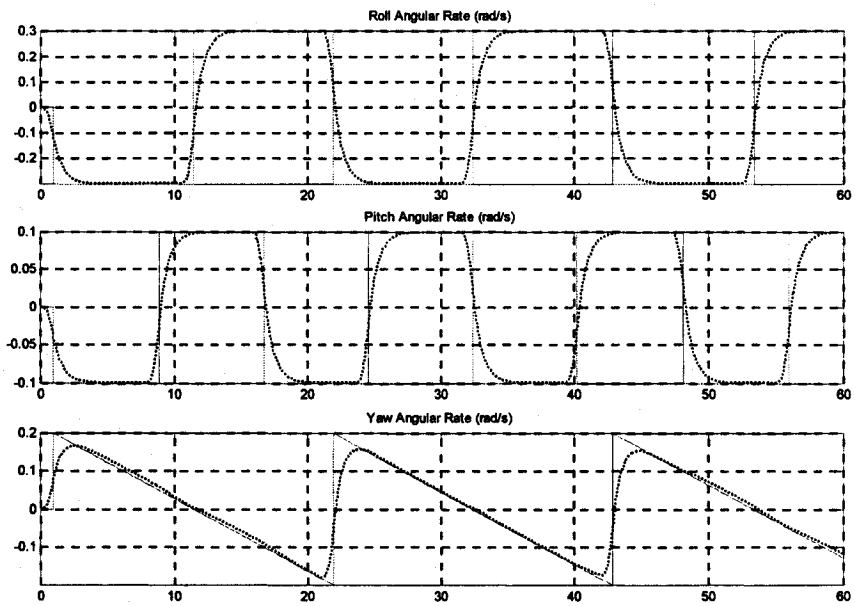



Figure 5.34 Model Predictive Control for Inner Loop (Ts = 0.005 nh = 200)

It is observed that the long horizon (nh) has a smoothing effect on the control surface deflection command and this can be used to preshape the command to suit the UAV's dynamics and avoid exciting various modes of vibration. Moreover, these results confirm that dynamic inversion is still effective in permitting model predictive to achieve the desired roll, pitch and yaw angular rates. Also these results confirm the advantage of model predictive control due to the predictive nature, shown by changes of angle rates that occur prior to actual changes in the desired angle rates. This advantage is important in achieving collision avoidance in UAVs formation flight starting the trajectory correction as soon as collision damage is sensed.

# Chapter 6

# 6 CONCLUSIONS AND RECOMMENDATIONS FOR FUTURE RESEARCH

This section is organized in two sections. The first section gives concluding remarks on the dynamic inversion and model predictive control, and the second section contains recommendations for future studies.

## 6.1 Conclusions

The accomplished work in this thesis is summarized as follows:

1. The complete nonlinear rigid body 6 DOF UAV is modeled by force equations, moment equations, kinematic equations and navigation equations.
2. The nonlinear controller is proposed based on the inner and outer loop dynamic inversion. A proportional controller is applied for inner loop to follow angular rates command and a proportional – integral controller is used for the outer loop to generate the command for the inner loop.
3. Dynamic Inversion permits the use of a linear model predictive control for the inner loop.
4. The simulation results are compared with classical controller and verify that the inner / outer loop dynamic inversion controller is superior to the classical controller in the whole flight envelope.

5. The combination of model predictive control with dynamic inversion proved to provide a solution for the autonomous control of an UAV to collision avoidance system.

## 6.2 Recommendations for Future Research

1. In the dynamic inversion controller approach, we assumed that the necessary states are observable at all time for the controller synthesis. The observer to estimate the specified states and estimation error effects are topics for the future study.
2. For the nonlinear UAV model, the aerodynamic coefficients are expressed by Taylor series approximation. In the multiple UAVs formation, these coefficients are significantly different from single UAV flight case. The robustness of controller has to be considered to deal with the uncertainty of each coefficient.
3. To complete the accurate path following, the navigation loop has to be added to generate the command reference as the inputs for the outer loop.
4. The high performance predictive controller with operational constraints has to be investigated.
5. To improve the performance of UAV, the high level path planning system has to be included.

# REFERENCES


Adams R. J. et al (1994), *"Design of Nonlinear Control Laws for High-Angle-of-Attack Flight"* Journal of Guidance, Control and Dynamics, Vol 17 No. 4 July-August 1994

Avanzini G. et al (2002), "*Modeling and Simulation of A Shrouded-fan UAV for Environmental monitoring*" 1st UAV Conference (AIAA 2002-3464)

Azam M. and Singh S. N. (1994), "*Invariability and Trajectory Control for Nonlinear Maneuvers of Aircraft*" Journal of Guidance, Control, and Dynamics Vol. 17, No. 1, January-February 1994

Baum M. L. and Passino K. M. (2002), "*A Search-Theoretic Approach to Cooperative Control for Uninhabited Air Vehicles*" AIAA Guidance, Navigation, and Control Conference and Exhibit (AIAA 2002-4589)

Berlin F. and Frank P.M. (1994), "*Design and realization of a MIMO predictive controller for a 3-tank system*", *Advances in Model-Based Predictive Control*, Oxford University Press, pp. 446-457, 1994

Bhattacharya R. et al (2002), *"Nonlinear Receding Horizon Control of an F-16 Aircraft"* Journal of Guidance, Control and Dynamics, Vol 25 No. 5 September-October 2002

Boiffier J. L. (1998), *The Dynamics of Flight The Equations*, John Wiley & Sons Ltd.

Botto M. A. and Costa J. S. (1998), *"Constrained predictive control of a multivariable feedback linearized system: application to a crane container"*, In 17th IASTED International Conference on Modelling, Identification and Control, pp. 328-330, Grindelwald, Switzerland, 1998

Bugajski D. J. and Enns D. F. (1992), *"Nonlinear Control Law with Application to High Angle-of-Attack Flight"* Journal of Guidance, Control and Dynamics Vol. 15, No. 3, May-June 1992

Clarke D. (1994), *Advances in model-based predictive control*, Oxford, New York, Oxford University Press

Etkin, B and Reid, L.D (1996), *Dynamics of Flight-Stability and Control*. John Wiley & Sons, 3rd Edition.

Frazzoli E. (2002), "*Maneuver-based Motion Planning and Coordination for Single and Multiple UAV's*" 1st UAV Conference (AIAA 2002-3472)

Gregory I. M. (1999), *"Modified Dynamic Inversion To Control Large Flexible Aircraft – What's Going On?"* AIAA Guidance, Navigation, and Control Conference and Exhibit, Portland, OR, Aug. 9-11, 1999, Collection of Technical Papers. Vol. 1 (AIAA 1999-3998)

Heise S. A. and Maciejowski J.M. (1996), *"Model Predictive Control of Supermaneuverable Aircraft"*, AIAA, Guidance, Navigation and Control Conference, San Diego, CA, July 29-31, 1996 (AIAA-96-3768)

In Joong HA and Chong S. (1992), "*Design of a CLOS Guidance Law Via Feedback Linearization*" IEEE TRANSACTIONS ON AEROSPACE AND ELECTRONIC SYSTEMS VOL 28. NO.1 JANUARY, 1992

Ito D. et al (2001), "*Robust Dynamic Inversion Controller Design and Analysis for the X-38*" AIAA Guidance, Navigation, and Control Conference and Exhibit, Montreal, Canada. Aug. 6-9, 2001. (AIAA2001-4380)

Jang I. J. S. and Tomlin C. J. (2001), "*Autopilot Design for the Stanford DragonFly UAV: Validation through Hardware-in-the-Loop Simulation*" AIAA Guidance, Navigation, and Control Conference and Exhibit, Montreal, Canada, Aug. 6-9, 2001 (AIAA 2001-4179)

Joshi A. and Pramodh P. G. (2002), "*Modeling and Simulation of Launch Vehicle Digital Auto Pilot*" AIAA Modeling and Simulation Technologies Conference and Exhibit (AIAA 2002-4969)

Jung Y. C. and Hess R. A. (1991), "Precise Flight-Path Control Using a Predictive Algorithm" Journal of Guidance, Control and Dynamics, Vol 14 No.5 pp. 936-942

Kato O. and Sugiura I. (1992), "*Nonlinear Inversion Flight Control for a Supermaneuverable Aircraft*" Journal of Guidance, Control, and Dynamics Vol. 15, No. 4, July-August 1992

Lane S. H. and Stengel R. F. (1988), *"Flight Control Design Using Non-Linear Inverse Dynamics"* Automatica, Vol. 24, No. 4, pp. 471-483

Lee T. and Kim Y. (2001), *"Nonlinear Adaptive Flight Control Using Backstepping and Neural Networks Controller"* Journal of Guidance, Control and Dynamics, Vol 24 No. 4 July-August 2001

Liang F. (2002), "*Rapid Development of UAV Autopilot Using Matlab/Simulink*" AIAA

Looye G. (1999), "*Integrated Flight Mechanics and Aeroelastic Aircraft Modeling Using Object-Oriented Modeling Techniques*" AIAA Modeling and Simulation Technologies Conference and Exhibit, Portland, OR, Aug. 9-11, 1999, Collection of Technical Papers (AIAA 1999-4192)

Modeling and Simulation Technologies Conference and Exhibit (AIAA 2002-4976)

Maciejowski J. M. (2002), *Predictive control with constraints*, Essex, England; Prentice Hall

Mayne D. Q. et al (2000), *"Constrained model predictive control: Stability and optimality"* Automatica 36 (2000) 789-814

McLean D. (1990), *Automatic Flight Control Systems*, Englewood Cliffs, N.J.: Prentice Hall.

Mehra R.K. et al (2001), *"Model Predictive Control Design for XV-15 Tilt Rotor Flight Control"* AIAA Guidance, Navigation, and Control Conference and Exhibit, Montreal, Canada, Aug. 6-9, 2001

Menon P.K.A. et al (1987), "*Nonlinear Flight Test Trajectory Controller for Aircraft*" Journal of Guidance Vol. 10, No.1, JAN-FEB 1987.

Mettler M. B., et al (2002), "*Trajectory reconfiguration for an Unmanned Aircraft*" AIAA Guidance, Navigation, and Control Conference and Exhibit (AIAA 2002-4674)

Nelson R. C. (1998), *Flight Stability and Automatic Control* second edition, McGraw-Hill

Okan, A. et al. (2002), "*Flight Control of a Tilt-Duct VTOL UAV*" 1st UAV Conference (AIAA 2002-3466)

Reiner J. et al (1995), "*Robust Dynamic Inversion For Control of Highly Maneuverable Aircraft*" Journal of Guidance, Control, and Dynamics Vol. 18, No. 1, January-February 1995.

Schumacher C. (1999), "*Adaptive Flight Control Using Dynamic Inversion and Neural Networks*", AIAA Guidance, Navigation, and Control Conference and Exhibit, Portland, OR, Aug. 9-11, 1999, Collection of Technical Papers. Vol. 2 (AIAA 1999-4086)

Schumacher C. J. and Kumar R. (2000), "*Adaptive Control of UAVs in Close-Coupled Formation Flight*" in Proceedings of the American Control Conference, Chicago, Illinois, June 2000

Schumacher C. J. and Singh S. N. (2000), "*Nonlinear Control of Multiple UAVs in Closed-Coupled Formation Flight*" AIAA Guidance, Navigation, and Control Conference and Exhibit, Denver, CO, Aug. 14-17, 2000 (AIAA 2000-4373)

Singh S. N. and Steinberg M. (1996), *"Adaptive Control of Feedback Linearizable Nonlinear Systems With Application to Flight Control"* AIAA Guidance Navigation and control conference July 29-31, 1996 San Diego, CA (AIAA-96-3771)

Slotine J. E. and Li W. (1991), *Applied Nonlinear Control*. Prentice-Hall Inc.

Snell S. A. et al (1992), "*Nonlinear Inversion Flight Control for a Supermaneuverable Aircraft*" Journal of Guidance, Control and Dynamics, Vol 15 No. 4 July-August 1992

Steinbery M. L. (2001), "*Comparison of Intelligent, Adaptive and Nonlinear Flight Control Laws*" Journal of Guidance, Control, and Dynamics Vol. 24, No. 4 July-August 2001

Stevens B. L. and Lewis F. L. (1992), *Aircraft Control and Simulation*, A Wiley-Interscience Publication

Soeterboek, R. (1992), *Predictive Control: A Unified Approach*, International Series in Systems and Control Engineering, Prentice-Hall, Upper Saddle River, NJ, 1992

[1] http://www.u-dynamics.com/aerosim/

# Appendix A

## Hypotheses

The airplane is symmetrical.

The gravity force is a constant.

The mass of plane doesn't change during the operation, without considering fuel consume.

The origin of the body frame is located at the mass center.

The airplane is a rigid body.

The air is static.

All states are available at any time for the nonlinear feedback loop computation.

The earth fixed frame is a non rotated and non accelerated frame

The dynamics of actuator does not consider.

As the aileron, rudder, and elevator produce small forces, we assume $C_y^{\delta_a} = C_y^{\delta_r} = 0$ and engine thrust force $F_T$ is constant.

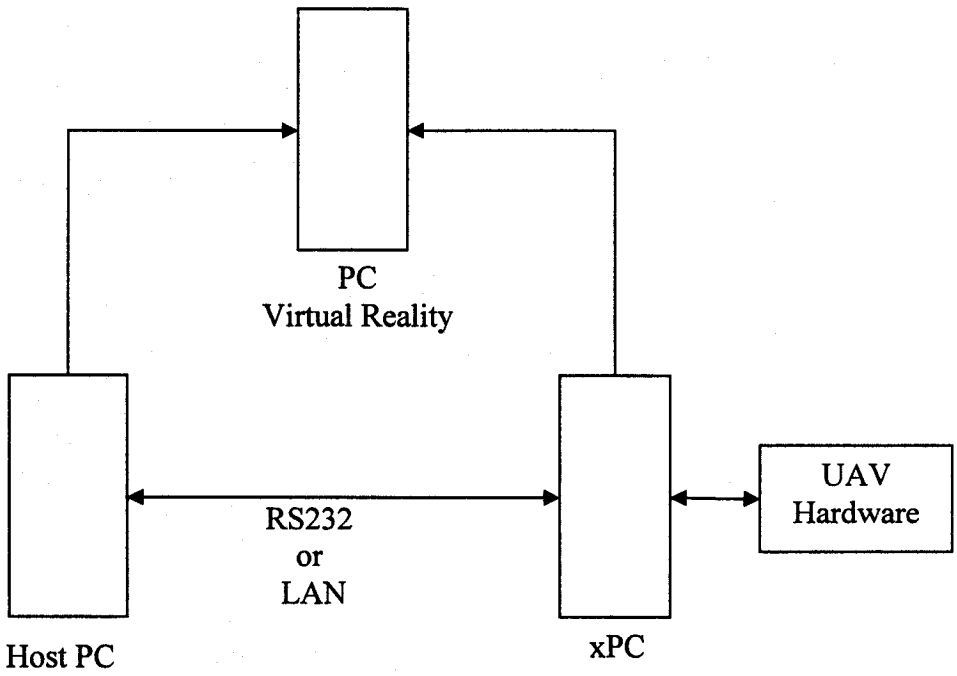
# Appendix B

## Proposal for the Hardware In the Loop System Configuration

Although the characteristics of each element are modeled as accurate as possible in the UAV controller design, the control algorithm can't be applied to the real system directly due to unexpected imperfect modeling and interface incompatible. Therefore, it is essential to validate the control algorithm with actual hardware to ensure successful functioning of each part. In the following part, the Hardware In the Loop system is proposed.

The configuration is built up with help of the xPC Target, which is a product of mathworks.

In the host PC, the Simulink program is converted to C code and compiled the code into a target application with Real-Time Workshop®, then the target application is downloaded into xPC where the executable code runs in real time and the parameters of program can be tuned in the host PC during execution. The data transfer between two PCs is carried out one of two connections, serial port or network. In the serial connection, the information between host PC and xPC is communicated by a serial cable using RS232 ports. In the network connection, the information is transferred by the TCP/IP protocol.




The data acquisition device is installed in the xPC. The actuator's commands and the data from sensors are transferred between the host PC and xPC in real time. In order to monitor the motion of UAV in 3D environment and display the data of effectors in real time without reducing the performance of host PC, the third computer is chosen to achieve these tasks. The reference motion commands and actuator's commands are transferred to the virtual reality PC from host PC and the data of sensors are transferred from xPC.

# Appendix C

# Programmes

## Part 1:

Maple Program for the Transformation Matrices derivation

```
> restart: clear:with(linalg, adj, coldim, det, diag,
inverse, minor, multiply, rowdim, transpose) :
```

DCM1 transfer from Body to Wind

```
> R[alpha]:=<<cos(alpha) | 0 | sin(alpha)> , <0 | 1 | 0> ,
<-sin(alpha) | 0 | cos(alpha)>>;
```

$$R_{\alpha} := \begin{bmatrix} \cos(\alpha) & 0 & \sin(\alpha) \\ 0 & 1 & 0 \\ -\sin(\alpha) & 0 & \cos(\alpha) \end{bmatrix}$$

```
> R[beta]:=<<cos(beta) | sin(beta) | 0> , <-sin(beta) |
cos(beta) | 0> , <0 | 0 | 1>>;
```

$$R_{\beta} := \begin{bmatrix} \cos(\beta) & \sin(\beta) & 0 \\ -\sin(\beta) & \cos(\beta) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

```
> DCM1:=multiply(R[beta],R[alpha]);
```

$$DCM1 := \begin{bmatrix} \cos(\beta)\cos(\alpha) & \sin(\beta) & \cos(\beta)\sin(\alpha) \\ -\sin(\beta)\cos(\alpha) & \cos(\beta) & -\sin(\beta)\sin(\alpha) \\ -\sin(\alpha) & 0 & \cos(\alpha) \end{bmatrix}$$

```
> DCM1_inv:=simplify(inverse(DCM1));
```

$$DCM1\_inv := \begin{bmatrix} \cos(\beta)\cos(\alpha) & -\sin(\beta)\cos(\alpha) & -\sin(\alpha) \\ \sin(\beta) & \cos(\beta) & 0 \\ \cos(\beta)\sin(\alpha) & -\sin(\beta)\sin(\alpha) & \cos(\alpha) \end{bmatrix}$$

DCM2 transfter from Inertail to Body

```
> Rx:=<<1 | 0 | 0> , <0 | cos(phi) | sin(phi)> , <0 | -
sin(phi) | cos(phi)>>;
```

$$Rx := \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\phi) & \sin(\phi) \\ 0 & -\sin(\phi) & \cos(\phi) \end{bmatrix}$$

```
> Ry:=<<cos(theta) | 0 | -sin(theta)> , <0 | 1 | 0> ,
<sin(theta) | 0 | cos(theta)>>;
```

$$Ry := \begin{bmatrix} \cos(\theta) & 0 & -\sin(\theta) \\ 0 & 1 & 0 \\ \sin(\theta) & 0 & \cos(\theta) \end{bmatrix}$$

```
> Rz:=<<cos(psi) | sin(psi) | 0> , <-sin(psi) | cos(psi) |
0> , <0 | 0 | 1>>;
```

$$Rz := \begin{bmatrix} \cos(\psi) & \sin(\psi) & 0 \\ -\sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

```
> DCM2:=multiply(Rx,multiply(Ry,Rz));
```

$DCM2 :=$

$[\cos(\theta)\cos(\psi), \cos(\theta)\sin(\psi), -\sin(\theta)]$

$[-\cos(\phi)\sin(\psi) + \sin(\phi)\sin(\theta)\cos(\psi), \cos(\phi)\cos(\psi) + \sin(\phi)\sin(\theta)\sin(\psi), \sin(\phi)\cos(\theta)]$

$[\sin(\phi)\sin(\psi) + \cos(\phi)\sin(\theta)\cos(\psi), -\sin(\phi)\cos(\psi) + \cos(\phi)\sin(\theta)\sin(\psi), \cos(\phi)\cos(\theta)]$

```
>
Rx_inv:=simplify(inverse(Rx)):Ry_inv:=simplify(inverse(Ry))
:Rz_inv:=simplify(inverse(Rz)):

> DCM2_inv:=multiply(Rz_inv,multiply(Ry_inv,Rx_inv));
```

$DCM2\_inv :=$

$[\cos(\theta)\cos(\psi), -\cos(\phi)\sin(\psi) + \sin(\phi)\sin(\theta)\cos(\psi), \sin(\phi)\sin(\psi) + \cos(\phi)\sin(\theta)\cos(\psi)]$

$[\cos(\theta)\sin(\psi), \cos(\phi)\cos(\psi) + \sin(\phi)\sin(\theta)\sin(\psi), -\sin(\phi)\cos(\psi) + \cos(\phi)\sin(\theta)\sin(\psi)]$

$[-\sin(\theta), \sin(\phi)\cos(\theta), \cos(\phi)\cos(\theta)]$

```
> multiply(DCM1,DCM2);
```

$[\cos(\beta)\cos(\alpha)\cos(\theta)\cos(\psi) + \sin(\beta)(-\cos(\phi)\sin(\psi) + \sin(\phi)\sin(\theta)\cos(\psi))$
$+ \cos(\beta)\sin(\alpha)(\sin(\phi)\sin(\psi) + \cos(\phi)\sin(\theta)\cos(\psi)),$
$\cos(\beta)\cos(\alpha)\cos(\theta)\sin(\psi) + \sin(\beta)(\cos(\phi)\cos(\psi) + \sin(\phi)\sin(\theta)\sin(\psi))$
$+ \cos(\beta)\sin(\alpha)(-\sin(\phi)\cos(\psi) + \cos(\phi)\sin(\theta)\sin(\psi)),$
$-\cos(\beta)\cos(\alpha)\sin(\theta) + \sin(\beta)\sin(\phi)\cos(\theta) + \cos(\beta)\sin(\alpha)\cos(\phi)\cos(\theta)]$
$[-\sin(\beta)\cos(\alpha)\cos(\theta)\cos(\psi)$
$+ \cos(\beta)(-\cos(\phi)\sin(\psi) + \sin(\phi)\sin(\theta)\cos(\psi))$
$- \sin(\beta)\sin(\alpha)(\sin(\phi)\sin(\psi) + \cos(\phi)\sin(\theta)\cos(\psi)),$
$-\sin(\beta)\cos(\alpha)\cos(\theta)\sin(\psi) + \cos(\beta)(\cos(\phi)\cos(\psi) + \sin(\phi)\sin(\theta)\sin(\psi))$
$- \sin(\beta)\sin(\alpha)(-\sin(\phi)\cos(\psi) + \cos(\phi)\sin(\theta)\sin(\psi)),$
$\sin(\beta)\cos(\alpha)\sin(\theta) + \cos(\beta)\sin(\phi)\cos(\theta) - \sin(\beta)\sin(\alpha)\cos(\phi)\cos(\theta)]$
$[-\sin(\alpha)\cos(\theta)\cos(\psi) + \cos(\alpha)(\sin(\phi)\sin(\psi) + \cos(\phi)\sin(\theta)\cos(\psi)),$
$-\sin(\alpha)\cos(\theta)\sin(\psi) + \cos(\alpha)(-\sin(\phi)\cos(\psi) + \cos(\phi)\sin(\theta)\sin(\psi)),$
$\sin(\alpha)\sin(\theta) + \cos(\alpha)\cos(\phi)\cos(\theta)]$

## Part 2:

UAV Parameters Initial File

```
% Clear workspace
clear all;

% Name of the MAT-file that will be generated
cfgmatfile = 'aerocfg';

%%% AERODYNAMICS %%%
% Aerodynamic force application point (usually the aerodynamic center)[x y z]
rAC = [0.1425 0 0]; % m

%%% Aerodynamic parameter bounds %%%
% Airspeed bounds
VaBnd = [5 100];
% VaBnd = [15 50]; % m/s
% Sideslip angle bounds
BetaBnd = [-2 2]; % rad
% BetaBnd = [-0.5 0.5]; % rad
% Angle of attack bounds

AlphaBnd = [-2 2]; % rad
% AlphaBnd = [-0.1 0.3]; % rad

%%% Aerodynamic reference parameters %%%
% Mean aerodynamic chord
MAC = 0.189941; % m
% Wind span
b = 2.8956; % m
% Wing area
S = 0.55; % m^2

% ALL aerodynamics derivatives are per radian:
%%% Lift coefficient %%%
% Zero-alpha lift
CL0 = 0.23;
% alpha derivative
CLa = 5.6106;
% Lift control (flap) derivative
CLdf = 0.74;
% Pitch control (elevator) derivative
CLde = 0.13;
% alpha-dot derivative
```

```
CLalphadot = 1.9724;
% CLalphadot = 0;
% Pitch rate derivative
CLq = 7.9543;
% Mach number derivative
CLM = 0;

%%% Drag coefficient %%%
% Lift at minimum drag
CLmind = 0.23;
% Minimum drag
CDmin = 0.0434;
% Lift control (flap) derivative
CDdf = 0.1467;
% Pitch control (elevator) derivative
CDde = 0.0135;
% Roll control (aileron) derivative
CDda = 0.0302;
% Yaw control (rudder) derivative
CDdr = 0.0303;
% Mach number derivative
CDM = 0;
% Oswald's coefficient
osw = 0.75;

%%% Side force coefficient %%%
% Sideslip derivative
CYbeta = -0.83;
% Roll control derivative
CYda = -0.075;
% Yaw control derivative
CYdr = 0.1914;
% Roll rate derivative
CYp = 0;
% Yaw rate derivative
CYr = 0;

%%% Pitch moment coefficient %%%
% Zero-alpha pitch
Cm0 = 0.135;
% alpha derivative
Cma = -2.7397;
% Lift control derivative
Cmdf = 0.0467;
% Pitch control derivative
```

```
Cmde = -0.9918;
% alpha_dot derivative
% Cmalphadot = 0
Cmalphadot = -10.3796;
% Pitch rate derivative
Cmq = -38.2067;
% Mach number derivative
CmM = 0;

%%% Roll moment coefficient %%%
% Sideslip derivative
Clbeta = -0.13;
% Roll control derivative
Clda = -0.1695;
% Yaw control derivative
Cldr = 0.0024;
% Roll rate derivative
Clp = -0.5051;
% Yaw rate derivative
Clr = 0.2519;

%%% Yaw moment coefficient %%%
% Sideslip derivative
Cnbeta = 0.0726;
% Roll control derivative
Cnda = 0.0108;
% Yaw control derivative
Cndr = -0.0693;
% Roll rate derivative
Cnp = -0.069;
% Yaw rate derivative
Cnr = -0.0946;

%%% ENGINE %%%
Cth=13;

%%% INERTIA %%%
% Aircraft mass
m_aircraft=11; %kg
% Inertia
Ix=0.80195; %kg*m^2
Iy=1.1285; %kg*m^2
Iz=1.7555; %kg*m^2
Ixz=0; %kg*m^2
```

```
%Gravity
g=9.8;

% Save workspace variables to MAT file
save(cfgmatfile);

% Output a message to the screen
fprintf(strcat('\n Aircraft configuration saved as:\t', strcat(cfgmatfile),'.mat'));
fprintf('\n');
```

## Part 3:

S-function for Inner and Outer Loop Dynamic Inversion

```
/*Dynamic Inverse Program
 Copyright 2003 Zhaoquan Cheng and Dan Necsulescu
 Version 1.0 Date: September 10, 2003

 INPUT
        u[0]: VT        u[1]: beta              u[2]: alpha
        u[3]: p         u[4]: q         u[5]: r
        u[6]: phi       u[7]: theta             u[8]: psi
        u[9]: dphi      u[10]:dtheta    u[11]:dpsi
        u[12]: alphadot u[13]: delat_th

        u2[0]: VT_dot   u2[1]: beta_dot u2[2]: alpha_dot

        Parameters:
        Ix              Iy              Iz
        rho             b               Warea       mac
        CYbeta          CYp             CYr
        Clbeta          Clp             Clr
        Cnbeta          Cnp             Cnr
        Cm0             Cma             Cmalphadot  Cmq
        Cth             m               g
        CYda            CYdr
        Clda            Cldr
        Cnda            Cndr
        Cmde
        CL0             CLa             CLq

        OUTPUT:
        f:
        f[0]: f_p
        f[1]: f_q
        f[2]: f_r
        A [3x3] matrix
*/
double coeff,f_alpha,f_beta,f_theta,f_phi;
int i;


coeff=0.5*(*rho)*u[0]*u[0]*(*Warea);
```

```
f_alpha = (-(*Cth)*u[13]*sin(u[2])-
0.5*(*rho)*u[0]*u[0]*((*CL0)+(*CLa)*u[2]+0.5*(*mac)*(*CLq)/u[0]))/((*m)*u[0]*cos
(u[1]))
+(-u[3]*cos(u[2])*sin(u[1])+u[4]*cos(u[1])-u[5]*sin(u[2])*sin(u[1]))/cos(u[1])
+(*g)*(sin(u[2])*sin(u[7])+cos(u[2])*cos(u[6])*cos(u[7]))/(u[0]*cos(u[1]));

f_beta = (-
(*Cth)*u[13]*cos(u[2])*sin(u[1])+0.5*(*rho)*u[0]*u[0]*(*Warea)*((*CYbeta)*u[1]+0.5
*(*b)*((*CYp)*u[3]+(*CYr)*u[5])/u[0]))/((*m)*u[0])
- (-u[3]*sin(u[2])+u[5]*cos(u[2]))
+(*g)*(cos(u[2])*sin(u[1])*sin(u[7])
+cos(u[1])*sin(u[6])*cos(u[7])-sin(u[1])*sin(u[2])*cos(u[6])*cos(u[7]))/u[0];

f_theta = u[4]*cos(u[6])-u[5]*sin(u[7]);

f_phi = u[3]+tan(u[7])*(u[4]*sin(u[6])+u[5]*cos(u[6]));


f[0] = ((*Iy)-(*Iz))/(*Ix)*u[5]*u[4]
+ 0.5*(*rho)*u[0]*u[0]*(*Warea)*(*b)*((*Clbeta)*u[1]
+0.5*(*b)*((*Clp)*u[3]+(*Clr)*u[5])/u[0])/(*Ix);

f[1] = ((*Iz)-(*Ix))/(*Iy)*u[3]*u[5]
+ 0.5*(*rho)*u[0]*u[0]*(*Warea)*(*mac)*((*Cm0)+(*Cma)*u[2]
+0.5*(*mac)*((*Cmq)*u[4]+(*Cmalphadot)*u[12])/u[0])/(*Iy);

f[2] = ((*Ix)-(*Iy))/(*Iz)*u[3]*u[4]
+ 0.5*(*rho)*u[0]*u[0]*(*Warea)*(*b)*((*Cnbeta)*u[1]
+0.5*(*b)*((*Cnp)*u[3]+(*Cnr)*u[5])/u[0])/(*Iz);
```

```
h[0]=0;

h[1]=0;

h[2] = -(*Cth)*u[13]*cos(u[2])*sin(u[1])/((*m)*u[0])
+ 0.5*(*rho)*u[0]*(*Warea)*(*CYbeta)*u[1]/(*m)
+ (*g)*(cos(u[2])*sin(u[1])*sin(u[7])+cos(u[1])*sin(u[6])*cos(u[7])-
sin(u[2])*sin(u[1])*cos(u[6])*cos(u[7]))/u[0];



A[0] = (*b)*(*Clda)/(*Ix) ;
A[1] = 0;
A[2] = (*b)*(*Cnda)/(*Iz);
A[3] = (*b)*(*Cldr)/(*Ix);
A[4] = 0;
A[5] = (*b)*(*Cndr)/(*Iz);
A[6] = 0;
A[7] = (*mac)*(*Cmde)/(*Iy);
A[8] = 0;

for(i=0; i < 9; i++)
A[i] *=coeff;

G[0] = 1;
G[1] = 0;
G[2] = 0.25*(*rho)*(*Warea)*(*CYp)*(*b)/(*m)+sin(u[2]);
G[3] = sin(u[6])*tan(u[7]);
G[4] = cos(u[6]);
```

```
G[5] = 0;
G[6] = cos(u[6])*tan(u[7]);
G[7] = -sin(u[6]);
G[8] = 0.25*(*rho)*(*Warea)*(*CYr)/(*m)-cos(u[2]);

LfG[0] = 0;
LfG[1] = 0;
LfG[2] = f_alpha*cos(u[2]);
LfG[3] = f_phi*cos(u[6])*tan(u[7])+f_theta*sin(u[6])/(cos(u[7])*cos(u[7]));
LfG[4] = -f_phi*sin(u[6]);
LfG[5] = 0;
LfG[6] = -f_phi*sin(u[6])*tan(u[7])+f_theta*cos(u[6])/(cos(u[7])*cos(u[7]));
LfG[7] = -f_phi*cos(u[6]);
LfG[8] = f_alpha*sin(u[2]);

LfH[0] = 0;
LfH[1] = 0;
LfH[2] =(-(*Cth)*u[13]*cos(u[2])*cos(u[1])/((*m)*u[0])
+ 0.5*(*rho)*u[0]*(*Warea)*(*CYbeta)/(*m)
+ (*g)*(cos(u[2])*cos(u[1])*sin(u[7])-sin(u[1])*sin(u[6])*cos(u[7])
-cos(u[2])*sin(u[1])*cos(u[6])*cos(u[7]))/u[0])*f_beta
+ (cos(u[1])*cos(u[6])*cos(u[7])+sin(u[1])*sin(u[2])*sin(u[6])*sin(u[7])*(*g))/u[0]*u[9]
+ (sin(u[1])*cos(u[2])*cos(u[7])-cos(u[1])*sin(u[6])*sin(u[7])
+sin(u[1])*sin(u[2])*cos(u[6])*sin(u[7]))*(*g)/u[0]*u[10];
LfZ[0] = f_phi;
LfZ[1] = f_theta;
LfZ[2] = f_beta;
```

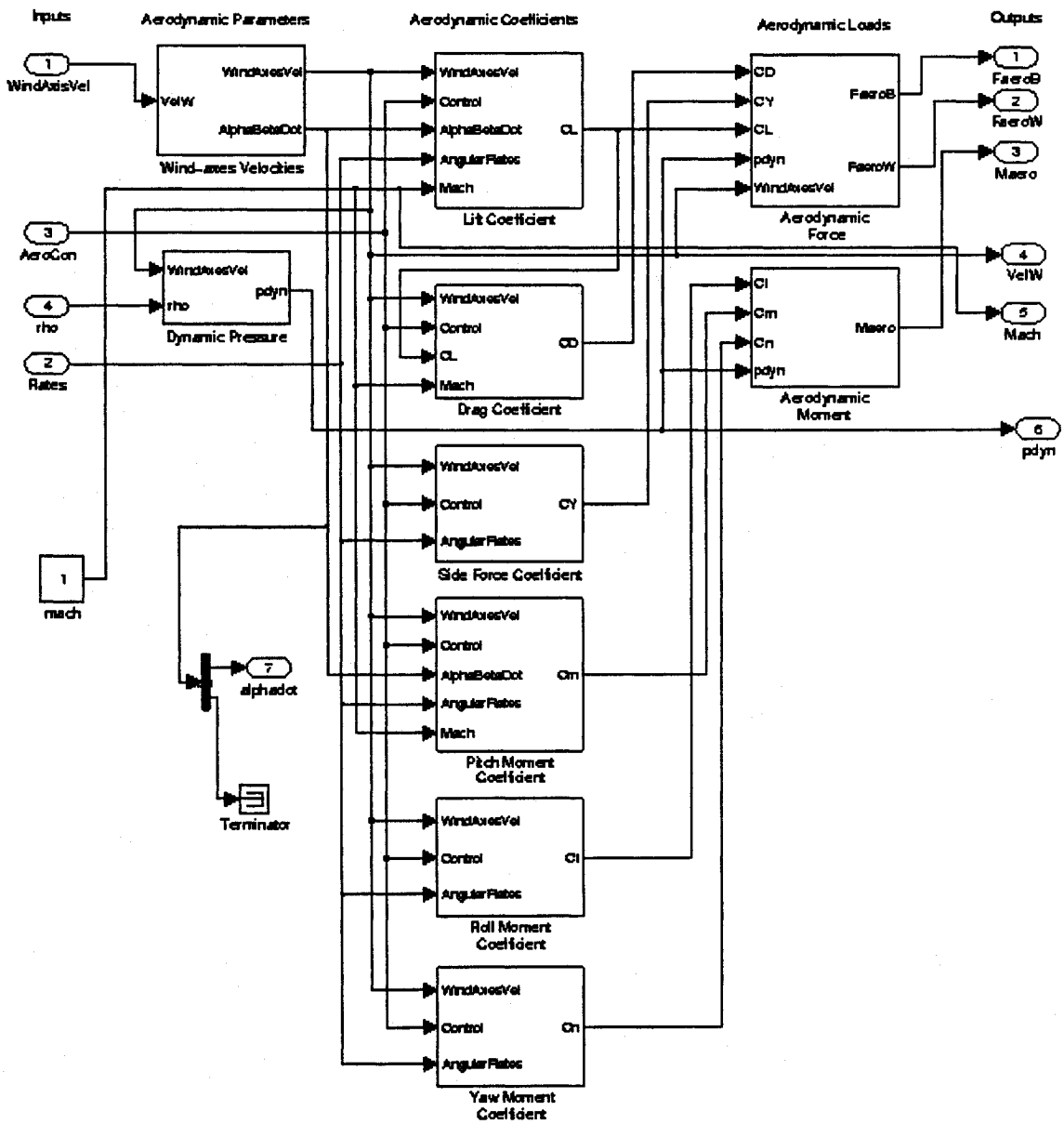
## Part 4:

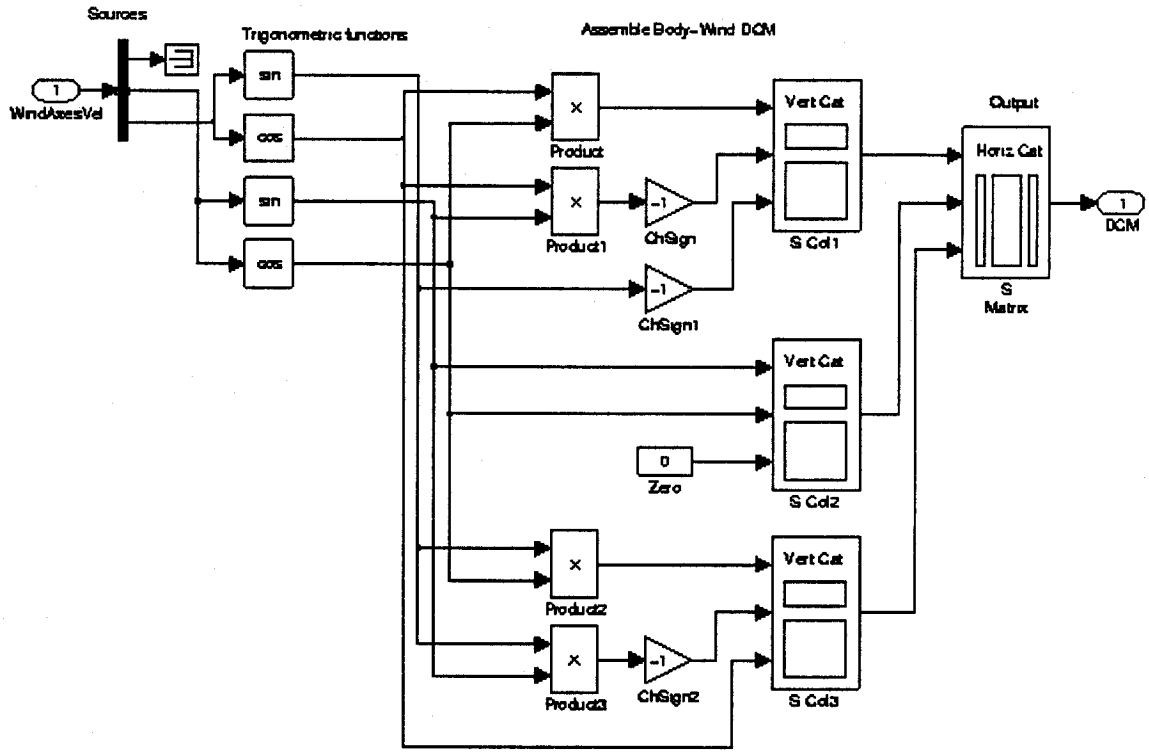
Model Predictive Control Initial file

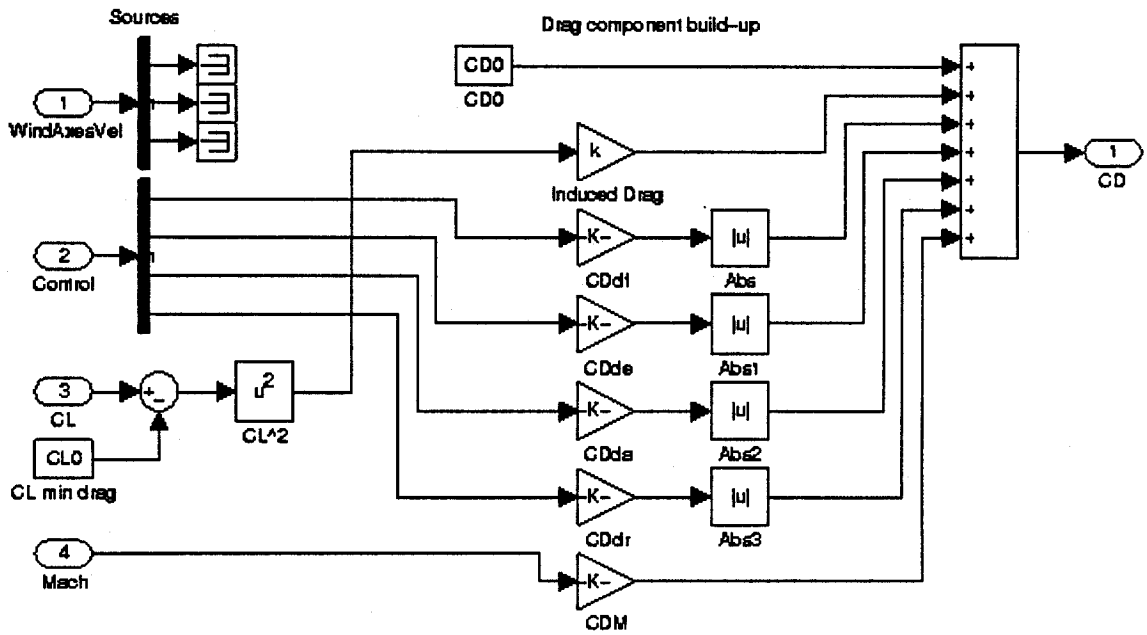
```
clear all; clc; close all;

Ts=0.005;
nh=50;
r=1e-9;
wn = 100;
Ta = 4.5*0.707/wn

% System Defination

A=0;
B=1;
C=1;
D=0;


[Ad,Bd,Cd,Dd]=c2dm(A,B,C,D,Ts,'zoh');

S(1,:)=Cd*Bd;

 for i=2:nh
    S(i,:)=Cd*Ad^(i-1)*Bd+S((i-1),:);
 end

for i=1:nh
   T(i,:)=Cd*Ad^i;
end
dum=S'*S;
n=max(size(dum));
G=inv(r*eye(n)+dum)*S';
GT=G*T;

% define delaying system
AD=[zeros(1,nh); eye(nh-1) zeros(nh-1,1)];
BD=[1;zeros(nh-1,1)];
CD=eye(nh);
DD=zeros(nh,1);
```

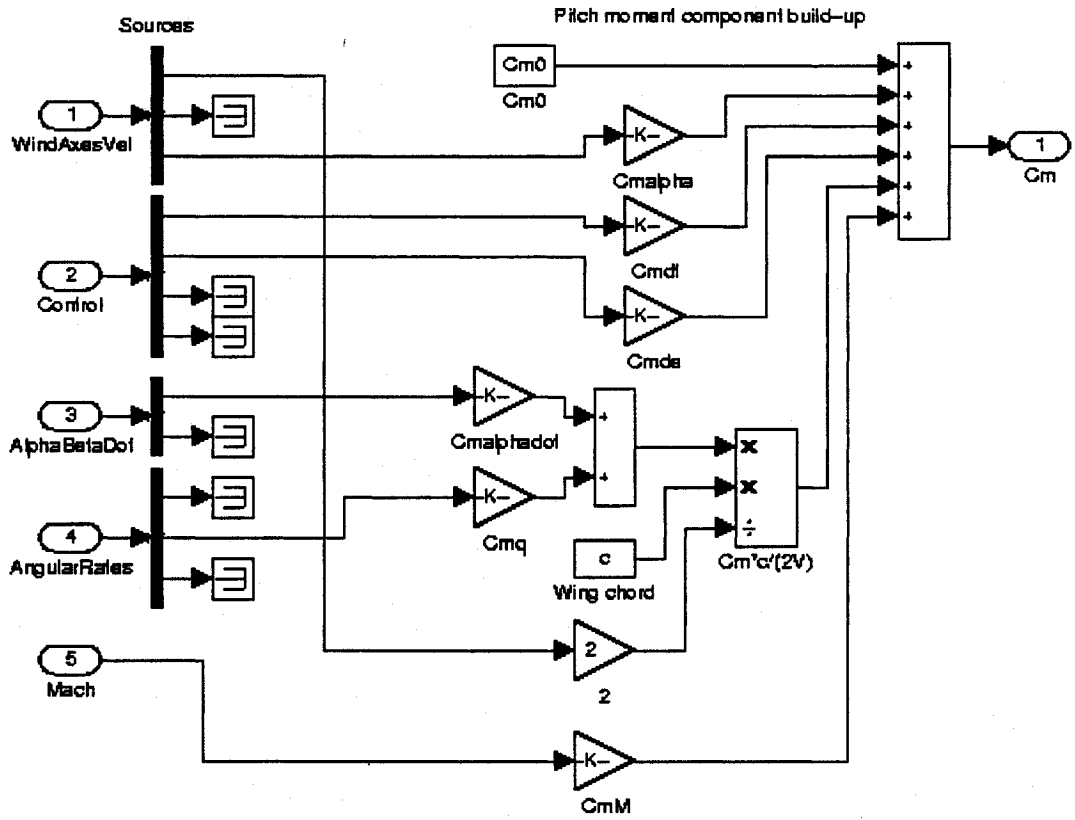
## Part 5:

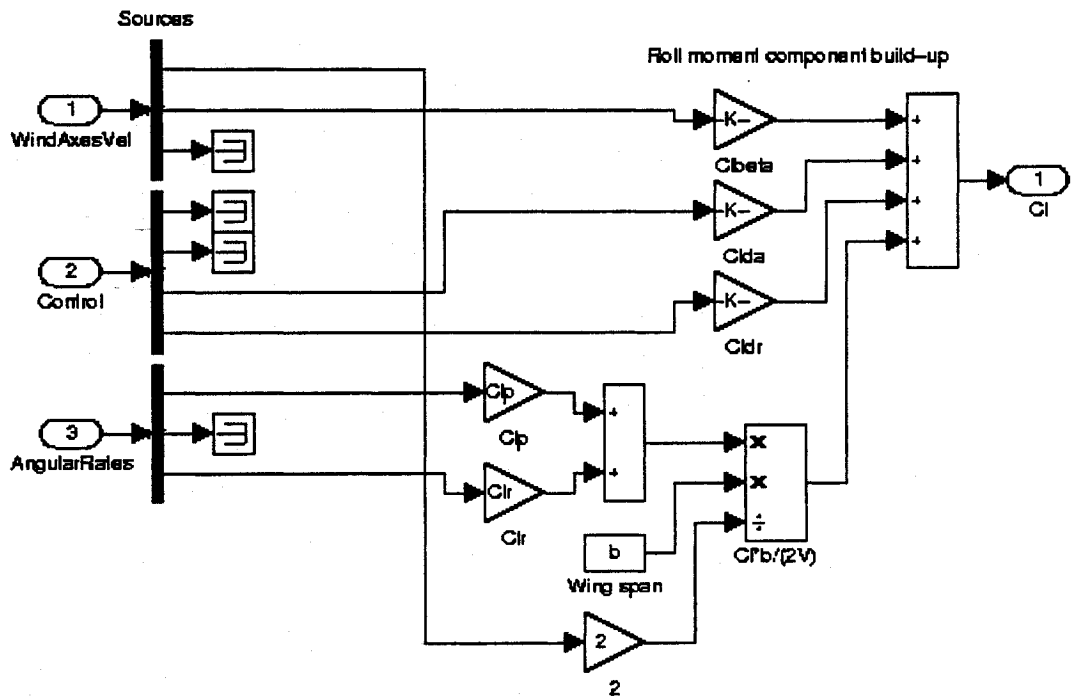
Simulink Program for UAV Model based on Aerosim

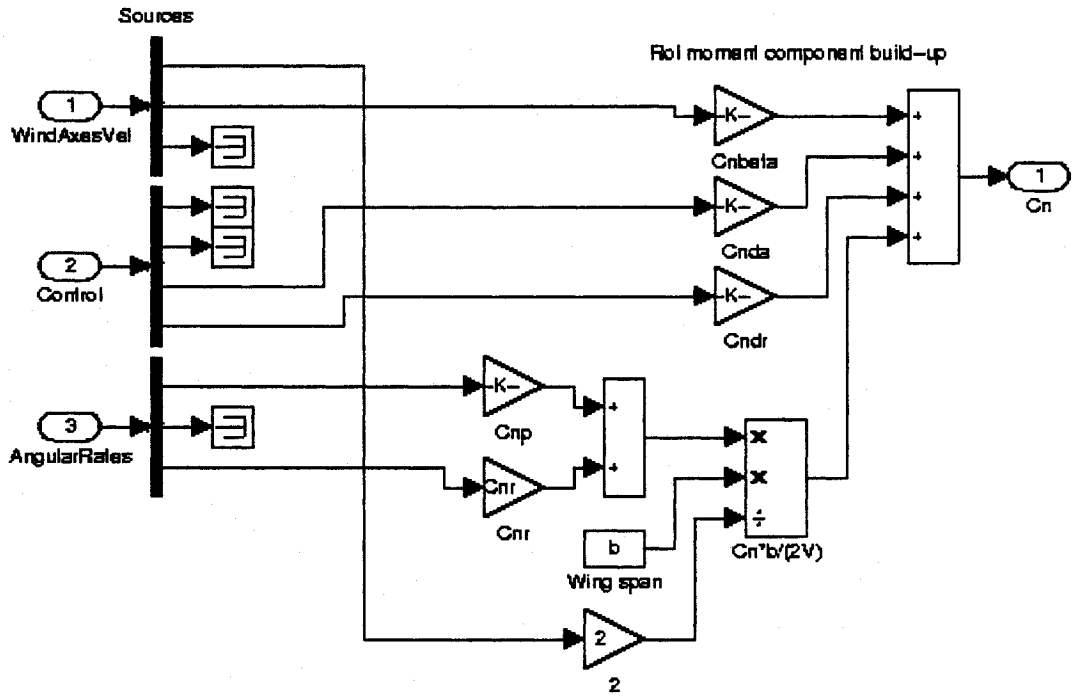


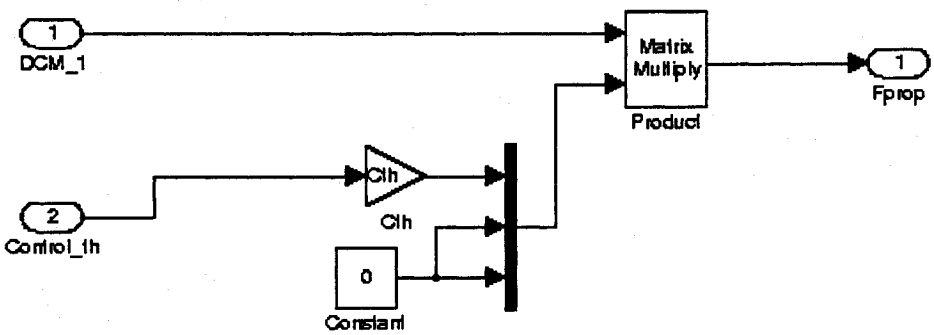

Sources
Trigonometric functions
Assemble Body-Wind DCM
1
WindAxesVel
sin
cos
sin
cos
Product
Product1
ChSign
ChSign1
Vert Cat
S Col1
Output
Horiz Cat
1
DCM
S
Matrix
Vert Cat
0
Zero
S Col2
Product2
Product3
ChSign2
Vert Cat
S Col3


Sources
Drag component build-up
CD0
CD0
1
WindAxesVel
k
Induced Drag
2
Control
K-
CDdi
|u|
Abs
K-
CDde
|u|
Abs1
3
CL
CL^2
K-
CDda
|u|
Abs2
CL0
CL min drag
K-
CDdr
|u|
Abs3
4
Mach
K-
CDM
1
CD

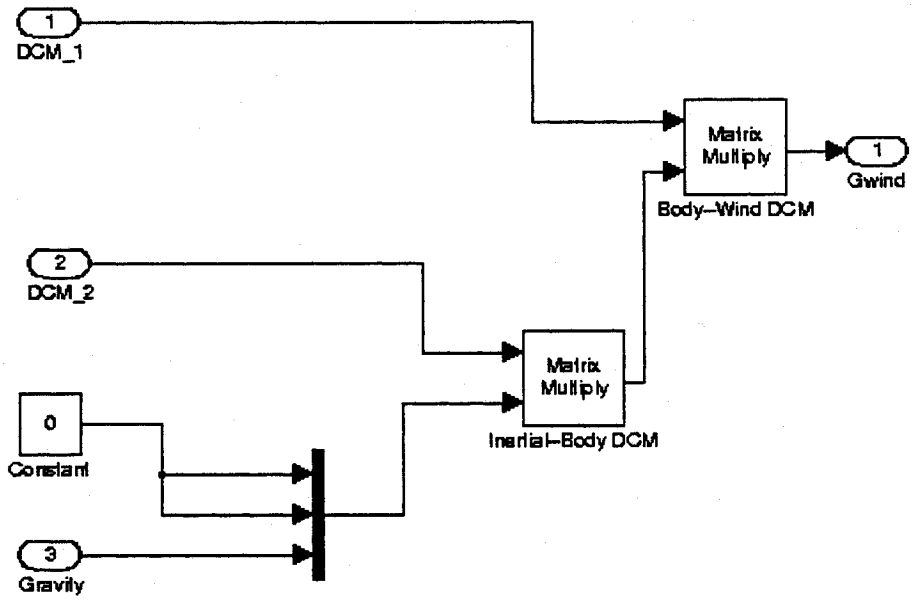

Sources
Pitch moment component build-up
1
WindAxesVel
2
Control
3
AlphaBetaDot
4
AngularRates
5
Mach
Cm0
Cm0
Cmalpha
Cmdl
Cmde
Cmalphadot
Cmq
c
Wing chord
2
2
CmM
1
Cm
Cm*c/(2V)

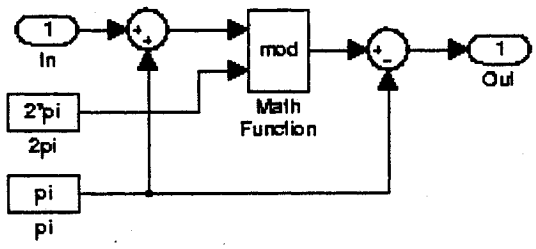

Sources
Roll moment component build-up
1
WindAxesVel
2
Control
3
AngularRates
Clbeta
Clda
Cldr
Clp
Clp
Clr
Clr
b
Wing span
2
2
Cl*b/(2V)
1
Cl

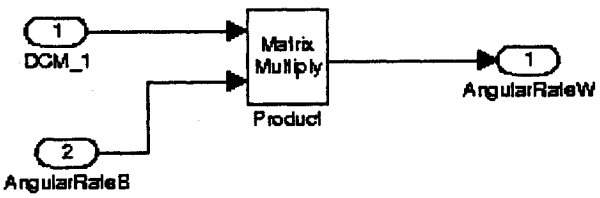


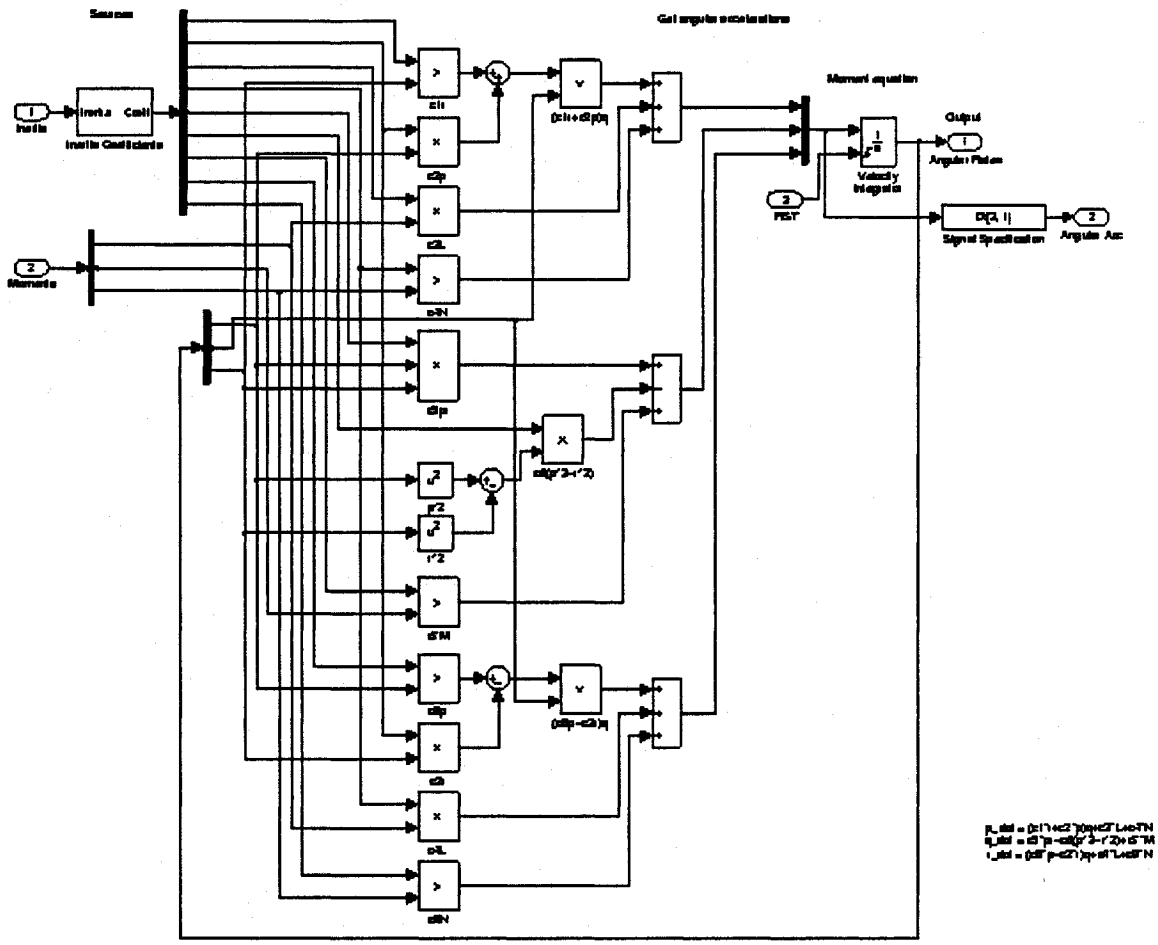



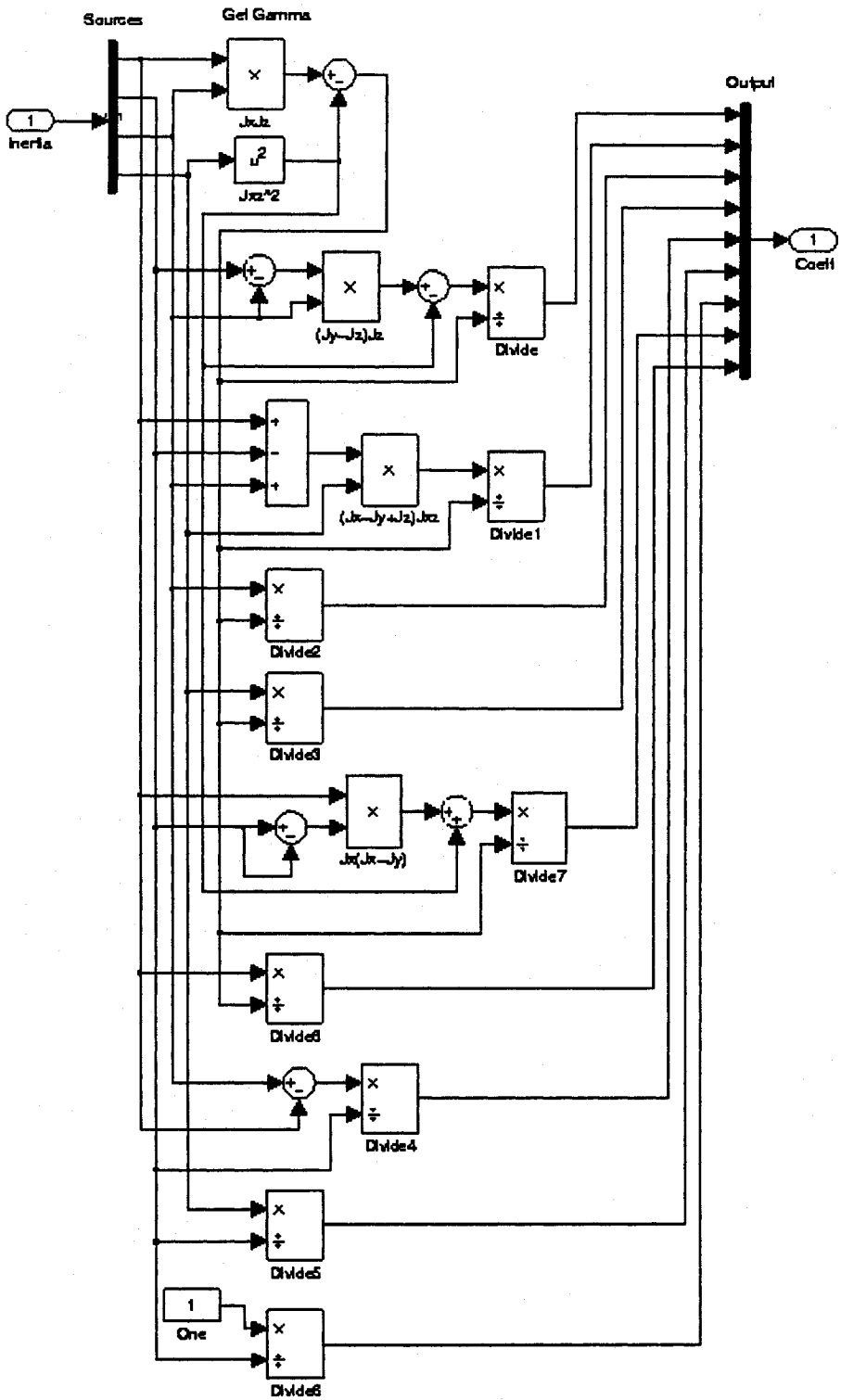

DCM_1: Body Frame —> Wind Frame


1
DCM_1
Matrix
Multiply
1
Gwind
Body–Wind DCM
2
DCM_2
Matrix
Multiply
Inertial–Body DCM
0
Constant
3
Gravity


1
In
2*pi
2pi
pi
pi
mod
Math
Function
1
Out


1
DCM_1
2
AngularRateB
Matrix
Multiply
Product
1
AngularRateW




Sources
Gel Gamma
Output
1
Inertia
Jx.Jz
Jxz^2
(Jy-Jz).Jx
Divide
(Jx-Jy+Jz).Jxz
Divide1
Divide2
Divide3
Jx(Jx-Jy)
Divide7
Divide8
Divide4
Divide5
1
One
Divide6
1
Coeff