# Control of a PWA System Cascaded with a Nonlinear System: An Application to a UAV Path Following Problem

Samer Shehab

A Thesis

in

The Department

of

Mechanical and Industrial Engineering

Presented in Partial Fulfillment of the Requirements

for the Degree of Master of Applied Science at

Concordia University

Montréal, Québec, Canada

April 2006

*Your file Votre référence*
*ISBN: 0-494-14313-4*
*Our file Notre référence*
*ISBN: 0-494-14313-4*

# ABSTRACT

**Control of a PWA System Cascaded with a Nonlinear System: An Application to a UAV Path Following Problem**

**Samer Shehab**

Over the past few decades, there has been a significant need for the use of Uninhabited Aerial Vehicles (UAVs) in both civilian and military applications. Nowadays, UAVs are becoming increasingly popular to perform hard missions that include surveillance, sensing (of, for example, chemical agents), acquiring weather data, and reconnaissance over oceanic and remote areas. Motivated by such importance of UAV missions, this thesis presents a new control methodology applied to a UAV path following problem in the longitudinal plane. This control methodology takes into account the actuator dynamics that is used to deflect the elevator of the UAV. In particular, it considers a model of the Coulomb friction that exists in the dynamics of the actuator. For controller design purposes, the overall dynamics of the UAV are divided into two sets of dynamics that are in cascade connection. One set of dynamics describes the steering motion of the UAV and another set describes the translational motion of the UAV, where both motions are in the longitudinal plane. Each set is treated separately in the controller design. A piecewise-affine state feedback controller is designed for the dynamics of the steering subsystem of the UAV and a nonlinear controller is designed for the dynamics of the translational velocity subsystem of the UAV. Stability of the novel cascade interconnection of the two subsystems is investigated. Simulation results show the effectiveness of the proposed control methodology.

*To my family*

# ACKNOWLEDGEMENTS

I would like to thank Dr. Rodrigues for guiding and supporting me throughout this project. I also would like to acknowledge the Natural Sciences and Engineering Research Council of Canada (NSERC) for partially funding this research.

To all my lab mates, I had such a beautiful time throughout the course of this project. I thank you for your help and friendship. A special thank you goes to Behzad and Narendra who were always helpful and amusing whenever needed. Last, but not the least, I would like to thank my parents and Zeina for their immense support during this project. Zeina, I appreciate your constant patience and encouragement; you were very understanding and there for me every time I needed you. I am so glad and lucky to have known such a wonderful person like you.

# CONTENTS

# LIST OF FIGURES

# LIST OF TABLES

# NOMENCLATURE

| | |
|---|---|
| $\alpha$ | angle of attack |
| $\beta$ | sideslip angle |
| $\{B\}$ | body-fixed frame |
| $\{E\}$ | Earth-fixed frame |
| $\{F\}$ | Serret-Frenet frame |
| $V_T$ | total velocity |
| $q$ | pitch rate |
| $\theta_v$ | vehicle heading angle with respect to the horizontal plane |
| $\theta_c$ | orientation of $\{F\}$ with respect to the horizontal plane |
| $\delta_e$ | elevator deflection |
| $V_m$ | voltage input to the DC motor |
| $u$ | control input |
| $(x_1, \ldots, x_n)$ | state variables |
| $x = (x_1 \ldots x_n)^T$ | state vector |
| $V(x)$ | Lyapunov function, control Lyapunov function |
| $\dot{V} = \frac{dV}{dt}$ | time derivative of $V$ |
| LQR | linear quadratic regulator |
| GAS | globally asymptotically stable |
| ISS | input-to-state stability |
| PWA | piecewise-affine |
| $\rho$ | air density |

# Chapter 1

# Introduction

## 1.1 Motivation and Background

Over the past few decades, there has been a significant need for the use of Uninhabited Aerial Vehicles (UAVs) in both civilian and military applications. Nowadays, UAVs are becoming increasingly popular to perform hard missions that include surveillance, sensing (for *eg.* chemical agents), acquiring weather data, and reconnaissance over oceanic and remote areas. Uninhabited aircraft have taken many names and forms over their long history. They have been called UAVs, Organic Aerial Vehicles (OAVs), Uninhabited Combat Aerial Vehicles (UCAVs), and Micro Air Vehicles (MAVs), to list only a few. For consistency, the acronym UAV is adopted throughout this thesis, but whichever the acronym used, such vehicles are particularly useful in performing the beforementioned applications because they minimize the risk of loss of human life, and they are able to perform monotonous duties for long periods of time. Furthermore, the generally low cost of UAVs makes them well suited for coordinated activities because a larger number of vehicles is affordable to perform sophisticated tasks that one vehicle alone could not perform. A detailed

history of unmanned aircraft is given by Holder [17], McDaid [30], Munson [32,33], Wagner [52,53], and Werrell [54], to name only a few sources.

The introduction of UAVs into the aviation world has led to extensive research on the design and control of autonomous UAVs to achieve specific mission goals. Of utmost importance in UAV missions is the problem of path planning and path following. Some other problems of particular interest to researchers have been the automatic control of a group of UAVs following a path in formation, as well as the reconfiguration of a UAV formation [6]. Because of the nature of their missions, UAVs might be required to perform certain maneuvers such as vertical loops or changing altitude to avoid a collision with another aircraft. These maneuvers involve large changes in physical variables. The dynamics of the UAV path following problem are nonlinear. Linear controllers can be designed if the nonlinear dynamics are linearized around a certain operating point. The linear controllers are then designed to stabilize the system while working around that operating point. This obviously limits the operation of the system to a small region around the operating point. However, as mentioned above, in most missions of a UAV, maneuvers with large changes in variables such as pitch angle might be necessary. Therefore, for such UAV missions, controller design cannot be handled effectively by linear techniques. In contrast to linear models, piecewise-affine (PWA) models offer a global approximation to a nonlinear system. The basic idea is that the whole state space is partitioned into several regions, each of which has its own affine (linear with offset) model. PWA models can thus be used as a good approximation to complex systems involving nonlinearities. This approximation is in fact exact for many cases of practical interest since a wide variety of nonlinearities in physical systems are actually PWA. For instance, Coulomb friction, the dead-zone phenomena in DC motors (and hydraulic actuators), and the characteristics of a saturated linear actuator are

piecewise-affine. As a step to make the system model more realistic, Coulomb friction and viscous friction are included in the actuator dynamics when formulating the path following problem of the UAV in this thesis. The Coulomb friction is a hard nonlinearity that is PWA; hence the primary motivation for using PWA control in this thesis. PWA controllers have never been used in aircraft systems. Previous work on advanced continuous-time autopilot design has concentrated on other techniques such as adaptive control, neural networks control, gain scheduling, feedback linearization and backstepping control. The next section details previous research in autopilot design.

## 1.2 Literature Review

The design of autopilots for high-performance aircraft operating over a wide range of speeds and altitudes was one of the primary motivations for active research on adaptive control in the early 1950s [18]. This makes adaptive control one of the techniques that has long been used in aircraft control systems. However, adaptive control may exhibit local instability and complex nonlinear behavior when adequate process information is not supplied to the parameter estimator [12]. Moreover, large transients occur when the controller is switched on and the parameters have not yet converged to its desired values. Because of the high transients, input saturation can occur, which degrades performance and can even affect stability. Additionally, the design and analysis of nonlinear adaptive systems is difficult and can lead to relatively expensive solutions in computational terms.

Researchers in artificial neural networks (ANN) argue that most of the problems just mentioned in adaptive control can be mitigated using ANN [1, 21, 51]. Despite their potential in many pattern recognition applications, neural networks

require a great deal of computational effort in order to achieve a good set of final weights. In addition, neural networks are sometimes criticized because of their lack of repeatability when computing the weights for a particular problem, given the high dependence of the final weights on the initial values of the weights. Once the weights are found, another drawback of neural networks for control is that no proof of stability is available. Furthermore, the relations between the input variables and the output variables are not developed by engineering judgment. This inherent "black-box" nature of the operation of neural networks causes many engineers to be reluctant of relying heavily on the results from a system they cannot truly understand nor have intuition to modify.

Another popular method used in flight control for handling parameter variations is gain scheduling [22, 38]. This method partitions the flight envelope into smaller regions, each of which has its own steady state operating point. The dynamics of the aircraft are then linearized around each operating point, and a linear controller for each region is designed. A gain-scheduler is then applied to blend the linear controllers together such that the transition between different regions is made smooth. Obviously, one big advantage of the gain-scheduling technique is that it allows the designer to use for each operating point all the classical tools for control design and robustness analysis for linear systems. However, a major drawback of using this technique is that it is often necessary to assume that both the scheduling parameters and their time rate of change are measured because of stability issues. However, this assumption occurs seldom in practice. Moreover, only the nonlinear system behavior in speed and altitude is considered when using this method in flight control; therefore, stability is only guaranteed for low angular rates [29].

Feedback linearization is another control method that uses feedback to linearize the aircraft dynamics globally. A good introduction to feedback linearization can

be found in Slotine and Li [40], and Khalil [24]. Note that the essence of feedback-linearization is to first cancel the nonlinearities appearing in the system dynamics, and then to design a linear controller to stabilize the obtained linear dynamics. Therefore, this method relies on the complete knowledge of the nonlinear plant dynamics. This includes knowledge of the aerodynamic forces and moments, which in practice are never known without uncertainty. So even though this technique potentially allows the use of a single linear controller for the whole flight regime, feedback linearization depends on full knowledge of the aerodynamic coefficients to completely cancel the nonlinear dynamics. Another disadvantage of using feedback linearization is that some nonlinearities act as stabilizing to the system and can contribute to faster convergence. Therefore, leaving such nonlinearities in the system can have beneficial effects on system's response and on the control magnitude [25], [14].

Whereas feedback linearization control requires precise models and often cancels useful nonlinearities, backstepping control design offers a choice of design tools for handling uncertain nonlinearities and can avoid wasteful cancellations of nonlinearities. Backstepping uses a systematic approach for the construction of feedback control laws as well as Lyapunov functions. This reduces the difficulties encountered in obtaining a control Lyapunov function (*clf*) for higher-order systems. Backstepping is a recursive design procedure where the main idea is to let certain states act as a virtual control to others [13, 14, 25]. A major drawback in backstepping control is that the system has to be in a lower triangular feedback form for the recursive design procedure to be applicable. Fontaine and Kokotović, however, were able to improve this procedure by proposing what is called dynamic backstepping [10]. Their method considers a second order system, which is a cascade of one subsystem with a feedforward term and one subsystem with a feedback term. Their design

results in a dynamic control law; hence, the term dynamic backstepping is used. The second order system can then be embedded into a higher order system by cascading it with a lower triangular subsystem to which backstepping is applicable. Although dynamic backstepping extends the class of systems to which backstepping is applicable, its applicability to only a certain form of second order systems still presents some limitations on using this method. Another issue with backstepping control is the selection of the controller parameters. These parameters are tuned to achieve good system response and performance. However, no systematic or optimal way has been suggested for their selection.

Related to autopilot design, the problem of path following for autonomous vehicles has received a significant attention in the past decade. For example, in [37] trajectory tracking of UAVs is addressed using the tracking-error model presented in [23] where the equations of motion are expressed with respect to a fixed reference frame. The approach in this thesis will however be significantly different. It will use the path parameterization method suggested in [43] to transform the problem coordinates to an error space. This error space is formed by the distance between the UAV and a reference point to be tracked on the desired path and the velocity heading error. In [43], Soeanto *et al.* have proposed a parameterization method that allows the rate of progression of a virtual target along the path to be an extra design parameter. This extra degree of freedom overcomes singularity problems that may arise when the position of the virtual target is defined by the projection of the actual vehicle on that path, as it was done in [31]. Furthermore, global convergence of the actual vehicle trajectory to the desired path can be achieved using the method from [43].

## 1.3 Proposed Methodology

As mentioned previously, the Coulomb friction in the actuator dynamics is a PWA nonlinearity. This urges the use of PWA control, which requires first to find a PWA approximation of the nonlinear dynamics in the model. However, the path following problem has nonlinear terms that involve the product of two variables that are in the domain of the nonlinearity. Approximating such nonlinear terms with PWA functions may increase the complexity of the PWA control design. To avoid this, the dynamics of the UAV in the longitudinal plane are divided into two sets: one set describing the translational velocity dynamics of the UAV, and another set describing the steering dynamics of the UAV (the term 'steering dynamics' is chosen to denote the dynamics of the UAV that are used in the controller design to steer the UAV to the desired path in the longitudinal plane). Each set is treated separately in the controller design. Furthermore, the UAV path following problem is modeled in four different cases with increasing level of complexity and, correspondingly, increasing practical relevance. The model in case one has only steering dynamics, but no actuator dynamics are included. In case two, the dynamical model used in case one is augmented to include the dynamics of the actuator that causes the elevator deflection. In cases three and four, the dynamical models used in case one and two, respectively, are augmented to account for the translational velocity dynamics. Therefore, cases three and four have a cascade interconnection between the steering and translational velocity dynamics of the UAV. A nonlinear controller is designed for the translational velocity dynamics in cases three and four to drive the translational velocity to a desired value. On the other hand, a PWA controller is designed for the steering dynamics in all four cases. The objective of the PWA controller is to steer the UAV to follow a specified path in the longitudinal plane. PWA controllers

are scheduled controllers that have the advantage that no assumption on the time rate of change of the scheduling parameters is necessary. Instead of continuously scheduling the gains such as in gain scheduling, PWA controllers switch among a finite number of gains. One of the main advantages of using PWA control is the fact that many of the hard nonlinearities that exist in dynamical systems are actually piecewise-affine. For example, the dead-zone phenomena that exists in a DC motor that is used to actuate the control surfaces in some aircrafts. Due to the static friction at the motor shaft, rotation will only occur if the torque provided by the motor due to some voltage input is sufficiently large.

Dead-zone is a phenomena caused by static friction. Static friction imposes a minimum force or torque required to initiate motion from rest. There are also other types of friction that exist when the system is in motion such as the Coulomb friction and viscous friction. The Coulomb friction is a function of the sign of the velocity, whereas the viscous friction is directly proportional to the velocity and goes to zero at zero velocity. As a step to make the system model more realistic, Coulomb friction and viscous friction are included in the actuator dynamics when formulating the path following problem of the UAV in Chapter 3. Figure 1.1 displays the relationship of Coulomb and viscous friction with respect to the velocity. In the two regions $R_1$ and $R_2$ friction is represented by affine functions. $R_1$ and $R_2$ cover operation with negative and positive velocity, respectively. For more details on modeling friction in physical systems or machines, the reader may refer to [2, 3].

Owing to the above considerations, the advantages of using the proposed PWA control methodology are now summarized.

1. Using knowledge from linear control design. In Section 4.1, a linear controller

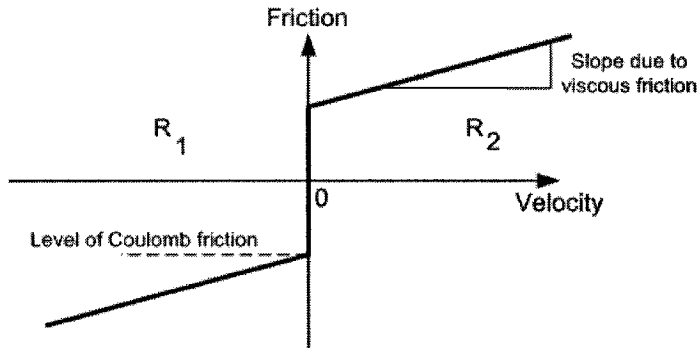


Figure 1.1: Coulomb and viscous friction.

is first designed to stabilize the system in a neighbourhood around the equilibrium point. This controller is then extended to a PWA controller, which covers the whole state-space [39].

2. A systematic way for finding a Lyapunov function that proves global asymptotic stability of the closed-loop system.

3. A systematic way for finding the controller parameters.

4. The method can be extended to output feedback control design [46].

5. PWA models can be regarded as exact models to some hard nonlinearities (dead-zone, actuator saturation, and Coulomb friction) that appear in physical systems, such as in UAV actuator dynamics (DC motor).

## 1.4 Objectives and Thesis Contributions

The objective of this thesis is to design a Lyapunov-based controller to make the UAV follow a desired path in the longitudinal plane. This controller is designed to overcome the Coulomb friction, which is a hard nonlinearity, that exists in the actuator dynamics of the elevator of the UAV.

The main contributions of this thesis are the following:

- The first successful application of the PWA control methodology to a flight control problem.
- Proposing a novel architecture for controller design consisting of a cascade interconnection of a PWA control subsystem (to control the steering dynamics of the UAV) with a nonlinear control subsystem (to control the translational velocity dynamics of the UAV).
- Modeling the Coulomb friction that exists in the actuator dynamics of the elevator in a piecewise affine framework.
- Dividing the path following problem dynamics into four models
    1. Proving stability of the closed-loop system for the first three cases
    2. Showing good control performance through simulation in all four cases

## 1.5 Thesis Outline

The outline of the thesis is as follows:

- **Chapter 1** presents the introduction, objective and contributions, and thesis outline.
- **Chapter 2** describes aircraft dynamics and some of the coordinate systems and angles commonly used to describe the equations of motion of an aircraft.
- **Chapter 3** formulates the model to be used for the path following problem of a UAV in the longitudinal plane and states the assumptions used in that model. The model also incorporates the dynamics of the actuator that is

used to deflect the elevator. Moreover, the path following dynamical model is divided into four models with increasing level of complexity.

- **Chapter 4** proposes a new control design procedure for a system composed of two subsystems that are in cascade connection: one subsystem representing the steering dynamics of the UAV, and another subsystem representing its translational velocity dynamics. The stability of the overall system is also investigated.

- **Chapter 5** presents the simulation results obtained after applying the controllers found in Chapter 4 to the models obtained in Chapter 3.

- **Chapter 6** states the conclusions and future work.

Chapters 3 and 4 are mainly based on the following two conference papers:

1. S. Shehab, and L. Rodrigues, "Preliminary results on UAV path following using piecewise-affine control," *Proceedings of the 2005 IEEE Conference on Control Applications*, pp. 358–363, Toronto, Canada, August 28-31, 2005.

2. S. Shehab, and L. Rodrigues, "UAV Path Following Using a Mixed Piecewise-Affine and Backstepping Control Approach," *Proceedings of the Eighth IASTED International Conference on Control and Applications*, Montréal, Canada, May 24-26, 2006. Accepted for publication.

# Chapter 2

# Modeling Preliminaries: Aircraft Dynamics

This chapter starts by introducing some of the coordinate systems and angles commonly used to describe the equations of motion of an aircraft in flight. Then, the full aircraft dynamics will be introduced. Most of what is presented in this chapter applies to aircraft in general, not only to UAVs. Moreover, the chapter will only cover what is needed in this thesis; [9] and [34] may be referred to for a more detailed explanation of aircraft coordinate systems and dynamics.

## 2.1 Axis-Systems and Aircraft Orientation

Four right-handed axis-systems are commonly used to describe the equations of motion of an aircraft in flight. These are known as the body axis-system, wind axis-system, stability axis-system, and inertia axis-system. In the body axis-system, the axes are fixed to the aircraft and move with it, whereas in the stability axis-system, the x-axis is the perpendicular projection of the relative wind on the plane

of symmetry of the aircraft. In the wind axis-system, on the other hand, the x-axis points into the relative wind. Moreover, the angle between the stability x-axis, $x_s$, and the body x-axis, $x_b$, is defined as the angle of attack $\alpha$, whereas the angle between $x_s$ and the wind x-axis, $x_w$, is called the sideslip $\beta$. The total velocity vector $\vec{V_T}$ points along $x_w$, and is at an angle $\gamma$, called the flight path angle, from the horizontal plane (see Figure 2.1). The inertia axis-system is used when formulating the equations of motion of an aircraft. If Earth is regarded as flat and stationary in inertial space, then any coordinate system attached to it can be considered as an inertial system in which Newton's laws are valid. The Earth with its x-axis pointing to true North, z-axis pointing vertically down, and y-axis completing the right handed triad will be used as an inertial reference frame when deriving the equations of motion of an aircraft in Section 2.2. The Earth reference frame $\{E\}$ as well as the angles $\alpha$, $\beta$, and $\gamma$ are illustrated in Figure 2.1.

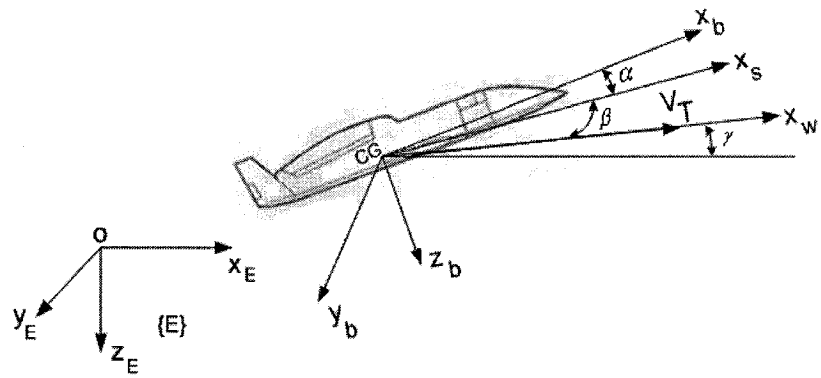


Figure 2.1: Illustration of the angle of attack $\alpha$, the sideslip angle $\beta$, the flight path angle $\gamma$, and the reference frame $\{E\}$.

The orientation of the aircraft relative to the Earth-fixed frame $\{E\}$ can be described using a set of angles called Euler angles. Figure 2.2 shows how to get such orientation from three transformations using $z$-$y$-$x$ Euler angles. $\{E\}$ is first translated parallel to itself until its origin coincides with the origin of $\{B\}$, then it

is rotated through three consecutive rotations performed in the following order:

1. Rotation by an angle $\psi$ around $\mathrm{oz}_E$ to obtain the new axes $\mathrm{ox}_{B_1}$ and $\mathrm{oy}_{B_1}$.

2. Rotation by an angle $\theta$ around $\mathrm{oy}_{B_1}$ to obtain the new axes $\mathrm{ox}_{B_2}$ and $\mathrm{oz}_{B_2}$.

3. Rotation by an angle $\phi$ around $\mathrm{ox}_{B_2}$ to obtain the body-fixed frame $\{B\}$.

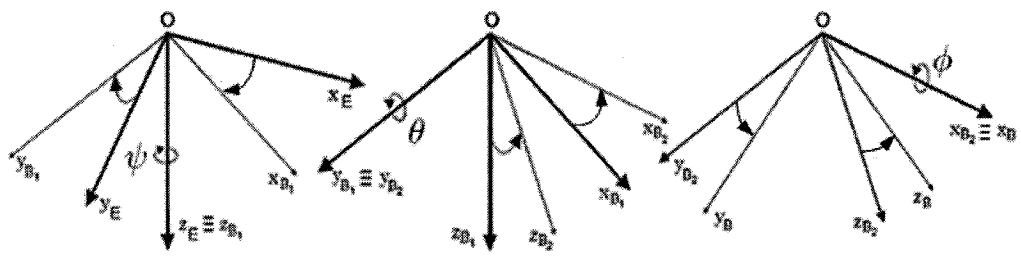


Figure 2.2: $z$-$y$-$x$ Euler angles.

It is often necessary to rotate from $\{E\}$ to $\{B\}$, such as when one wants to obtain the absolute velocity in $\{E\}$ in terms of the velocity components of the aircraft in $\{B\}$ and the Euler angles $(\psi, \theta, \phi)$. This can be achieved by left multiplying the velocity components of the aircraft in $\{B\}$ by the rotation matrix ${}^E_B R$. Based on the rotations illustrated in Figure 2.2, ${}^E_B R$ can be obtained as (see page 102 of reference [34])

$$
{}^E_B R = \begin{bmatrix}
\cos\theta\cos\psi & \sin\phi\sin\theta\cos\psi - \cos\phi\sin\psi & \cos\phi\sin\theta\cos\psi + \sin\phi\sin\psi \\
\cos\theta\sin\psi & \sin\phi\sin\theta\sin\psi + \cos\phi\cos\psi & \cos\phi\sin\theta\sin\psi - \sin\phi\cos\psi \\
-\sin\theta & \sin\phi\cos\theta & \cos\phi\cos\theta
\end{bmatrix}. \tag{2.1}
$$

Another important relationship is the one that relates the time rate of change of the Euler angles $(\dot{\psi}, \dot{\theta}, \dot{\phi})$ with the angular velocity components of the aircraft in $\{B\}$. This can be determined to be [34]

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & \sin\phi\tan\theta & \cos\phi\tan\theta \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi\sec\theta & \cos\phi\sec\theta \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix}, \tag{2.2}$$

where $p$, $q$, and $r$ are the components of the angular velocity vector of the aircraft in the reference frame $\{B\}$.

## 2.2 Equations of Motion

In this section, the previously defined frames $\{E\}$ and $\{B\}$ are used to derive the equations of motion of an aircraft. Before proceeding, however, the following standard assumptions are made:

1. $OX_B$ and $OZ_B$ are in the plane of symmetry of the aircraft.
2. The origin of $\{B\}$ is chosen to coincide with the center of gravity $(CG)$ of the aircraft.
3. The aircraft is a rigid body and has a constant mass.
4. $\{E\}$ is used as the inertial reference coordinate system.

The equations of translational and rotational motion of an airplane can be obtained from Newton's and Euler's laws of motion. Newton's law states that the sum of all external forces acting on a body must be equal to the time rate of change of the linear momentum of its center of mass. Newton-Euler's law of rotational motion, on

the other hand, states that the sum of all the external moments acting on a body must be equal to the time rate of change of its angular momentum. These laws can be represented as the two vector equations

$$\sum \vec{F} = m \left. \frac{d\,\vec{V_T}}{dt} \right|_{\{E\}} \tag{2.3}$$

$$\sum \vec{M}_{CG} = \left. \frac{d\,\vec{H}_{CG}}{dt} \right|_{\{E\}} \tag{2.4}$$

where,

$$\left. \frac{d\,\vec{V_T}}{dt} \right|_{\{E\}} = \left. \frac{d\,\vec{V_T}}{dt} \right|_{\{B\}} + \vec{\omega} \times \vec{V_T} \tag{2.5}$$

$$\vec{H}_{CG} = \int \vec{r} \times (\vec{\omega} \times \vec{r}) dm. \tag{2.6}$$

In (2.6), $\vec{H}_{CG}$ denotes the moment of momentum around the $CG$, $m$ is the mass of the aircraft, and $\vec{r}$ is the moment arm from each element mass $dm$ to the $CG$. The linear velocity $\vec{V_T}$ and angular velocity $\vec{\omega}$ of the aircraft can be written as:

$$\vec{V_T} = u\,\vec{i_b} + v\,\vec{j_b} + w\,\vec{k_b} \tag{2.7}$$

$$\vec{\omega} = p\,\vec{i_b} + q\,\vec{j_b} + r\,\vec{k_b} \tag{2.8}$$

where $\vec{i_b}$, $\vec{j_b}$, and $\vec{k_b}$ are unit vectors along $x_b$, $y_b$, and $z_b$, respectively. The external forces and moments acting on the body of the aircraft are shown in Figure 2.3 and represented as

$$\vec{F} = F_x\,\vec{i_b} + F_y\,\vec{j_b} + F_z\,\vec{k_b} \tag{2.9}$$

$$\vec{M} = M_x\,\vec{i_b} + M_y\,\vec{j_b} + M_z\,\vec{k_b} \tag{2.10}$$

By substituting (2.7) - (2.10) into (2.3) and (2.4) and using the previously mentioned assumptions, the translational and rotational equations of motion are obtained (see [9] and [34] for more details)

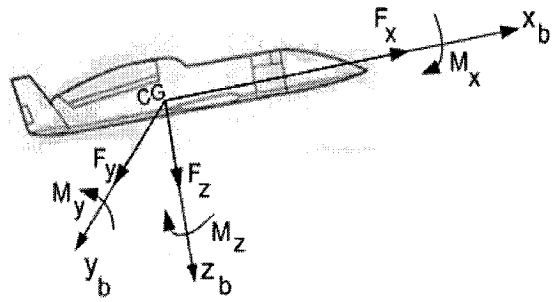


Figure 2.3: External forces and moments.

Translational equations of motion:

$$F_x = m(\dot{u} + qw - vr) \tag{2.11a}$$

$$F_y = m(\dot{v} + ru - pw) \tag{2.11b}$$

$$F_z = m(\dot{w} + pv - qu) \tag{2.11c}$$

Rotational equations of motion:

$$M_x = J_x\dot{p} - J_{xz}\dot{r} + (J_z - J_y)qr - J_{xz}pq \tag{2.12a}$$

$$M_y = J_y\dot{q} + (J_x - J_z)pr + J_{xz}(p^2 - r^2) \tag{2.12b}$$

$$M_z = J_z\dot{r} - J_{xz}\dot{p} + (J_y - J_x)pq + J_{xz}qr \tag{2.12c}$$

The force components $F_x$, $F_y$, and $F_z$, consist of gravity, aerodynamic forces such as lift and drag, and the engine thrust, which is used as a control force. The moment components $M_x$, $M_y$, and $M_z$ consist of moments around the $CG$ caused by the external forces. Of particular interest are the moments caused by deflecting the control surfaces on an aircraft in order to control its attitude. For example, one of the control surfaces is the elevator at the tail of the aircraft, which provides control of the angle $\theta$ (pitch attitude) of the aircraft in the vertical plane. The parameters $J_x$, $J_y$, and $J_z$ are the mass moments of inertia of the aircraft about its $x$, $y$, and

$z$ axes, respectively, and $J_{xz}$ is called the product of inertia[1]. Both the moments of inertia and the product of inertia depend on the shape of the aircraft and the manner in which its mass is distributed. These parameters are given by

$$J_x = \int (y^2 + z^2) dm \tag{2.13a}$$

$$J_y = \int (x^2 + z^2) dm \tag{2.13b}$$

$$J_z = \int (x^2 + y^2) dm \tag{2.13c}$$

$$J_{xz} = \int (xz) dm \tag{2.13d}$$

## 2.3 Longitudinal Motion

Aircraft dynamics are often divided into two important motions, the longitudinal motion and the lateral motion. Longitudinal motion occurs in the plane of symmetry, or vertical plane, of the aircraft. Lateral motion, such as yawing and rolling, displace the plane of symmetry. A very common approach to flight control design is to control the longitudinal motion and the lateral motion separately. This separation of equations allows the designer to reduce the complexity of the control design. Therefore, equations (2.11)-(2.12) are broken down into two sets of equations: one set for the longitudinal motion and another for the lateral motion of the aircraft. The UAV path following problem presented in Chapter 3 will only concentrate on the longitudinal motion for which the aircraft only moves in its $xz$ plane. In other words, there are no side forces, no roll motion around $x_b$, and no yaw motion around $z_b$. This means that $p = r = v = 0$. Taking this into account, the equations of motion become

[1]There also exist $J_{xy}$ and $J_{yz}$ as products of inertia. However, since the $x - z$ plane of the aircraft is assumed to be a plane of symmetry, these two products of inertia are equal to zero (see [9]).

Longitudinal equations of motion:

$$\dot{u} = \frac{1}{m}(F_x) - qw \tag{2.14a}$$

$$\dot{w} = \frac{1}{m}(F_z) + qu \tag{2.14b}$$

$$\dot{q} = \frac{M_y}{J_y} \tag{2.14c}$$

In Chapter 3, the UAV path following problem is formulated. A parameterization method for the path to be followed in the longitudinal plane by the UAV is first presented. Then the UAV dynamics, which also include the dynamics of the actuator used to deflect the elevator, are presented.

# Chapter 3

# UAV Path Following Problem Formulation

In certain missions, UAVs might be required to perform maneuvers such as vertical loops or changing altitude to avoid a collision with another aircraft. A collision avoidance incident is illustrated in Figure 3.1.

Figure 3.1: Collision avoidance.

To be able to perform such maneuvers, a path must be planned and a controller that ensures the path is followed must be designed. As explained in the introduction, this controller cannot be linear because of the large changes in the relevant physical variables such as the pitch angle. In this chapter, a path parameterization method is first introduced to transform the path following problem coordinates to an error space. Then the UAV dynamics as well as the dynamics of the actuator that is

used to deflect the elevator are presented. When the Coulomb friction, an inherent PWA hard nonlinearity, is included in the actuator dynamics, PWA control design is the appropriate tool. The controllers designed in the next chapter will be applied to the dynamics presented in this chapter, and simulation results showing a UAV following a circular path in the vertical plane will be shown in Chapter 5. The assumptions listed in Section 2.2 will also be used in this chapter. However, three more assumptions are added in this chapter. These assumptions are:

- **Assumption 1:** the dynamics of the vertical component of the velocity, $w$, are neglected. It is assumed that $w$ is kept small. This assumption will be validated in Chapter 5.
- **Assumption 2:** the moment $M_e$ around $y_b$ caused by the elevator deflection is only a function of the elevator deflection and the UAV velocity.
- **Assumption 3:** the lift component caused by the elevator is neglected (this lift component is small compared to the gravitational force).

## 3.1 Path Parameterization

This section is based primarily on the work described in [43]. For a UAV to be able to follow a path, the distance from the vehicle to the path and the angle between the vehicle's velocity vector and the tangent to the path should be reduced to zero. This suggests that the kinematical model of the UAV be derived with respect to a Serret-Frenet frame $\{F\}$ that moves along the path ($\{F\}$ is defined by two vectors: one vector tangent to the path and another vector normal to the path). Figure 3.2 shows a UAV and a trajectory ($c$) to be followed in the $x$-$z$ vertical plane. Point $P$ is the origin of the Serret-Frenet frame $\{F\}$, which moves along the desired path

of the vehicle (trajectory $(c)$). Point $Q$ is the position of the vehicle. It can either be expressed in the inertial frame $\{E\}$ by the coordinates $\mathbf{q} = [x\ 0\ z]^T$, or in the moving frame $\{F\}$ by the coordinates $\mathbf{r} = [x_1\ 0\ z_1]^T$. The heading angle of the

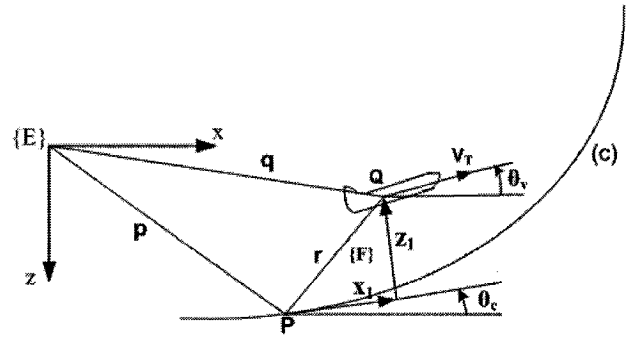


Figure 3.2: Path parameterization description

UAV velocity and the orientation of the Serret-Frenet frame are represented by $\theta_v$ and $\theta_c$, respectively. Both these angles are measured with respect to the x-axis of the inertial frame. Denoting by $s$ the signed curvilinear abscissa of $P$ along the path, define $c_c(s)$ as the path curvature and $\mathbf{t}$ as the tangent vector at point $P$ on the path. For a circular path of radius $R$, $c_c = R^{-1}$ is constant, and the rate of change of $s$ with respect to time, $\dot{s}$, is also constant.

The inertial velocity of point $Q$ expressed in $\{F\}$ is

$$ {}^F_E R \left(\frac{d\mathbf{q}}{dt}\right)_I = \left(\frac{d\mathbf{p}}{dt}\right)_F + \left(\frac{d\mathbf{r}}{dt}\right)_F + (\omega_c \times \mathbf{r})_F , \tag{3.1} $$

where the rotation matrix from $\{E\}$ to $\{F\}$ is

$$ {}^F_E R = \begin{bmatrix} \cos\theta_c & 0 & -\sin\theta_c \\ 0 & 1 & 0 \\ \sin\theta_c & 0 & \cos\theta_c \end{bmatrix} . \tag{3.2} $$

The inertial velocity of $P$ expressed in $\{F\}$ is

$$ \left(\frac{d\mathbf{p}}{dt}\right)_F = \dot{s}\ (\mathbf{t})_F = \begin{bmatrix} \dot{s} & 0 & 0 \end{bmatrix}^T . \tag{3.3} $$

The inertial velocity of $Q$ in $\{E\}$ is

$$\left(\frac{d\mathbf{q}}{dt}\right)_E = \begin{bmatrix} \dot{x} & 0 & \dot{z} \end{bmatrix}^T, \tag{3.4}$$

while the velocity of $Q$ in $\{F\}$ is

$$\left(\frac{d\mathbf{r}}{dt}\right)_F = \begin{bmatrix} \dot{x}_1 & 0 & \dot{z}_1 \end{bmatrix}^T. \tag{3.5}$$

Since $\dot{\theta}_c = c_c(s)\dot{s}$, the cross product term of (3.1) is

$$(\omega_c \times \mathbf{r})_F = \begin{bmatrix} 0 \\ c_c(s)\dot{s} \\ 0 \end{bmatrix} \times \begin{bmatrix} x_1 \\ 0 \\ z_1 \end{bmatrix} = \begin{bmatrix} c_c(s)\dot{s}z_1 \\ 0 \\ -c_c(s)\dot{s}x_1 \end{bmatrix}. \tag{3.6}$$

Combining and rearranging equations (3.2)–(3.6), equation (3.1) becomes

$$\dot{x}_1 = \dot{x}\cos\theta_c - \dot{z}\sin\theta_c - c_c(s)\dot{s}z_1 - \dot{s} \tag{3.7}$$

$$\dot{z}_1 = \dot{x}\sin\theta_c + \dot{z}\cos\theta_c + c_c(s)\dot{s}x_1 \tag{3.8}$$

The inertial velocity of point $Q$ is

$$\begin{bmatrix} \dot{x} \\ \dot{z} \end{bmatrix} = V_T \begin{bmatrix} \cos\theta_v \\ -\sin\theta_v \end{bmatrix}, \tag{3.9}$$

where $V_T$ is the body-axis speed of the UAV. Defining now $\theta = \theta_v - \theta_c$ and substituting (3.9) into (3.7) and (3.8) yields

$$\dot{\theta} = q - c_c(s)\dot{s} \tag{3.10a}$$

$$\dot{x}_1 = -c_c(s)\dot{s}z_1 + V_T\cos\theta - \dot{s} \tag{3.10b}$$

$$\dot{z}_1 = c_c(s)\dot{s}x_1 - V_T\sin\theta, \tag{3.10c}$$

where $q = \dot{\theta}_v$ is the pitch rate of the UAV. Next, the path following dynamics of the UAV in longitudinal motion are presented.

## 3.2 UAV Dynamics

Assumption 1 is used in this section. Even though it is not valid for a general trajectory, this is a simplifying assumption for this research and is a reasonable assumption for small angles of attack. The UAV in longitudinal motion is shown in Figure 3.3. $F_T$ is the engine thrust, and $D$ is the drag force which is opposite in direction to the velocity of the UAV. The moment $M_e$ is assumed to be only a function of the elevator deflection $\delta_e$ and the velocity $V_T$ (Assumption 2). Moreover, the the lift component caused by the elevator is neglected (Assumption 3). Such assumption is particularly good for aircrafts with tails (see page 33 of reference [9]).

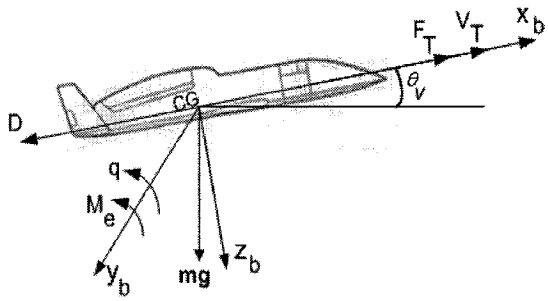


Figure 3.3: UAV in longitudinal motion.

The time rate of change of the velocity $V_T$ is described by

$$m\dot{V}_T = F_T - D - mg\sin\theta_v, \tag{3.11}$$

where $m$ is the mass of the UAV, and $g$ is the gravity. The drag force $D$ is given by

$$D = \frac{1}{2}C_D\rho S V_T^2, \tag{3.12}$$

where $\rho$ is the air density, $C_D$ is the drag coefficient, which is assumed to be constant, and $S$ is the wing area [9].

The time derivative of the pitch rate is given by

$$\dot{q} = \frac{M_e}{J_y} \tag{3.13}$$

Equation (3.13) can also be expressed in terms of $\delta_e$ and $V_T$ as

$$\dot{q} = \frac{1}{2}\frac{C_{L_t}\rho S_t V_T^2 x_e}{J_y}\delta_e, \tag{3.14}$$

where $C_{L_t}$ is the tail lift coefficient, $S_t$ is the tail planform area, $x_e$ is the moment arm from the elevator center of pressure to the $CG$, and $J_y$ is the moment of inertia of the UAV around the $y_b$ axis. The next section will describe the dynamics of the actuator that causes the deflection $\delta_e$.

### 3.2.1 Actuator Dynamics

A DC motor is used to provide the elevator deflection $\delta_e$. The actuator dynamics are thus formed by modeling the DC motor as well as the mechanical connections between the output shaft of the motor and the elevator. The mechanical connections are formed by a small disk mounted on the motor output shaft with radius $r$ and with negligible moment of inertia. This disk is connected to the elevator by a mechanical rod with negligible mass and negligible moment of inertia. A schematic of the elevator and actuator is shown in Figure 3.4, where $V_m$ denotes the voltage applied to the armature of the DC motor, $J_m$ is the rotor moment of inertia of the DC motor, $\omega_m$ is the angular speed at which the output shaft of the motor is rotating, $T_m$ is the motor torque, $b_m$ is the torsional viscous damping of the motor, and $F_c$ is the Coulomb friction torque. $J_e$ is the moment of inertia of the elevator. $H_e$ is the elevator hinge moment, which is an aerodynamic load that the motor torque has to overcome to produce $\delta_e$. $H_e$ is defined as [9] by equation 3.15.

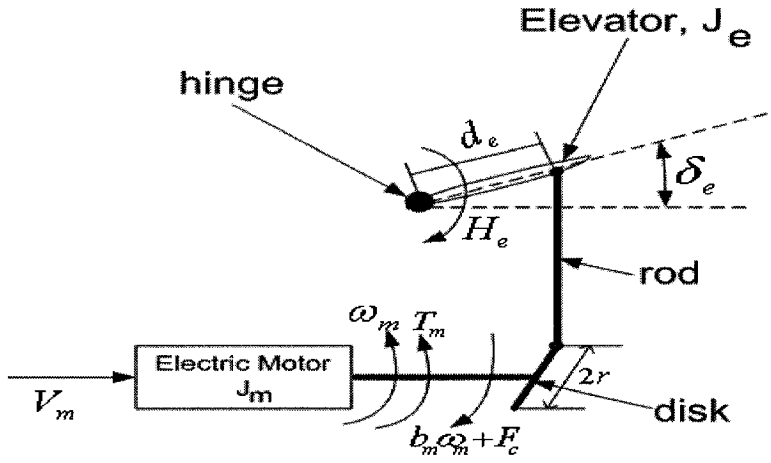


Figure 3.4: Schematic of the actuator and elevator.

$$H_e = \frac{1}{2} C_{h_e} \rho S_e V_T^2 c_e, \tag{3.15}$$

where $C_{h_e}$ is the coefficient of the elevator hinge moment, which can be expressed as $C_{h_e} = b\delta_e$, where b is a constant, $S_e$ is the area of the portion of the elevator that lies aft of the elevator hinge line, and $c_e$ is the chord measured from the elevator hinge line to the trailing edge of the elevator.

By modeling the DC motor[1] with its external loads, the following dynamics for $\omega_m$ and $\delta_e$ are obtained

$$\left( J_m + \left( \frac{r}{d_e} \right)^2 J_e \right) \dot{\omega}_m + b_m \omega_m + \frac{r}{d_e} H_e + F_c sgn(\omega_m) = k_m i_a \tag{3.16a}$$

$$\dot{\delta}_e = \frac{r}{d_e} \omega_m, \tag{3.16b}$$

where $d_e$ is the distance shown in Figure 3.4, $k_m$ is the motor constant, $i_a$ is the armature current, and $k_m i_a$ gives the motor torque $T_m$.

[1]The modeling of a DC motor can be found in most books on modeling of physical systems such as [7] and [11].

The rate of change of $i_a$ is given by the following electrical equation

$$L_a \frac{di_a}{dt} + R_a i_a = V_m - K_e \omega_m, \tag{3.17}$$

where $k_e$ is the back electromotive-force constant, $L_a$ is the armature inductance, and $R_a$ is the armature resistance. The electrical dynamics of the motor are usually fast as compared to the mechanical dynamics and can be neglected. Thus, the inductance $L_a$ in (3.17) can be neglected. Taking this into account and substituting the value of $i_a$ from (3.17) into (3.16a), the actuator dynamics are obtained

$$\left( J_m + \left( \frac{r}{d_e} \right)^2 J_e \right) \dot{\omega}_m = - \left( b_m + \frac{k_e k_m}{R_a} \right) \omega_m - \frac{r}{d_e} H_e - F_c sgn(\omega_m) + \frac{k_m}{R_a} V_m \tag{3.18a}$$

$$\dot{\delta}_e = \frac{r}{d_e} \omega_m, \tag{3.18b}$$

For reasons that will be explained in the next section, the dynamics of the UAV in the longitudinal plane are divided into two sets of dynamics: the translational velocity dynamics and the steering dynamics. Moreover, four different cases are presented, each of which has its own dynamical model and assumptions.

### 3.2.2 Translational Velocity and Steering Dynamics

As mentioned earlier, given the hard nonlinearity in the actuator dynamics that is used to deflect the elevator, a PWA controller is sought for the UAV path following problem. For this controller design, a PWA approximation of the nonlinear dynamics has to be found first. However, equations (3.10b) and (3.10c) have nonlinear terms that involve the product of the velocity $V_T$ with a nonlinear function in the variable $\theta$, such as the term $V_T \cos\theta$ in (3.10b). Approximating such nonlinear terms with

PWA functions may increase the complexity of the PWA control design because there are two variables in the domain of the nonlinearity. To avoid this, the dynamics of the UAV are divided into two sets: one set describing the steering dynamics of the UAV in the longitudinal plane, and another set describing the translational velocity dynamics of the UAV. Each set will be treated separately in the controller design. Such separation of dynamics has been used in other control methods. For instance, in [14, 15] backstepping control is used with the assumption that the time derivative of the aircraft speed is neglected. In other words, $V_T$ was assumed to be constant. In [13], on the other hand, a backstepping design for flight path angle control is presented where the airspeed control is handled separately using results from the literature on linear quadratic control design and $\mu$-analysis. To use the backstepping control design procedure, it was necessary to separate the dynamics of airspeed in [13–15].

In this thesis, the UAV path following problem is modeled in four different cases with increasing level of complexity and, correspondingly, increasing practical relevance. In the first case, the velocity $V_T$ is assumed to be constant and equal to $V_{T_{des}}$. Moreover, no actuator dynamics are considered, and the moment $M_e$ in (3.13) is taken as the control input. In case two, the model used in case one is augmented to account for the actuator dynamics given by (3.18), and the input voltage to the DC motor, $V_m$, is taken as the control input to the model. Moreover, $M_e$ is replaced by its expression that is a function of the elevator deflection and the velocity. Thus, equation (3.14) is used instead of (3.13). In case three, the dynamical model used in the first case is cascaded with the dynamics of the velocity $V_T$ given by (3.11), and the engine thrust force $F_T$ is used to control the velocity $V_T$ to a desired value. The dynamics in case four, on the other hand, will be formed by cascading the dynamical model used in case two with (3.11), and similar to case three, the engine thrust force

$F_T$ is used to control the velocity $V_T$ to a desired value. The next chapter presents the control design for each of the four cases, as well as a stability proof for each of the first three cases. Chapter 5 will present simulation results for each of the four cases showing the effectiveness of the proposed control methodology. Table 3.1 summarizes the level of complexity encountered in each of the four cases.

Table 3.1: Four cases with increasing level of complexity

| | **Actuator dynamics** | **Cascaded system** |
|---|---|---|
| **Case 1** | No | No |
| **Case 2** | Yes | No |
| **Case 3** | No | Yes |
| **Case 4** | Yes | Yes |

Now, the dynamical model used in each case is presented.

**Case one:**

In this case, $V_T$ is assumed to be equal to $V_{T_{des}}$ and $M_e$ is used as the control input to steer the UAV to the desired path. Thus, the UAV path following model will only have steering dynamics given by

- *Steering dynamics:*

$$\dot{q} = \frac{M_e}{J_y} \tag{3.19a}$$

$$\dot{\theta} = q - c_c(s)\dot{s} \tag{3.19b}$$

$$\dot{x}_1 = -c_c(s)\dot{s}z_1 + V_{T_{des}}\cos\theta - \dot{s} \tag{3.19c}$$

$$\dot{z}_1 = c_c(s)\dot{s}x_1 - V_{T_{des}}\sin\theta. \tag{3.19d}$$

**Case two:**

Here the actuator dynamics are added to the steering dynamics of case one, and $V_m$, the voltage input to the DC motor, is taken as the control input to the system. Let

$K_{elevator} = C_{L_t}\rho S_t x_e$, $J_{m_{total}} = J_m + \left(\frac{r}{d_e}\right)^2 J_e$, $c_1 = \frac{r}{d_e}\frac{1}{2}b\rho S_e c_e$, $c_2 = \frac{k_e k_m}{R_a}$, $c_3 = \frac{k_m}{R_a}$, and $c_4 = \frac{r}{d_e}$. The model used in this case is as follows

- *Steering dynamics:*

$$\dot{q} = \frac{1}{2}\frac{K_{elevator}V_T^2}{J_y}\delta_e \tag{3.20a}$$

$$\dot{\theta} = q - c_c(s)\dot{s} \tag{3.20b}$$

$$\dot{\delta}_e = c_4\omega_m \tag{3.20c}$$

$$\dot{\omega}_m = \frac{1}{J_{m_{total}}}\left[-(b_m + c_2)\omega_m - c_1 V_T^2\delta_e - F_c sgn(\omega_m)\right] + \frac{c_3}{J_{m_{total}}}V_m \tag{3.20d}$$

$$\dot{x}_1 = -c_c(s)\dot{s}z_1 + V_T\cos\theta - \dot{s} \tag{3.20e}$$

$$\dot{z}_1 = c_c(s)\dot{s}x_1 - V_T\sin\theta. \tag{3.20f}$$

**Case three:**

In this case the steering dynamics are the same as in case one with $M_e$ being the control input. However, the dynamics of the velocity $V_T$ are added. Therefore, the model used in this case has steering and translational velocity dynamics as follows

- *Steering dynamics:*

$$\dot{q} = \frac{M_e}{J_y} \tag{3.21a}$$

$$\dot{\theta} = q - c_c(s)\dot{s} \tag{3.21b}$$

$$\dot{x}_1 = -c_c(s)\dot{s}z_1 + V_T\cos\theta - \dot{s} \tag{3.21c}$$

$$\dot{z}_1 = c_c(s)\dot{s}x_1 - V_T\sin\theta. \tag{3.21d}$$

- *Translational velocity dynamics:*

$$\dot{V}_T = \frac{1}{m}(F_T - D - mg\sin\theta_v). \tag{3.22}$$

**Case four:**

Here the steering dynamics in case two are augmented to include the translational velocity dynamics. Thus, this case includes both steering and translational velocity dynamics as follows

- *Steering dynamics:* same as (3.20)
- *Translational velocity dynamics:* same as (3.22)

For cases three and four, Figure 3.5 displays a block diagram showing the interconnection between the steering and translational velocity dynamics.

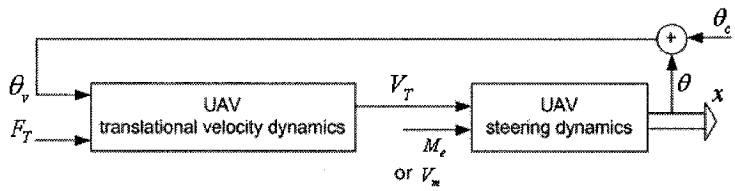


Figure 3.5: Block diagram for the cascaded system

In sum, this chapter presented the steering and translational velocity dynamics of the UAV path following problem in the longitudinal plane. Moreover, four different cases with increasing level of complexity have been modeled. Chapter 4 will include the synthesis of a PWA state-feedback controller and the design of a nonlinear controller that are used to control the steering motion and translational motion of the UAV in the longitudinal plane, respectively. Chapter 4 will also present stability results for the overall cascaded system (the steering dynamics cascaded with the translational velocity dynamics) for case three.

# Chapter 4

# Controller Synthesis

In Chapter 3, it was shown how the dynamics of the UAV path following problem can be divided into two sets of equations: one set describing the steering dynamics of the UAV, and another set describing the translational velocity dynamics of the UAV in the longitudinal plane. Moreover, the UAV path following problem was modeled for four different cases. This chapter will present the design of a PWA state-feedback controller that is used in each of the four cases to drive the state vector $x$ of the steering dynamics to its closed-loop equilibrium point $x_{cl}$. A nonlinear controller design will also be presented in this chapter to find the thrust force $F_T$ that can make the velocity $V_T$ converge to its desired value $V_{T_{des}}$ for cases three and four. Finally, the stability of the closed-loop system in cases one, two, and three will be proved. Figure 4.1 shows a block diagram with the interconnection between each of the closed-loop systems of the steering and translational velocity dynamics. For case three, the input to the PWA system is $M_e$, whereas for case four it is the voltage input of the DC motor.

The controller synthesis presented in this thesis is Lyapunov-based where a candidate control Lyapunov function (an energy-like function) is searched for. This

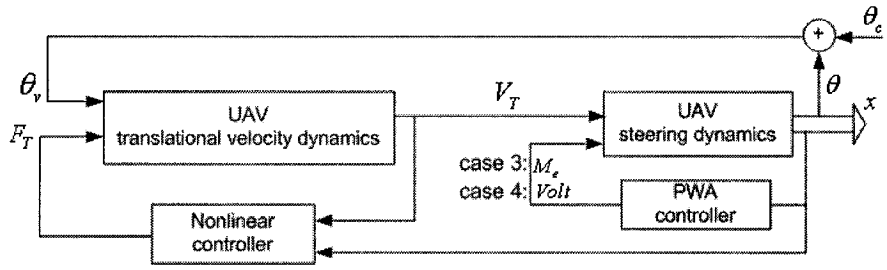


Figure 4.1: Interconnection between the two closed-loop subsytems.

function is a scalar function and is denoted by $V(x)$, where $x$ is the state vector of the system. $V(x)$ can be thought of as representing the energy contained in the system. Intuitively speaking, if energy is dissipated from the system, then the system itself must be moving towards the equilibrium. This approach of proving stability is called Lyapunov's direct method (or second method), which can be found in any introductory textbook on nonlinear control, such as those by Slotine and Li [40], and Khalil [24]. Before proceeding to the formal control design for each of the steering and translational velocity dynamics, let us introduce the following preliminary definitions.

**Definition 1** *a scalar function $V(x)$ is said to be*

- *positive definite if $V(0) = 0$, and $V(x) > 0 \quad for\ x \neq 0$*
- *positive semidefinite if $V(0) = 0$, and $V(x) \geq 0 \quad for\ x \neq 0$*
- *negative definite if $-V(x)$ is positive definite*
- *negative semidefinite if $-V(x)$ is positive semidefinite*
- *radially unbounded if $V(x) \rightarrow \infty$ as $||x|| \rightarrow \infty$.*

□

## 4.1 PWA State-Feedback Controller Design (Steering Dynamics)

From Chapter 3, four cases were modeled, each of which has steering dynamics. To be able to find a PWA controller, a PWA approximation of the steering dynamics must be found first. The space of variables in the domain of the nonlinearity should thus be partitioned into simplicial[1] cells. The reader is referred to [45] and the references therein for details on how to obtain such a partition and piecewise-affine approximation of a class of nonlinear systems.

In the PWA controller design, the translational velocity $V_T$ is assumed constant and equal to $V_{T_{des}}$. Thus, the steering dynamics used for the PWA controller design are the same for cases one and three. Additionally, in cases two and four the same steering dynamics are used in the PWA controller design. Let $x = [q \ \theta \ x_1 \ z_1]^T$ be the state vector in (3.21). System (3.21) in state-space form is

$$\begin{bmatrix} \dot{q} \\ \dot{\theta} \\ \dot{x}_1 \\ \dot{z}_1 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -c_c\dot{s} \\ 0 & 0 & c_c\dot{s} & 0 \end{bmatrix} \begin{bmatrix} q \\ \theta \\ x_1 \\ z_1 \end{bmatrix} + \begin{bmatrix} 0 \\ -c_c\dot{s} \\ V_T\cos\theta - \dot{s} \\ -V_T\sin\theta \end{bmatrix} + \begin{bmatrix} \frac{1}{J_y} \\ 0 \\ 0 \\ 0 \end{bmatrix} M_e. \tag{4.1}$$

Now let $x = [q \ \theta \ \delta_e \ w_m \ x_1 \ z_1]^T$ be the state vector in (3.20). System (3.20) in state-space form is

[1]The convex hull of a set $\mathcal{S}$ is the smallest convex set that contains $\mathcal{S}$. A simplex in $\mathbb{R}^n$ is defined as the convex hull of $n+1$ affinely independent points. For example, a simplex in $\mathbb{R}$ is an interval, and in $\mathbb{R}^2$ is a triangle.

$$\begin{bmatrix} \dot{q} \\ \dot{\theta} \\ \dot{\delta}_e \\ \dot{w}_m \\ \dot{x}_1 \\ \dot{z}_1 \end{bmatrix} = \begin{bmatrix} 0 & 0 & \frac{1}{2}\frac{K_{elevator}V_T^2}{J_y} & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & c_4 & 0 & 0 \\ 0 & 0 & \frac{-c_1 V_T^2}{J_{m_{total}}} & \frac{-(b_m+c_2)}{J_{m_{total}}} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -c_c\dot{s} \\ 0 & 0 & 0 & 0 & c_c\dot{s} & 0 \end{bmatrix} \begin{bmatrix} q \\ \theta \\ \delta_e \\ w_m \\ x_1 \\ z_1 \end{bmatrix} + \begin{bmatrix} 0 \\ -c_c\dot{s} \\ 0 \\ \frac{-F_c sgn(\omega_m)}{J_{m_{total}}} \\ V_T\cos\theta - \dot{s} \\ -V_T\sin\theta \end{bmatrix} + \begin{bmatrix} \frac{c_3}{J_{m_{total}}} \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} V_m \tag{4.2}$$

Three nonlinear functions are found in the steering dynamics of the UAV. These are the sine and cosine functions, which are functions of the state $\theta$, and the Coulomb friction, which exists in the actuator dynamics and is a function of the sign of the angular velocity $\omega_m$. The Coulomb friction function, $F_c sgn(\omega_m)$, is already a PWA function defined based on the state $\omega_m$. This was illustrated previously in Chapter 1. The sine and cosine functions, on the other hand, must be approximated by PWA functions. The grid chosen to approximate the nonlinear functions by PWA functions based on the state variable $\theta$ is taken as

$$\theta_i = \left\{-\pi, -\frac{\pi}{2}, -\frac{\pi}{4}, -\frac{\pi}{30}, \frac{\pi}{30}, \frac{\pi}{4}, \frac{\pi}{2}, \pi\right\}, \quad \text{for } i = 1, \ldots, 8.$$

Figure 4.2 shows the PWA approximation of $sin\theta$ and $\cos\theta$ using the above grid. After approximating the sine and cosine by PWA functions, the resulting approximate dynamics for each of (4.1) and (4.2) will then be PWA described by

$$\dot{x}(t) = A_i x(t) + a_i + B_i u(t), \quad \text{for } x(t) \in \mathcal{R}_i, \tag{4.3}$$

where matrices $A_i \in \mathbb{R}^{n\times n}$, $a_i \in \mathbb{R}^n$ and $B_i \in \mathbb{R}^{n\times m}$ are constant within each $\mathcal{R}_i$. The polytopic cells, $\mathcal{R}_i$, $i \in \mathcal{I} = \{1, \ldots, M\}$, partition the state space $\mathcal{X} \subset \mathbb{R}^n$ such that $\mathcal{R}_i \cap \mathcal{R}_j = \emptyset$, $i \neq j$ and $\cup_{i=1}^{M} \overline{\mathcal{R}}_i = \mathcal{X}$. Following [16, 19, 35], each cell

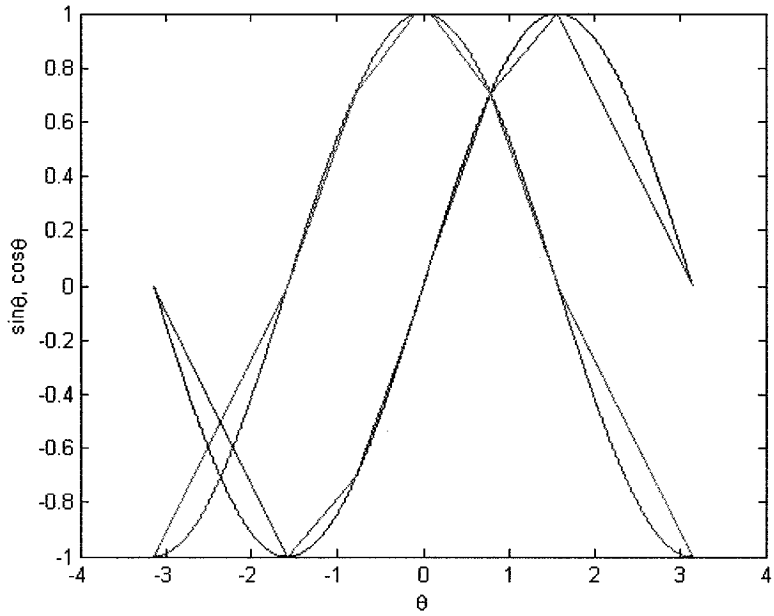


Figure 4.2: PWA approximation of $sin\theta$ and $\cos\theta$.

is constructed as the intersection of a finite number $(p_i)$ of half spaces and can be expressed as

$$\mathcal{R}_i = \{x \mid \bar{E}_i \bar{x} > 0\}, \tag{4.4}$$

where $\bar{x} = \begin{bmatrix} x^T & 1 \end{bmatrix}^T$ and $\bar{E}_i \in \mathbb{R}^{p_i \times (n+1)}$.

Any two cells sharing a common facet will be called *level-1* neighboring cells. Let $\mathcal{N}_i = \{$ *level*-1 neighboring cells of $\mathcal{R}_i\}$. A parametric description of the boundaries can then be obtained as

$$\overline{\mathcal{R}}_i \cap \overline{\mathcal{R}}_j \subseteq \{F_{ij} s + f_{ij} \mid s \in \mathbb{R}^{n-1}\}, \tag{4.5}$$

for $i = 1, \ldots, M$, $j \in \mathcal{N}_i$, where $\overline{\mathcal{R}}_i$ denotes the closure of $\mathcal{R}_i$, $F_{ij} \in \mathbb{R}^{n \times (n-1)}$ is a full rank matrix and $f_{ij} \in \mathbb{R}^n$. This parametric description of the boundaries is depicted in Figure 4.3 below. Furthermore, it is assumed that $\mathcal{R}_i$ can be outer approximated

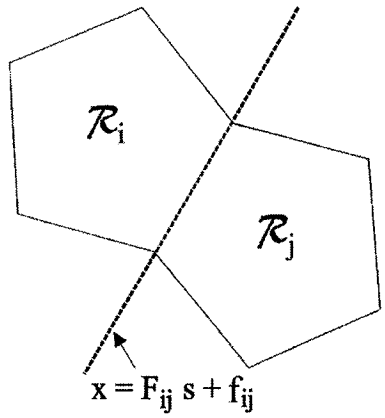


Figure 4.3: Polytopic regions $\mathcal{R}_i$, $\mathcal{R}_j$ and boundary.

by a (degenerate) quadratic curve $\varepsilon_i$ defined as

$$\varepsilon_i = \{x | \bar{x}^T \bar{S}_i \bar{x} > 0\}. \tag{4.6}$$

One possible configuration for $\bar{S}_i$ is [39]

$$\bar{S}_i = \check{E}_i^T \Lambda_i \check{E}_i, \tag{4.7}$$

where $\Lambda_i \in \mathbb{R}^{(p_i+1)\times(p_i+1)}$ is a matrix with nonnegative entries and

$$\check{E}_i = \left[\begin{array}{c|c} 0_{1\times n} & 1 \\ \hline & \\ \bar{E}_i & \end{array}\right] \in \mathbb{R}^{(p_i+1)\times(n+1)}. \tag{4.8}$$

In this case, it is said that $\bar{S}_i$ defines a *bounding ellipsoid* for region $\mathcal{R}_i$. For PWA systems of the form (4.3), we will now review the PWA synthesis algorithm developed in [39, 45]. The goal is to stabilize the equilibrium point $x_{cl}$ for system (4.3) by designing a PWA state feedback control law

$$u = \bar{K}_i \bar{x}, \quad \text{for } x(t) \in \mathcal{R}_i \tag{4.9}$$

where

$$\bar{K}_i = \begin{bmatrix} K_i & m_i \end{bmatrix}, \tag{4.10}$$

and $-\bar{K}_{Lim} \prec \bar{K}_i \prec \bar{K}_{Lim}$, with $\bar{K}_{Lim}$ being a vector of upper bounds for the entries of $\bar{K}_i$, $i = 1, \ldots, M$. Replacing (4.9) into (4.3) yields

$$\dot{\bar{x}}(t) = (\bar{A}_i + \bar{B}_i \bar{K}_i)\bar{x}(t), \quad \text{for } x(t) \in \mathcal{R}_i \tag{4.11}$$

where $\bar{A}_i \in \mathbb{R}^{(n+1)\times(n+1)}$ and $\bar{B}_i \in \mathbb{R}^{(n+1)\times m}$ are

$$\bar{A}_i = \begin{bmatrix} A_i & a_i \\ 0 & 0 \end{bmatrix}, \quad \bar{B}_i = \begin{bmatrix} B_i \\ 0 \end{bmatrix}. \tag{4.12}$$

The controller will be designed by searching for a globally quadratic candidate control Lyapunov function defined as

$$V(x) = \begin{bmatrix} x \\ 1 \end{bmatrix}^T \begin{bmatrix} P & -Px_{cl} \\ -x_{cl}^T P^T & r \end{bmatrix} \begin{bmatrix} x \\ 1 \end{bmatrix} = \bar{x}^T \bar{P} \bar{x}, \tag{4.13}$$

where $P = P^T > 0$, and $\bar{P}$ and $\bar{x}$ are as follows

$$\bar{P} = \begin{bmatrix} P & -Px_{cl} \\ -x_{cl}^T P^T & r \end{bmatrix}$$

$$\bar{x} = \begin{bmatrix} x & 1 \end{bmatrix}^T.$$

To design a stabilizing PWA controller for system (4.3), the following constraints and controller synthesis algorithm are used [39,47].

### Constraints on the Lyapunov function

Positive definiteness of the candidate control Lyapunov function is guaranteed by

$$V(x) > 0, \quad \text{for } x \neq 0,$$

This inequality is satisfied by the following constraint

$$\bar{P} > 0. \tag{4.14}$$

The following inequality ensures the decrease of the candidate control Lyapunov function over time

$$\frac{dV(x)}{dt} < -\alpha V(x), \tag{4.15}$$

where $\alpha \geq 0$ is a bound on the decay rate for the candidate Lyapunov function. Using the description of the cells (4.6) and the $\mathcal{S}-procedure$ [5], it can be shown that sufficient conditions for satisfying the above inequality for each region $\mathcal{R}_i$ are the existence of matrix $\bar{P}$, and matrices $\Lambda_i$ with nonnegative entries satisfying

$$\bar{P}(\bar{A}_i + \bar{B}_i\bar{K}_i) + (\bar{A}_i + \bar{B}_i\bar{K}_i)^T\bar{P} - \alpha\bar{P} + \check{E}_i^T\Lambda_i\check{E}_i < 0, \quad \text{for } i = 1, \ldots, M. \tag{4.16}$$

### Constraint on the Control Input

Continuity of the control input at the boundaries can be enforced by $u_i(x) = u_j(x)$ for $x \in \overline{\mathcal{R}}_i \cap \overline{\mathcal{R}}_j$, i.e.

$$\bar{K}_i\bar{x} = \bar{K}_j\bar{x}, \quad \text{for } x = F_{ij}s + f_{ij}.$$

For this to be satisfied, the following constraint in each region should be considered [39,47]:

$$(\bar{K}_i - \bar{K}_j)\bar{F}_{ij} = 0, \quad \text{for } j \in \mathcal{N}_i, \tag{4.17}$$

where $\bar{F}_{ij}$ is defined as

$$\bar{F}_{ij} = \begin{bmatrix} F_{ij} & f_{ij} \\ 0 & 1 \end{bmatrix}. \tag{4.18}$$

### Desired Closed-Loop Dynamics

The control design will start by searching for a linear local controller to achieve the desired closed-loop dynamics in the region where the closed-loop equilibrium point ($x_{cl}$) is located, $\mathcal{R}_{i^\star}$. Consider the dynamics of the system in this region

$$\dot{x}(t) = A_{i^\star}x(t) + a_{i^\star} + B_{i^\star}u(t), \quad \text{for } x(t) \in \mathcal{R}_{i^\star}. \tag{4.19}$$

Introducing a new variable $z(t) = x(t) - x_{cl}$ yields

$$\dot{z}(t) = A_{i^\star} z(t) + A_{i^\star} x_{cl} + a_{i^\star} + B_{i^\star} u(t). \tag{4.20}$$

We assume that there exists a vector $m_{i^\star}$ satisfying

$$B_{i^\star} m_{i^\star} + A_{i^\star} x_{cl} + a_{i^\star} = 0. \tag{4.21}$$

Thus, by using the control input

$$u(t) = K_{i^\star} z(t) + m_{i^\star}, \tag{4.22}$$

the closed-loop dynamics in region $\mathcal{R}_{i^\star}$ are now

$$\dot{z}(t) = (A_{i^\star} + B_{i^\star} K_{i^\star}) z(t). \tag{4.23}$$

The matrix gain $K_{i^\star}$ can be designed using linear control methodologies to satisfy desired design objectives based on stability or performance measures. In this thesis, a Linear Quadratic Regulator (LQR) is used to find $K_{i^\star}$. The weighting matrix $Q \in \mathbb{R}^{n \times n}$ and the scalar weighting factor $R \in \mathbb{R}$ of the LQR used for the UAV path following problem will be given in Section 4.1.1.

The affine controller for region $\mathcal{R}_{i^\star}$ is thus given by

$$\bar{K}_{i^\star} = \begin{bmatrix} K_{i^\star} & m_{i^\star} \end{bmatrix} \tag{4.24}$$

The LQR in the region holding the equilibrium point was designed to satisfy requirements locally. The closed-loop dynamics of the system in this region can serve as a reference model for the closed-loop dynamics in other regions. Using the method proposed in [39], an upper bound on the norm of the difference between the closed-loop vector field of all regions and that of the region holding the equilibrium point is minimized. This can be formulated as minimizing $\beta > 0$ satisfying

$$\| \bar{A}_i + \bar{B}_i \bar{K}_i - (\bar{A}_{i^\star} + \bar{B}_{i^\star} \bar{K}_{i^\star}) \| < \beta. \tag{4.25}$$

### Controller Synthesis Algorithm

The synthesis algorithm for finding a PWA state-feedback controller for (4.3) is as follows [39]:

1. Design a local linear controller for (4.23) by choosing a controller gain $K_{i^\star}$ for region $\mathcal{R}_{i^\star}$, with $m_{i^\star}$ fixed by (4.21).

2. Given $x_{cl}$ and $\alpha$, fix $K_{i^\star}$ and $m_{i^\star}$, and solve

$$\begin{aligned} \min \quad & \beta \\ s.t. \quad & (4.14), (4.16), (4.17), (4.25), \\ & \beta > 0, \quad P = P^T > 0, \quad \Lambda_i \succ 0, \\ & -\bar{K}_{Lim} \prec \bar{K}_i \prec \bar{K}_{Lim}, \\ & \text{for } i \in \mathcal{I} = \{1, \ldots, M\}, \end{aligned} \tag{4.26}$$

   where $\succ$ and $\prec$ mean component-wise inequalities.

**Remark 1** The solution to this problem will be the PWA controller that minimizes the norm of the difference of the closed-loop vector field of all regions to the closed-loop vector field of the region containing the equilibrium point. □

**Remark 2** The constraints in the synthesis problem (4.26) include a set of Bilinear Matrix Inequalities (BMIs). BMIs are nonconvex constraints and this makes them hard to solve. Several numerical algorithms have been proposed to solve BMI problems locally and the one used here is implemented in the software package PENBMI [26]. □

If (4.26) is successful in finding a controller and a Lyapunov function, the following results can be established to prove stability of the closed-loop system of the steering dynamics given by each of (4.1) and (4.2).

- **Steering dynamics given by (4.1): cases one and three**

**Theorem 1** *For system (4.1), assume the Lyapunov function (4.13) is defined in $\mathcal{X} \subseteq \mathbb{R}^n$. If there is a solution to the design problem (4.26), then the PWA approximate closed-loop system is locally asymptotically stable inside any subset of the largest level set of the control Lyapunov function (4.13) that is contained in $\mathcal{X}$. Furthermore, if $\mathcal{X} = \mathbb{R}^n$ then the asymptotic stability is global.*

**Proof:** First note that the open-loop PWA dynamics are continuous. Furthermore, the $B$ matrix is the same for all regions, and the controller input $u$ is designed to be continuous at the switching boundaries. Thus, the closed-loop vector fields are continuous at the switching boundaries. Therefore, no sliding modes are generated at the switching boundaries. Following [39] and the arguments used in the proof given in [46], if there is a solution to the design problem (4.26), then the PWA approximate closed-loop system is locally asymptotically stable. If $\mathcal{X} = \mathbb{R}^n$, since $V$ is radially unbounded, the asymptotic stability is global. □

- **Steering dynamics given by (4.2): cases two and four**

**Theorem 2** *Assume there are no sliding modes at the switching boundaries. For system (4.2), assume the Lyapunov function (4.13) is defined in $\mathcal{X} \subseteq \mathbb{R}^n$. If there is a solution to the design problem (4.26), then the PWA approximate closed-loop system is locally asymptotically stable inside any subset of the largest level set of the control Lyapunov function (4.13) that is contained in $\mathcal{X}$. Furthermore, if $\mathcal{X} = \mathbb{R}^n$ then the asymptotic stability is global.*

**Proof:** Since it was assumed that there are no sliding modes at the switching boundaries, the proof follows from the proof of the theorem and the arguments in [39, 46]. □

**Remark 3** Finding a global control Lyapunov function with $\dot{V} < 0$ everywhere for all vector fields gives the guarantee that the system is stable for any switching [27]. Alternatively, the conditions in [46] can be used in the synthesis to guarantee that sliding modes are ruled out in closed-loop. □

For conditions under which the controller found also yields stability of the original nonlinear subsystem (steering dynamics) from which (4.3) was approximated, the reader is referred to [47].

### 4.1.1 PWA Controllers

**Steering dynamics given by (4.1):**

The bound $\alpha$ on the decay rate of the candidate Lyapunov function used in the PWA controller design is taken as one. The LQR parameters are as follows:

$$Q = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

and $R = 100$. The globally quadratic Lyapunov function found after solving (4.26) for the dynamics given by (4.1) is shown in Figure 4.4.

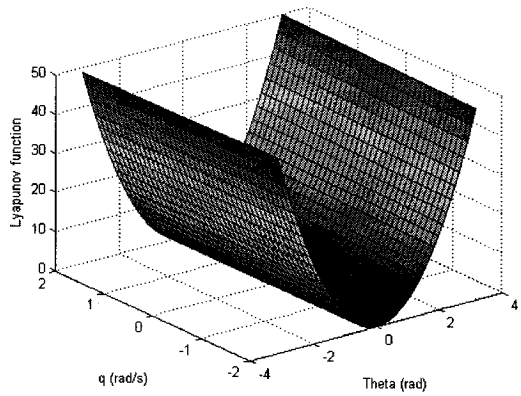


Figure 4.4: Lyapunov function for the steering dynamics of cases one and three.

The PWA controller found is

$$
\begin{aligned}
K_1 &= \begin{bmatrix} -6.3844 & 5.8438 & -0.0477 & 0.1331 \end{bmatrix}, \\
K_2 &= \begin{bmatrix} -6.3844 & -4.3543 & -0.0477 & 0.1331 \end{bmatrix}, \\
K_3 &= \begin{bmatrix} -6.3844 & -0.0829 & -0.0477 & 0.1331 \end{bmatrix}, \\
K_4 &= \begin{bmatrix} -6.3844 & -5.8215 & -0.0477 & 0.1331 \end{bmatrix}, \\
K_5 &= \begin{bmatrix} -6.3844 & -5.8128 & -0.0477 & 0.1331 \end{bmatrix}, \\
K_6 &= \begin{bmatrix} -6.3844 & 1.3957 & -0.0477 & 0.1331 \end{bmatrix}, \\
K_7 &= \begin{bmatrix} -6.3844 & 4.2989 & -0.0477 & 0.1331 \end{bmatrix}, \\
m_1 &= 14.2868, \; m_2 = -1.7323, \; m_3 = 1.6224, \\
m_4 &= 1.0215, \; m_5 = 1.0206, \; m_6 = -4.6410, \\
m_7 &= -9.2013.
\end{aligned}
$$

**Steering dynamics given by (4.2):**

The PWA controller design previously presented in this chapter is based on first designing an affine controller to stabilize the system in a neighborhood around the

equilibrium point, then this controller is extended to PWA controllers, which cover the whole state-space. However, since $w_m = 0$ is the equilibrium value for $w_m$, linearization about the equilibrium point is not possible because of the *sgn* function associated with the Coulomb friction. This was solved by running the algorithm in (4.26) twice. The first run was for positive $w_m$ with the same grid for the angle $\theta$ used in case one to approximate the nonlinear functions sine and cosine. The Lyapunov function found from that run was then fixed and (4.26) was run for the second time, but now for negative $w_m$ and with the same grid for $\theta$ as before. Each run for (4.26) will give seven controllers, yielding 14 controllers in total, seven controllers for positive $w_m$ and other seven controllers for negative $w_m$.

The reason for fixing the Lyapunov function after the first run and using it for the second run of (4.26) is to obtain guaranteed stability of the overall steering system when the controller switches among the controllers designed for positive and negative $w_m$.

The bound $\alpha$ on the decay rate of the candidate Lyapunov function used in the PWA controller design is taken as one. The LQR parameters are as follows:

$$Q = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix},$$

and $R = 100$.

The globally quadratic Lyapunov function found after solving (4.26) for the dynamics given by (4.2) is shown in Figure 4.5.

The PWA controller gains found for positive $w_m$ are

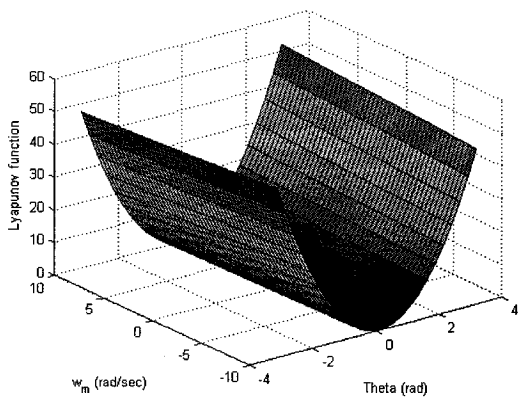


Figure 4.5: Lyapunov function for the steering dynamics of cases two and four.

$$
\begin{aligned}
K_1 &= \begin{bmatrix} -13.4344 & -2.0501 & -1.9875 & -0.5895 & -0.0263 & 0.1389 \end{bmatrix}, \\
K_2 &= \begin{bmatrix} -13.4344 & -1.1486 & -1.9875 & -0.5895 & -0.0263 & 0.1389 \end{bmatrix}, \\
K_3 &= \begin{bmatrix} -13.4344 & 2.1085 & -1.9875 & -0.5895 & -0.0263 & 0.1389 \end{bmatrix}, \\
K_4 &= \begin{bmatrix} -13.4344 & -8.5965 & -1.9875 & -0.5895 & -0.0263 & 0.1389 \end{bmatrix}, \\
K_5 &= \begin{bmatrix} -13.4344 & -1.4632 & -1.9875 & -0.5895 & -0.0263 & 0.1389 \end{bmatrix}, \\
K_6 &= \begin{bmatrix} -13.4344 & 0.7146 & -1.9875 & -0.5895 & -0.0263 & 0.1389 \end{bmatrix}, \\
K_7 &= \begin{bmatrix} -13.4344 & 1.3039 & -1.9875 & -0.5895 & -0.0263 & 0.1389 \end{bmatrix}, \\
m_1 &= 1.3265, \; m_2 = 2.7426, \; m_3 = 5.3007, \\
m_4 &= 4.1797, \; m_5 = 3.4327, \; m_6 = 1.7223, \\
m_7 &= 0.7966,
\end{aligned}
$$

and the PWA controller gains found for negative $w_m$ are

$$
\begin{aligned}
K_8 &= \begin{bmatrix} -13.4344 & -2.1267 & -1.9875 & -0.5895 & -0.0263 & 0.1389 \end{bmatrix}, \\
K_9 &= \begin{bmatrix} -13.4344 & -1.2171 & -1.9875 & -0.5895 & -0.0263 & 0.1389 \end{bmatrix}, \\
K_{10} &= \begin{bmatrix} -13.4344 & 1.9642 & -1.9875 & -0.5895 & -0.0263 & 0.1389 \end{bmatrix}, \\
K_{11} &= \begin{bmatrix} -13.4344 & -8.5965 & -1.9875 & -0.5895 & -0.0263 & 0.1389 \end{bmatrix}, \\
K_{12} &= \begin{bmatrix} -13.4344 & -1.5188 & -1.9875 & -0.5895 & -0.0263 & 0.1389 \end{bmatrix}, \\
K_{13} &= \begin{bmatrix} -13.4344 & 0.7830 & -1.9875 & -0.5895 & -0.0263 & 0.1389 \end{bmatrix}, \\
K_{14} &= \begin{bmatrix} -13.4344 & 1.2269 & -1.9875 & -0.5895 & -0.0263 & 0.1389 \end{bmatrix}, \\
m_8 &= 1.3583, \; m_9 = 2.7870, \; m_10 = 5.2856, \\
m_{11} &= 4.1797, \; m_{12} = 3.4385, \; m_{13} = 1.6307, \\
m_{14} &= 0.9333.
\end{aligned}
$$

## 4.2 Nonlinear Controller Design (Translational Velocity Dynamics)

In Section 3.2.2, four different cases for the UAV dynamics have been presented. In each of the four cases, the PWA state-feedback controller designed according to the method from the previous section is used to control the steering dynamics of the UAV path following problem. $V_T$ was assumed to be equal to $V_{T_{des}}$ in the PWA controller design. Cases three and four, however, have translational velocity dynamics represented by (3.22) for which a nonlinear controller is designed in this section to make $V_T$ converge to its desired value. This controller is designed using feedback linearization.

Recall from Chapter 1 that the essence of the feedback linearization control

method is to cancel the nonlinearities appearing in the system dynamics and then to design a linear controller to stabilize the linear dynamics. This can of course happen provided that the system is in feedback linearizable form or could be transformed to such a form [24, 40]. This condition is satisfied for the translational velocity dynamics given by (3.22). Thus, using feedback linearization, the control input $F_T$ in cases three and four is used to render the dynamics linear, and make $V_T$ track its desired value $V_{T_{des}}$.

Now, a nonlinear controller based on feedback linearization is designed for the translational velocity dynamics in each of cases three and four.

### Nonlinear controller: case three

Assume that (4.26) is successful in finding a globally quadratic Lyapunov function for the steering dynamics, which we will call $V_{steering}$. It will be shown in the proof of Theorem 3 presented in the next section that the following control law can be used to make $V_T$ converge to $V_{T_{des}}$ as well as to achieve global asymptotic stability of the cascaded system:

$$F_T = -\frac{mk_{vel}}{2}(V_T - V_{T_{des}}) + D + mg\sin\theta_v - m\frac{\partial V_{steering}}{\partial x}g(x), \tag{4.27}$$

where $k_{vel} > 0$ is a tunable gain, and $g(x)$ is a function of $x = [q \;\; \theta \;\; x_1 \;\; z_1]^T$, which is the state vector used for the steering dynamics in (3.21). Note that the term $\frac{\partial V_{steering}}{\partial x}g(x)$ in (4.27) exists because $V_{steering}$ is continuously differentiable with respect to the state $x$.

**Remark 4** The term $mg\sin\theta_v$ in (3.22) depends on $\theta_v$ which is constructed by integrating the pitch rate $q$ of the steering dynamics in (3.21). Since $\sin\theta_v$ is a bounded function between $-1$ and 1, canceling the term $mg\sin\theta_v$ using (4.27) should

not cause the control signal $F_T$ to become high. This, from a practical point of view, is necessary to avoid actuator saturation. □

**Nonlinear controller: case four**

A nonlinear control law that can make the equilibrium of (3.22) globally asymptotically stable ($GAS$) is

$$F_T = -\frac{mk_{vel}}{2}(V_T - V_{T_{des}}) + D + mg\sin\theta_v, \tag{4.28}$$

where $k_{vel} > 0$ is a design parameter. Taking $V_{velocity} = \frac{1}{2}(V_T - V_{T_{des}})^2$ as a control Lyapunov function ($clf$) for (3.22) yields

$$\dot{V}_{velocity} = -k_{vel}(V_T - V_{T_{des}})^2, \tag{4.29}$$

thus $\dot{V}_{velocity} < 0$ and $V_{velocity} > 0$; therefore, $V_T = V_{T_{des}}$ is $GAS$ according to standard Lyapunov theory [24, 40].

The same comment as in Remark 4 about canceling the term $mg\sin\theta_v$ using the control force in (4.28) also applies here.

## 4.3 Stability of the Overall Cascaded System

**Case three:**

It will be shown in the proof of the following theorem that the control law given by (4.27) can be used to make $V_T$ converge to $V_{T_{des}}$ and the closed-loop system is globally asymptotically stable.

**Theorem 3** *Consider the system (3.20). If there exists a feedback control input $M_e$ as a solution of (4.26), then this control input and the control force $F_T$ given by (4.27) render the equilibrium point of the closed-loop system GAS.*

**Proof:** First, note that by a similar reasoning to the proof of Theorem 1, sliding modes are ruled out in closed-loop. Let us write (3.21) in the following form

$$\dot{x} = f(x) + g(x)V_T + g_M(x)M_e. \tag{4.30}$$

Now we augment (4.30) with the translational velocity dynamics to obtain the following augmented system

$$\dot{x} = f(x) + g(x)V_T + g_M(x)M_e \tag{4.31a}$$

$$\dot{V}_T = \frac{1}{m}(F_T - D - mg\sin\theta_v). \tag{4.31b}$$

Introducing the error variable

$$\nu = V_T - V_{T_{des}}, \tag{4.32}$$

where $V_{T_{des}}$ is constant, and differentiating with respect to time yields

$$\dot{\nu} = \dot{V}_T - \dot{V}_{T_{des}} = \frac{1}{m}(\dot{F}_T - D - mg\sin\theta_v). \tag{4.33}$$

System (4.31) can now be rewritten in terms of the error state as

$$\begin{aligned} \dot{x} &= f(x) + g(x)\left[\nu + V_{T_{des}}\right] + g_M(x)M_e \\ \dot{\nu} &= \tfrac{1}{m}(F_T - D - mg\sin\theta_v). \end{aligned} \tag{4.34}$$

If there exists a feedback control moment $M_e$ as a solution of (4.26), this control moment guarantees that the function $\nabla_V$ defined as

$$\nabla_V = \frac{\partial V_{steering}}{\partial x}\left\{f(x) + g(x)V_{T_{des}} + g_M(x)M_e\right\} \tag{4.35}$$

verifies

$$\nabla_V < -\alpha V_{steering}. \tag{4.36}$$

A candidate control Lyapunov function for the overall system is thus proposed by augmenting $V_{steering}$ as follows

$$V = V_{steering} + \frac{1}{2}\nu^2. \tag{4.37}$$

The derivative of (4.37) along the state trajectories of the system (4.34) is

$$\begin{aligned}\dot{V} &= \tfrac{\partial V_{steering}}{\partial x}\dot{x} + \nu\dot{\nu} \\ &= \tfrac{\partial V_{steering}}{\partial x}\left\{f(x) + g(x)\left[\nu + V_{T_{des}}\right] + g_M(x)M_e\right\} + \nu\left\{\tfrac{1}{m}(F_T - D - mg\sin\theta_v)\right\} \\ &= \tfrac{\partial V_{steering}}{\partial x}\left\{f(x) + g(x)V_{T_{des}} + g_M(x)M_e\right\} \\ &\quad + \nu\left\{\tfrac{\partial V_{steering}}{\partial x}g(x) + \tfrac{1}{m}(F_T - D - mg\sin\theta_v)\right\},\end{aligned} \tag{4.38}$$

where all the terms containing the error state $\nu$ have been grouped together. Using (4.32) and (4.35), this last equation becomes:

$$\dot{V} = \nabla_V + (V_T - V_{T_{des}})\left\{\frac{\partial V_{steering}}{\partial x}g(x) + \frac{1}{m}(F_T - D - mg\sin\theta_v)\right\} \tag{4.39}$$

Assuming a feedback control moment $M_e$ was found that satisfies (4.36), the control force $F_T$ in (4.27) then guarantees that

$$\dot{V} < 0. \tag{4.40}$$

Therefore, condition (4.40) yields global asymptotic stability of $x = x_{cl}$ and $\nu = 0$ given that $V$ is radially unbounded. □

**Case four:**

The control force $F_T$ found for case three was able to make the translational velocity $V_T$ converge to its desired value $V_{T_{des}}$, as well as to guarantee stability of the overall system (steering dynamics plus translational velocity dynamics). This was possible because we were able to put the steering dynamics in (3.21) in the form given by (4.30) to which a backstepping argument can be applied. However, it is not possible to do that for the steering dynamics of case four, which are described by (3.20), because of the square of the velocity term that exists in (3.20a) and (3.20d).

Figure 4.1 shows the velocity $V_T$, which is the output of the translational velocity dynamics block, entering the steering dynamics block. Since the PWA controller

to the steering dynamics was found with the assumption that $V_T = V_{des}$, the question to be asked now is how much the steering dynamics will tolerate perturbations from $V_T$ as input without affecting the stability of the overall system. This is one of the interesting applications of the concept of input-to-state stability (ISS) [24]. However, to the best of the author's knowledge, no ISS results exist for PWA systems. Since this is not the scope of this thesis, the PWA controller and the nonlinear controller obtained for case four will only be evaluated through simulations without any proof of the global asymptotic stability of the cascaded system. Chapter 6, however, will review how the concept of ISS can be used to prove stability of interconnected subsystems.

In the next chapter, simulation results for each of the four cases will be given.

# Chapter 5

# Simulation Results

The system parameters used in simulation are:

Table 5.1: UAV parameters.

| | **value** | **unit** |
|---|---|---|
| mass: m | 18 | Kg |
| moment of inertia: $J_y$ | 3.5 | $Kgm^2$ |

Table 5.2: Path parameters.

| | **value** | **unit** |
|---|---|---|
| $c_c$ | 0.008 | $m^{-1}$ |
| $\dot{s}$ | 20 | m/s |

Table 5.3: Motor parameters [36].

| | **value** | **unit** |
|---|---|---|
| $K_m$ | $2.29 \times 10^{-2}$ | Nm/A |
| $K_e$ | $2.29 \times 10^{-2}$ | V/rad/s |
| $K_b$ | $3.5 \times 10^{-6}$ | Nms |
| $R_a$ | 0.71 | $\Omega$ |
| $J_m$ | $7.1 \times 10^{-6}$ | $kg \times m^2$ |
| $F_c$ | $5.6 \times 10^{-3}$ | Nm |

The Simulink block diagrams of the cascaded system for case three and four are shown in Figures 5.1 and 5.2, respectively.

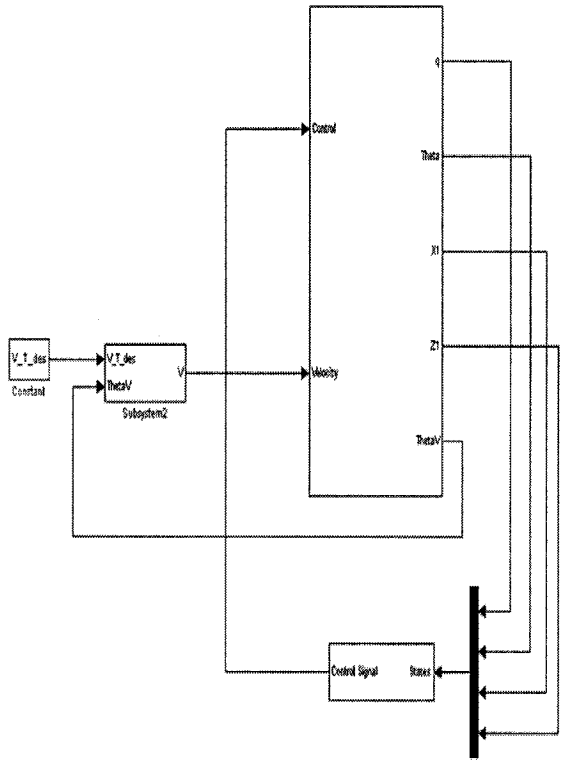


Figure 5.1: Simulink block diagram for case three.

The main objective of the controllers found in Chapter 4 is to drive the distance between the UAV and the desired path as well as the difference between the heading angle of the UAV's velocity vector and the orientation of the Serret-Frenet frame to zero. This is accomplished by driving the error states, $\theta, x_1$, and $z_1$ to zero. Note that in simulation, the PWA controller found after solving the controller algorithm in (4.26) is applied to the original nonlinear steering dynamics in each of the four cases, and not to the PWA approximate dynamics in (4.3). In simulation, the velocity gain $k_v$ is taken to be 5, and the desired velocity $V_{Tdes}$ is taken as $20m/s$.

The initial conditions used in simulation for the four cases are as follows:

**Case one:** $x_0 = [0.3 \ 0.4 \ 12 \ 8]^T$.

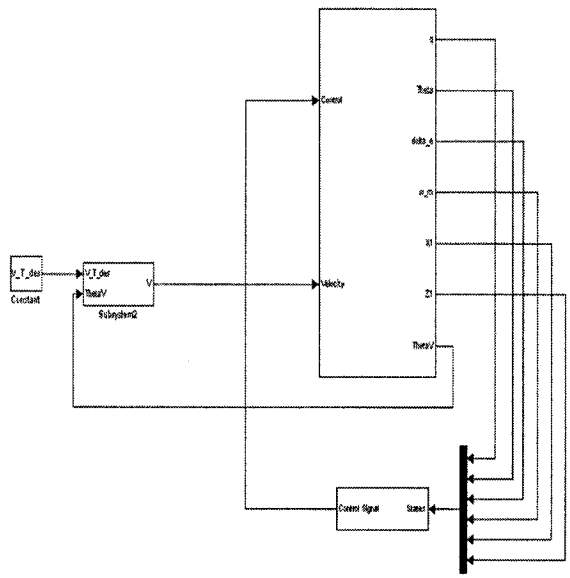

Figure 5.2: Simulink block diagram for case four.

**Case two:** $x_0 = [0.3\ \ 0.4\ \ 0.1\ \ 3\ \ 12\ \ 8]^T$.

**Case three:** $x_0 = [0.3\ \ 0.4\ \ 12\ \ 8]^T$, and $V_{T_0} = 10m/s$.

**Case four:** $x_0 = [0.3\ \ 0.4\ \ 0.1\ \ 3\ \ 12\ \ 8]^T$, and $V_{T_0} = 10m/s$.

Figure numbers of the simulation plots for all four cases are shown in the following table:

Table 5.4: Figure numbers of the simulation plots for all four cases.

| | $\theta, x_1$, and $z_1$ | $q$ | $V_T$ | UAV following a circular path |
|---|---|---|---|---|
| **Case 1** | Figure 5.3 | Figure 5.4 | - | Figure 5.5 |
| **Case 2** | Figure 5.6 | Figure 5.7 | - | Figure 5.8 |
| **Case 3** | Figure 5.9 | Figure 5.10 | Figure 5.11 | Figure 5.12 |
| **Case 4** | Figure 5.13 | Figure 5.14 | Figure 5.15 | Figure 5.16 |

Figures 5.3, 5.6, 5.9, and 5.13 show how the error states in all four cases converge to zero, thus confirming that the UAV is following the desired path. The pitch rate, on the other hand, converges to a steady state value equal to $c_c\dot{s}$. This is shown in Figures 5.4, 5.7, 5.10, and 5.14. Figures 5.11 and Figure 5.15 show the velocity $V_T$ converging to its desired value.

It is important to validate how strong Assumption 1 made at the beginning of Chapter 3 is. The component of the velocity on $z_b$ is defined as

$$w = V_T sin\alpha, \tag{5.1}$$

where $\alpha$ is the angle of attack. $w$ was simulated in the simulink model of case three, and the time history of $w$ is shown in Figure 5.17. The maximum value of $w$ is about $4.3m/s$. This value is 20% of the value of $V_T$. This shows that it is still important to consider the effect of $w$ for future work. Note that $w$ might be even larger for other kinds of paths.

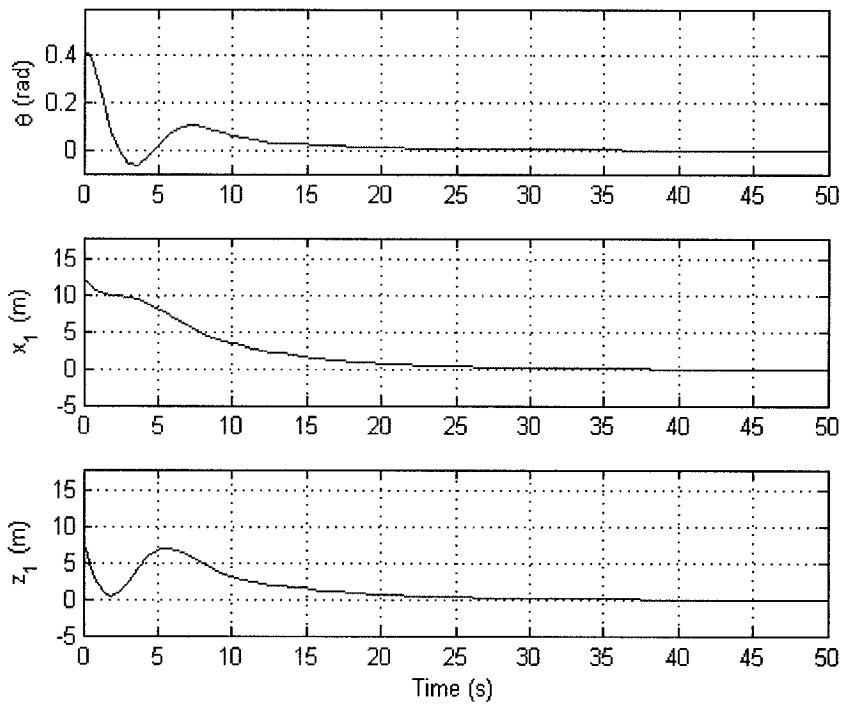


Figure 5.3: Case one: time histories of the error states.

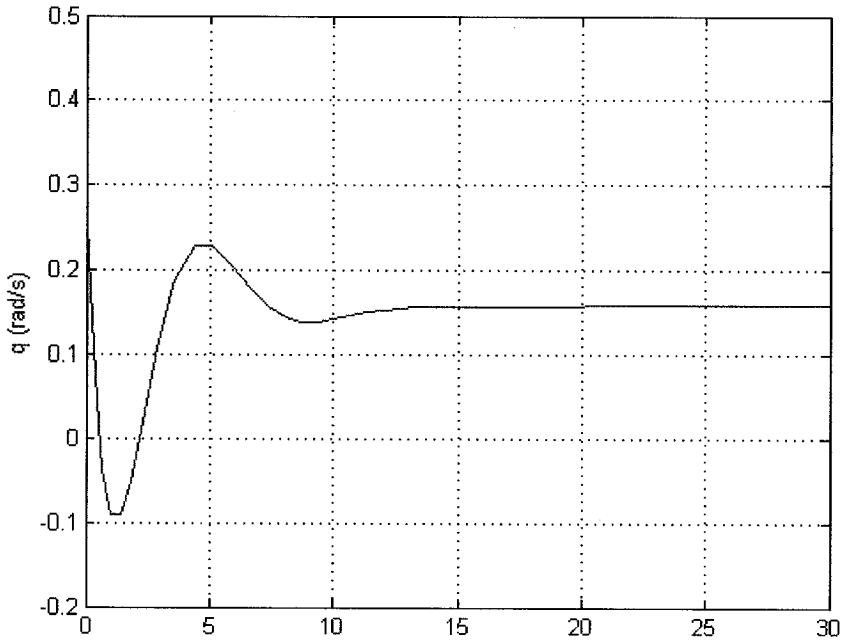


Figure 5.4: Case one: time history of the pitch rate.

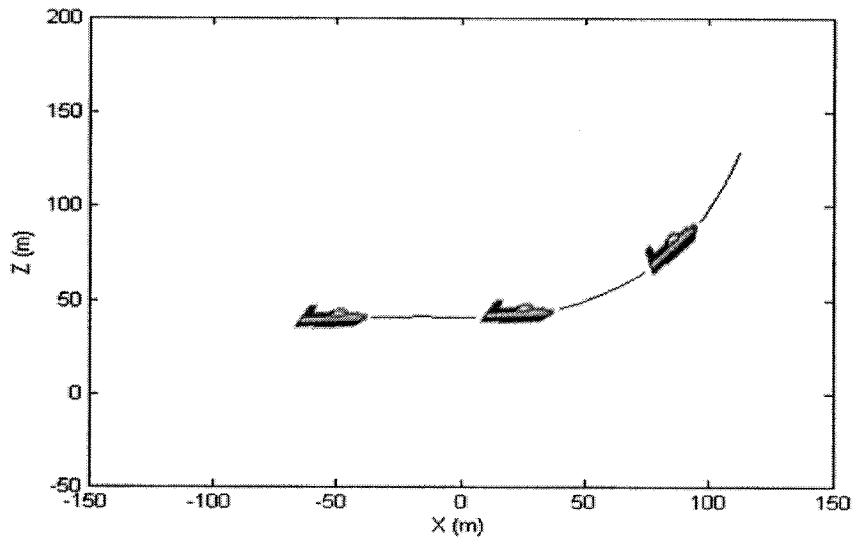


Figure 5.5: Case one: UAV following a circular path in the vertical plane.

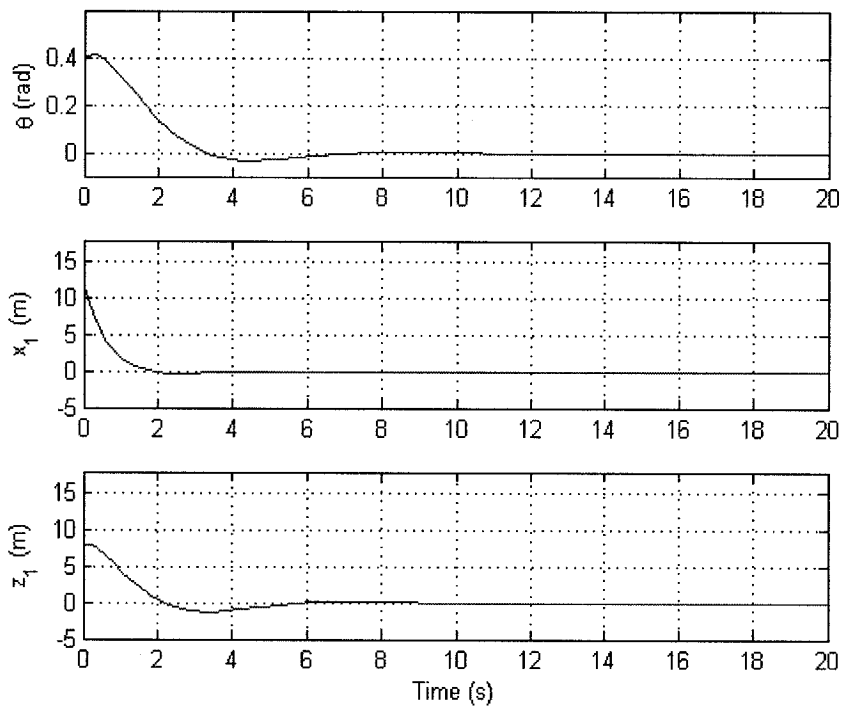


Figure 5.6: Case two: time histories of the error states.

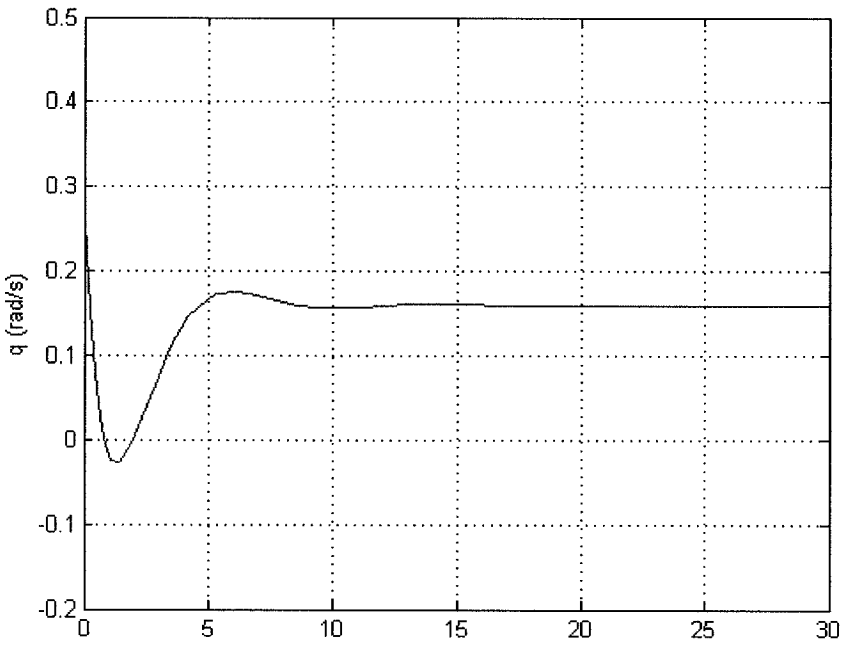


Figure 5.7: Case two: time history of the pitch rate.

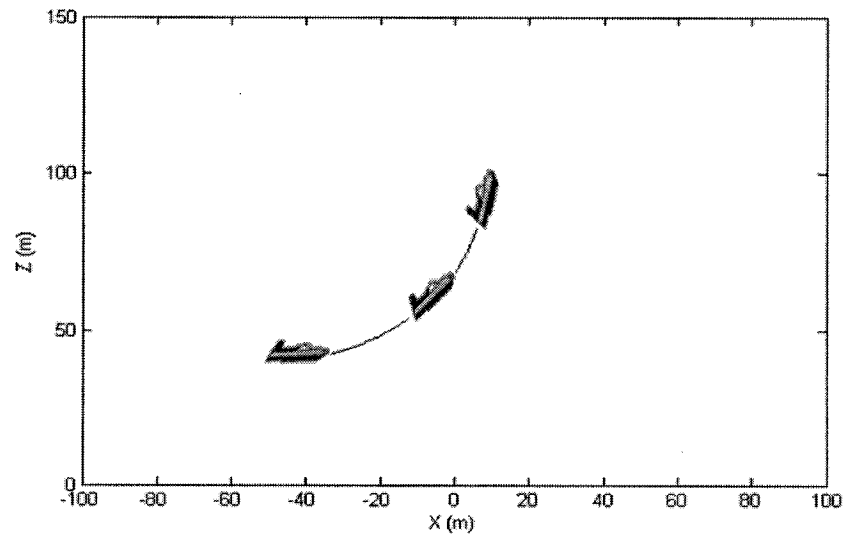


Figure 5.8: Case two: UAV following a circular path in the vertical plane.

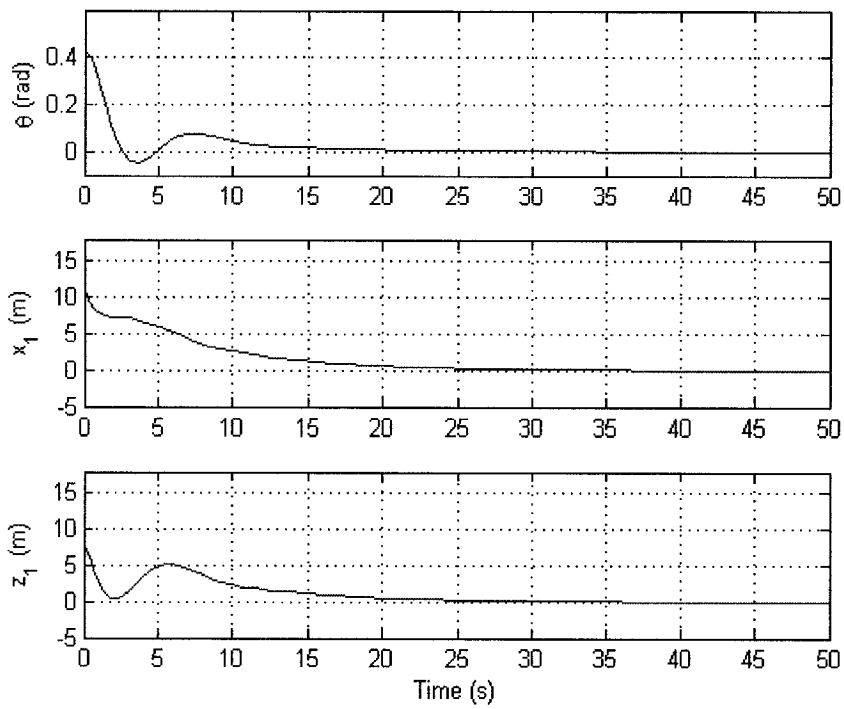


Figure 5.9: Case three: time histories of the error states.

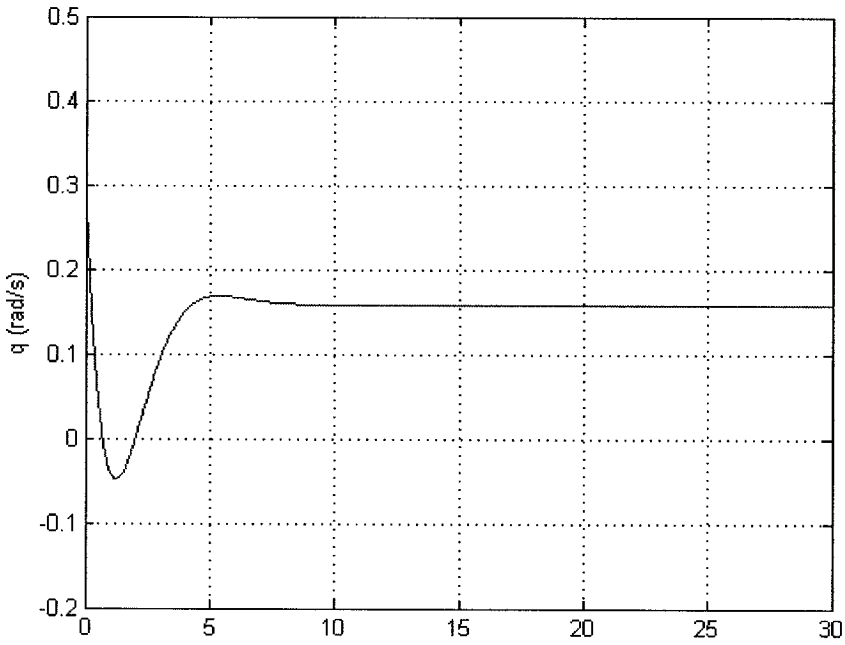


Figure 5.10: Case three: time history of the pitch rate.

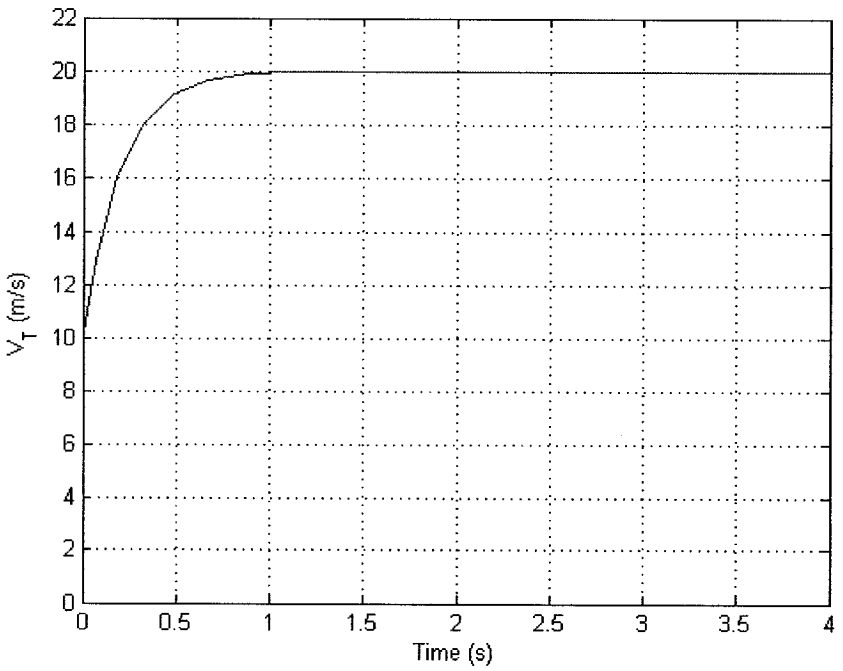


Figure 5.11: Case three: time history of the translational velocity.

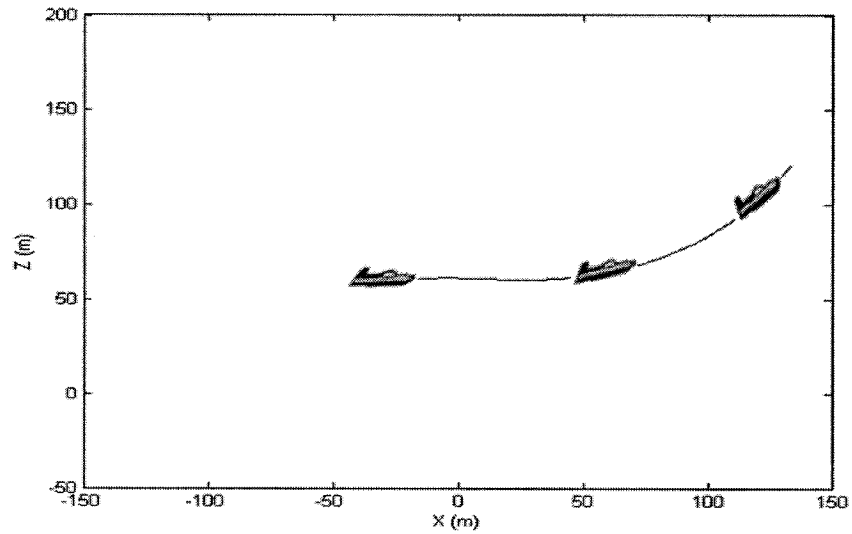


Figure 5.12: Case three: UAV following a circular path in the vertical plane.

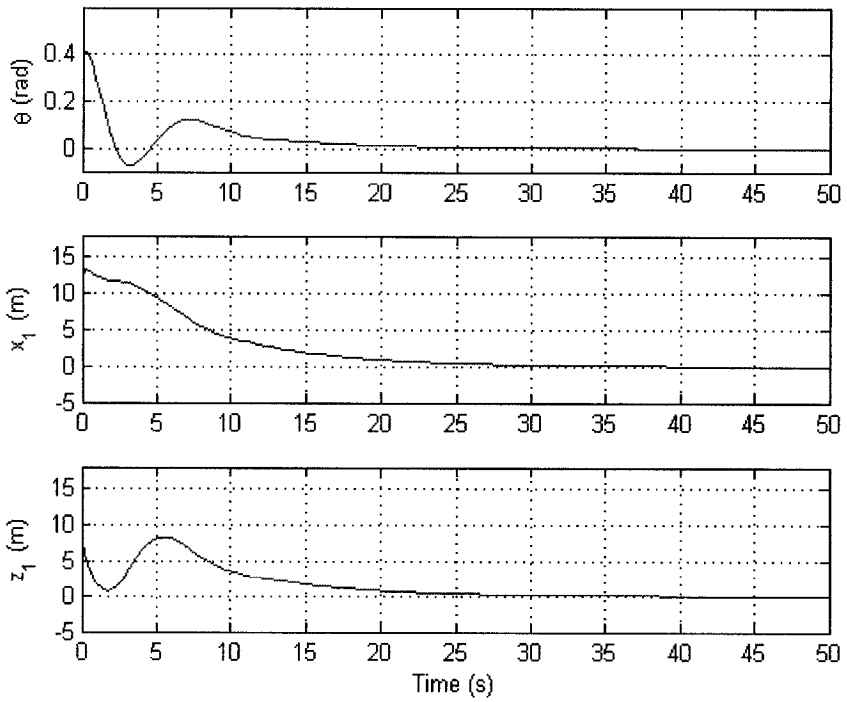


Figure 5.13: Case four: time histories of the error states.

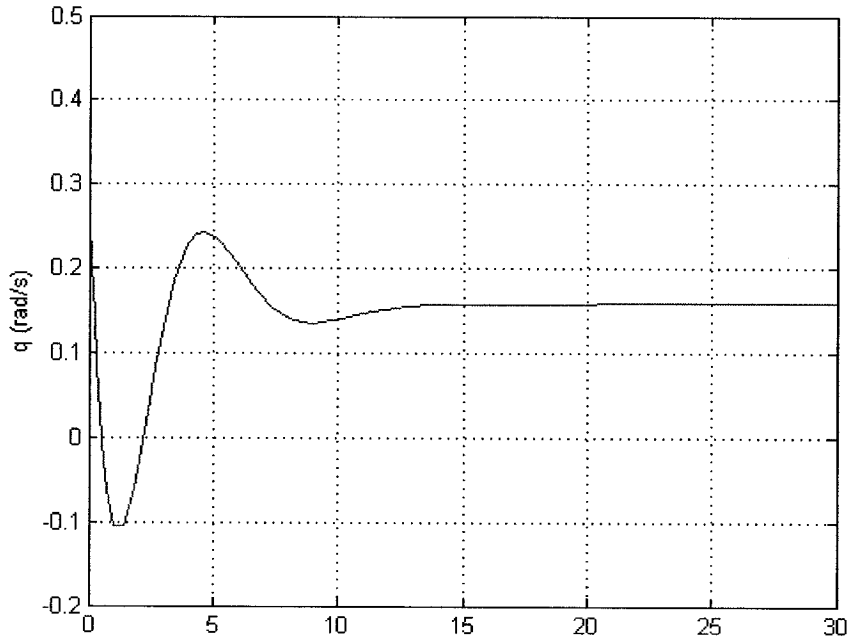


Figure 5.14: Case four: time history of the pitch rate.

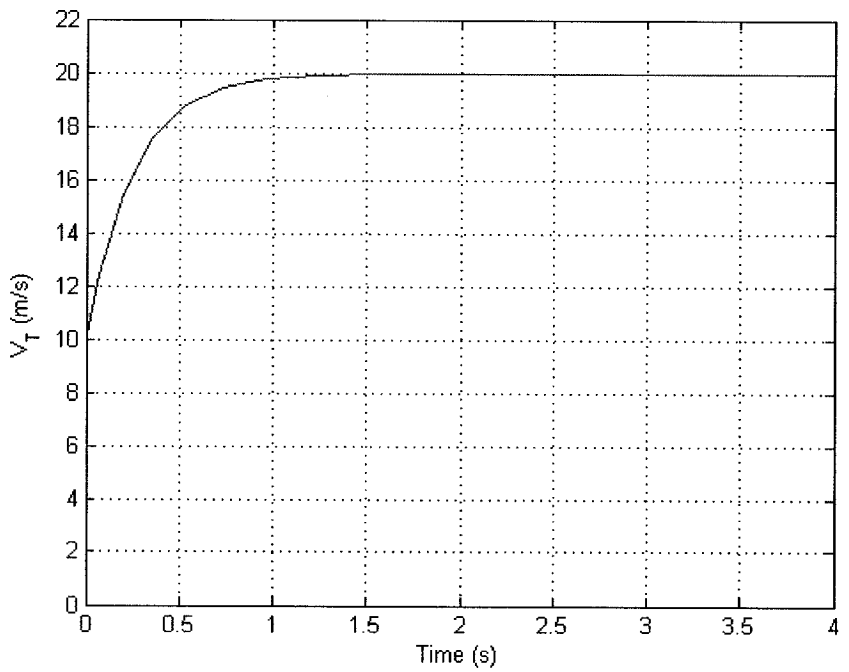


Figure 5.15: Case four: time history of the translational velocity.

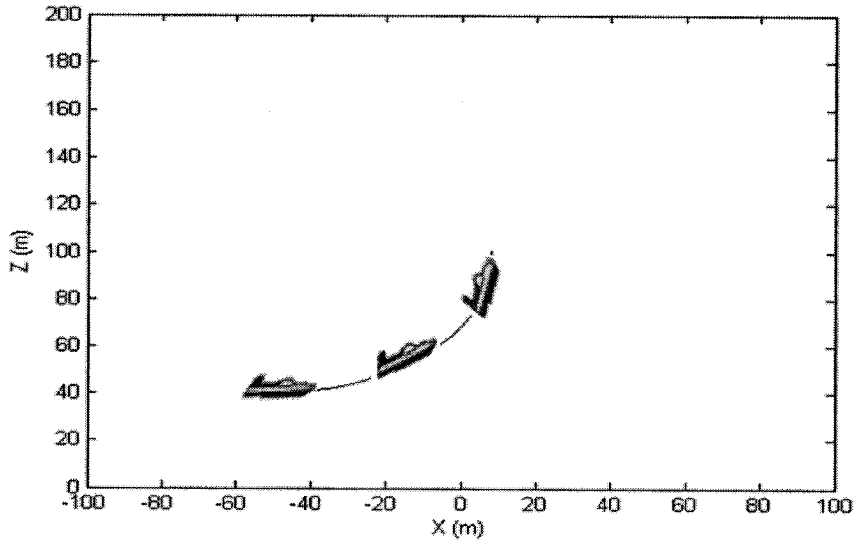


Figure 5.16: Case four: UAV following a circular path in the vertical plane.

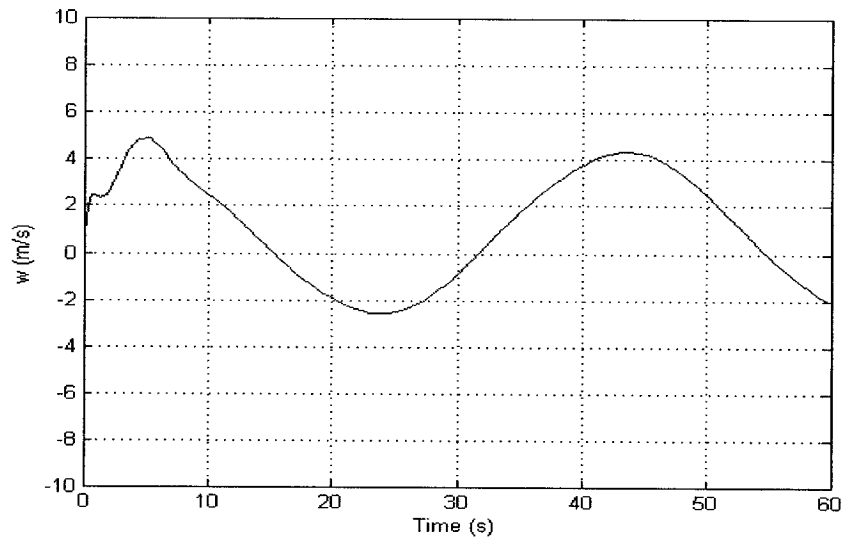


Figure 5.17: Time history of $w$ (using the simulink model of case three).

# Chapter 6

# Conclusions and Future Work

This thesis has presented a new control methodology for a UAV path following problem in the longitudinal plane. The dynamical model of the UAV in that plane incorporated the dynamics of the actuator that is used to deflect the elevator. To reduce the controller design complexity, the dynamics of the UAV were divided into two sets: one set describing the steering dynamics of the UAV, and another set describing the translational velocity dynamics of the UAV. Each set was treated separately in the control design. Moreover, the UAV path following problem was addressed in four different cases with increasing level of complexity and, correspondingly, increasing practical relevance. In the first case, the velocity $V_T$ is assumed to be constant and equal to $V_{T_{des}}$. Moreover, no actuator dynamics are considered, and the moment $M_e$ was the control input to the system. In case two, the model used in case one is augmented to account for the actuator dynamics, and the input voltage to the DC motor, $V_m$, was taken as the control input to the model. Moreover, $M_e$ was replaced by its expression that is a function of the elevator deflection and the velocity. In case three, the dynamical model used in the first case was cascaded with the dynamics of the translational velocity $V_T$, and the engine thrust force $F_T$

was used to control the velocity $V_T$ to a desired value. The dynamics in case four, on the other hand, were formed by cascading the dynamical model used in case two with the translational velocity dynamics, and similar to case three, the engine thrust force $F_T$ was used to control the velocity $V_T$ to a desired value. PWA functions were used to approximate nonlinearities that appear in the steering dynamics in each of the four cases.

A PWA state-feedback controller has been designed for the steering dynamics in all of the four cases. A nonlinear controller was designed based on feedback linearization for the translational velocity dynamics in cases three and four. The PWA controller found after solving the controller algorithm in (4.26) was simulated in feedback with the original nonlinear steering dynamics in each of the four cases, and not with the PWA approximate dynamics represented by (4.3). Simulation results showed the effectiveness of the proposed control methodology, where the desired path was followed in all four cases.

In conclusion, PWA functions were used to approximate the nonlinearities appearing in the steering dynamics and to model the Coulomb friction in the actuator dynamics of the elevator. The PWA controller synthesis is a systematic method for finding the controller parameters and a Lyapunov function that proves global asymptotic stability of the closed-loop system. However, the complexity of the PWA control system increases with the increase of the number of regions needed for approximating the nonlinear dynamics in the system. Minimizing such number of regions is still an open research topic.

The main contributions of this thesis are:

- The first successful application of the PWA control methodology to a flight control problem.

- Proposing a novel architecture for controller design consisting of a cascade interconnection of a PWA control subsystem (to control the steering dynamics of the UAV) with a nonlinear control subsystem (to control the translational velocity dynamics of the UAV).
- Modeling the Coulomb friction that exists in the actuator dynamics of an elevator in a piecewise affine framework.
- Dividing the path following problem dynamics into four models
    1. Proving stability of the closed-loop system for the first three cases
    2. Showing good control performance through simulation in all four cases

Stability of the overall cascaded system was proved for case three because it was possible to put the steering dynamics in (3.21) in the form given by (4.30). On the other hand, this was not possible to do for case four because of the square of the velocity term that exists in the dynamics for this case. Furthermore, it was mentioned how the concept of ISS can be used to prove stability of interconnected subsystems. However, to the best of the author's knowledge, no ISS results exist for PWA systems. This could be investigated in future work. As a motivation to this, the concept of ISS is briefly explained here. It is highlighted how one could use ISS to prove the overall stability of two cascaded smooth nonlinear subsystems. Assume the following two subsystems [24]

$$\dot{x}_1 = f_1(x_1, x_2) \tag{6.1a}$$

$$\dot{x}_2 = f_2(x_2) \tag{6.1b}$$

Figure 6.1 shows the interconnection between (6.1a) and (6.1b) where $x_2$ from (6.1b) works as an input to (6.1a).

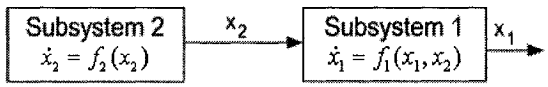


Figure 6.1: Cascade interconnection between two subsytems.

The question to be asked now is how much (6.1a) will tolerate perturbations from $x_2$ as input without affecting the stability of the overall system. Assume that $f_1$ and $f_2$ are locally Lipschitz in $x = [x_1 \ x_2]^T$, and that

$$\dot{x}_1 = f_1(x_1, 0) \tag{6.2a}$$

$$\dot{x}_2 = f_2(x_2),$$

have GAS equilibrium points at their respective origins. Then, the following theorem is stated [24], which can be used to prove stability of two interconnected subsystems.

**Theorem 4** *Under the stated assumptions, if subsystem (6.1a), with $x_2$ viewed as input, is input-to-state stable and the origin of (6.1b) is globally asymptotically stable, then the equilibrium of the interconnected system (6.1) is globally asymptotically stable.* □

**Proof:** See [24].

Unfortunately, PWA systems are not Lipschitz and therefore this result cannot be used for such systems. In addition to investigating the ISS of PWA systems, possible extensions of the work done in this thesis include studying the effectiveness of the controller in the presence of actuator saturation and plant parameter uncertainty. More general trajectories for UAV missions with the dynamics of the vertical component of the velocity could also be addressed in future work. Furthermore, feedback linearization was used in this work to find a nonlinear controller for the translational velocity dynamics. However, in Chapter 1 the disadvantages

of using feedback linearization were mentioned. Therefore, future work could also concentrate on finding other control techniques to avoid the disadvantages that exist in feedback linearization. Finally, experimental results to validate the models are also important to be considered in future work.

# Bibliography


[1] A. Abdelghani Zergaoui and A. Bennia, "Identification and control of an asynchronous machine using neural networks," *Proceedings of the 6th IEEE International Conference on Electronics, Circuits and Systems*, vol. 2, pp. 1043–1046 , Sept. 1999.

[2] B. Armstrong-Hélouvry, *Control of Machines with Friction*, Kluwer Academic Publishers, 1991.

[3] B. Armstrong-Hélouvry, P. Dupont, and C. Canudas de Wit, "A survey of models, analysis tools and compensation methods for the control of machines with friction," *Automatica*, vol. 30, no. 7, pp. 1083–1138, 1994.

[4] J. Blakelock, *Automatic Control of Aircraft and Missiles*, 2nd edition, John Wiley and Sons, 1991.

[5] S.P. Boyd, L.E. Ghaoui, E. Feron, and V. Balakrishnan, *Linear Matrix Inequalities in System and Control Theory (Studies in Applied Mathematics.)* Philadelphia: SIAM, 1994.

[6] A. Dogan and S. Venkataramanan, "Nonlinear Control for Reconfiguration of UAV Formation," *AIAA Journal of Guidance, Control and Dynamics*, vol. 28, no. 4, JulyAugust 2005.

[7] R. Dorf, and R. Bishop, *Modern Control Systems*, 9th edition, Prentice Hall, 2001.

[8] P. Encarnacao and A. Pascoal, "3D path following for autonomous underwater vehicle," *Proceedings of the 39th IEEE Conference on Decision and Control*, Sydney, Australia, December 2000.

[9] B. Etkin, and L. D. Reid, *Dynamics of Flight: Stability and Control*, 3rd edition, John Wiley and Sons, 1996.

[10] D. Fontaine, P. V. Kokotović, "Approaches to global stabilization of a nonlinear system not affine in control," *Proceedings of the American Control Conference*, Philadelphia, Pennsylvania June 1998.

[11] G. Franklin, J. Powell, and A. Emami-Naeini, *Feedback Control of Dynamic Systems*, 3rd edition, Addison Wesley, 1994.

[12] M. Golden and B. Ydstie, "Bifurcation analysis of drift instabilities in adaptive control," *Proceedings of the 30th IEEE Conference on Decision and Control*, vol. 2, pp. 1108–1109, Dec. 1991.

[13] O. Härkegård, S. Torkel Glad, "A backstepping design for flight path angle control," *Proceedings of the 39th IEEE Conference on Decision and Control*, vol. 4, pp. 3570–3575, December 2000.

[14] O. Härkegård, "Backstepping designs for aircraft control - What is there to gain?" *Technical report no. LiTH-ISY-R-2339*, Linköpings Universitet, Department of Electrical Engineering, Sweden, March 2001.

[15] O. Härkegård, "Flight Control Design Using Backstepping," *Masters thesis No. 875*, Linköpings Universitet, Department of Electrical Engineering, Sweden, 2001.

[16] A. Hassibi and S.P. Boyd, "Quadratic stabilization and control of piecewise-linear systems," *Proceedings of the American Control Conference*, pp. 3659–3664, Philadelphia, PA, June 1998.

[17] B. Holder, *Unmanned Air Vehicles: An Illustrated Study of UAVs*, Schiffer Publishing Inc, 2001.

[18] P. Ioannou and J. Sun, *Robust Adaptive Control*, Prentice Hall, 1996.

[19] M. Johansson and A. Rantzer, "Computation of piecewise quadratic lyapunov functions for hybrid systems," *IEEE Transactions on Automatic Control*, vol. 43, pp 555–559, April 1998.

[20] M. Johansson, *Piecewise Linear Control Systems*, Springer, 2003.

[21] E.N. Johnson and A.J. Calise, "Neural network adaptive control of systems with input saturation," *Proceedings of American Control Conference*, vol.5, pp. 3527–3532, June 2001.

[22] I. Kaminer, A. M. Pascoal, P. P. Khargonekar, and E. E. Coleman, "A velocity algorithm for the implementation of gain-scheduled controllers," *Automatica*, vol. 31, no. 8, pp. 1185–1191, 1995.

[23] Y. Kanayama, Y. Kimura, F. Miyazaki, and T. Noguchi, "A stable tracking control method for an autonomous mobile robot," *Proceedings of IEEE International Conference on Robotics and Automation*, vol.1, pp. 384–389, May 1990.

[24] H. Khalil, *Nonlinear Systems*, Prentice Hall, Upper Saddle River, NJ, 2nd edition, 1996.

[25] M. Krstić, I. Kanellakopoulos, and P. V. Kokotović, *Nonlinear and Adaptive Control Design*, John Wiley and Sons, 1995.

[26] M. S. M. Kocvara, F. Leibfritz and D. Henrion, "A nonlinear sdp algorithm for static output feedback problems in compleib," University of Trier, Germany, Tech. Rep., 2004.

[27] D. Liberzon, *Switching in Systems and Control*, Birkhauser, 2003.

[28] J. Lofberg, "Yalmip: a matlab toolbox for rapid prototyping of optimization problems," Automatic Control Laboratory, ETH Zürich, 2004.

[29] J. Magni, S. Bennani, and J. Terlouw, *Robust Flight Control: A Design Challenge*, Springer, 1997.

[30] H. McDaid and D. Oliver, *Smart Weapons: The Top Secret History of Remote Controlled Airborne Weapons*, Barnes and Noble, Inc., 1998.

[31] A. Micaelli, and C. Samson, "Trajectory tracking for unicycle-type and two-steering-wheels mobile robots," *Technical Report No. 2097*, **INRIA**, Sophia Antipolis, Nov. 1993

[32] K. Munson, *World Unmanned Aircraft*, Jane"'s Publishing Company Limited, London, 1988.

[33] K. Munson, *Janes Unmanned Aerial Vehicles and Targets*, Issue Thirteen, Jane"'s Information Group Limited, UK, 1999.

[34] R. C. Nelson, *Flight Stability and Automatic Control*, 2nd edition, McGraw-Hill, 1998.

[35] S. Pettersson, "Analysis and design of hybrid systems," *Ph.D. dissertation*, Chalmers University of Technology, Gteborg, June 1999.

[36] http://www.pennmotion.com, Gearmotor GM9236S013.

[37] W. Ren and R.W. Beard, "Trajectory tracking for unmanned air vehicles with velocity and heading rate constraints," *IEEE Transactions on Control Systems Technology*, vol. 12, no. 5, pp. 706–716, Sep. 2004.

[38] W. Rugh, and J. S. Shamma, "Research on gain scheduling," *Automatica*, vol. 36, no. 9, pp. 1401–1425, 2000.

[39] B. Samadi, and L. Rodrigues, "Piecewise-affine controller synthesis based on a local linear controller: toolbox for MATLAB using the PENBMI solver," Technical Report, Concordia University, 2005.

[40] J. Slotine and W. Li, *Applied Nonlinear Contol*, Prentice Hall, 1991.

[41] S. Shehab, and L. Rodrigues, "Preliminary results on UAV path following using piecewise-affine control," *Proceedings of the 2005 IEEE Conference on Control Applications*, pp. 358–363, August 28-31, 2005.

[42] S. Shehab, and L. Rodrigues, "UAV Path Following Using a Mixed Piecewise-Affine and Backstepping Control Approach," *Proceedings of the Eighth IASTED International Conference on Control and Applications*, May 24-26, 2006.

[43] D. Soeanto, L. Lapierre, and A. Pascoal, "Adaptive, non-singular path-following control of dynamic wheeled robots," *Proceedings of the 42nd IEEE Conference on Decision and Control*, vol. 2, pp. 1765–1770, December 2003.

[44] J. F. Sturm, "Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones (Updated for Version 1.1R2)," *Optimization Methods and Software* 11–12 (1999), pp. 625–653.

[45] L. Rodrigues, and J. How, "Automated control design for a piecewise-affine approximation of a class of nonlinear systems," *Proceedings of American Control Conference*, vol.4 , pp. 3189–3194, June 2001.

[46] L. Rodrigues, and J. How, "Observer-based control of piecewise-affine systems," *International Journal of Control,* vol. 76, pp. 459–477, 2003.

[47] L. Rodrigues, and J. How, "Synthesis of piecewise-affine controllers for stabilization of nonlinear systems," *IEEE Conference on Decision and Control*, vol. 3, pp. 2071–2076, December 2003

[48] G. Tao, "Adaptive control of systems with nonsmooth input and output nonlinearities," *IEEE Transactions on Automatic Control*, vol. 41, no. 9, pp. 1348–1352, September 1996.

[49] A. Taware and G. Tao, *Control of Sandwich Nonlinear Systems*, Springer, 2003.

[50] G. Tao, P. V. Kokotović, "Adaptive Control of Plants with Unknown Dead-Zones," *IEEE Transactions on Automatic Control*, vol. 39, no. 1, pp. 59–68, January 1994.

[51] M. A. Unar and D. J. Murray-Smith, "Automatic steering of ships using neural networks," *International Journal of Adaptive Control and Signal Processing*, vol. 13, no. 4, pp. 203–218, July 1999.

[52] W. Wagner, *Lightning Bugs and Other Reconnaissance Drones: The Can-Do Story of Ryan's Unmmanned 'Spy Plane'*, Armed Forces Journal International, Fallbrook, California, 1982.

[53] W. Wagner and W. Sloan, *Fireflies and Other UAVs (Unmanned Aerial Vehicles)*, Leicester, UK, Midland Publishing, 1992.

[54] K. Werrell, *The Evolution of the Cruise Missile*, Air University Press, Maxwell AFB, Alabama, September 1985.