A New Methodology for Obtaining Piecewise Affine Models Using a Set of Linearisation Points and Voronoi Partitions

Scott Casselman

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 (Mechanical Engineering) at
Concordia University
Montréal, Québec, Canada

May 2009

Library and Archives
Canada

Published Heritage
Branch

395 Wellington Street
Ottawa ON K1A 0N4
Canada

Bibliothèque et
Archives Canada

Direction du
Patrimoine de l'édition

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence
ISBN: 978-0-494-63110-2
Our file Notre référence
ISBN: 978-0-494-63110-2

ABSTRACT

A New Methodology for Obtaining Piecewise Affine Models Using a Set of Linearisation Points and Voronoi Partitions

Scott Casselman

To understand complex dynamical systems, approximations are often made by a linearisation about an operating point of interest. The drawback of this linear approximation is that it only describes the system locally, around the operating point. One possible solution to overcome this drawback is to approximate the complex nonlinear dynamical system with a piecewise affine (PWA) system. Approximating nonlinear dynamical systems is very important in system theory where one is interested in simplifying its analysis and numerical simulation. PWA modelling is a very powerful tool to represent nonlinear systems as a collection of a finite number of linear systems. In the literature of PWA, a uniform grid (UG) approximation is the current method being used to approximate a nonlinear system as a PWA system. The drawback of this method is the potential large amount of regions required to obtain a desired accuracy, which is most evident for systems with more than one variable in the domain of the nonlinearity. In order to reduce the number of regions, the proposed research will develop a new methodology for obtaining PWA models using a set of linearisation points (SLP) and the Voronoi partition. First, in order to generate a partition based on a SLP, the curvature of the nonlinearity is used as a tool for selecting appropriate locations for the linearisation points. Next, an algorithm is proposed to automate the SLP approximation for both curves and surfaces. The SLP and UG approximation methods are then compared over several simple examples. Finally, the newly proposed approximation methodology is applied to three case studies: modelling of a

nonlinear mechanical system and modelling and control of an unmanned aerial vehicle (UAV) and a micro air vehicle (MAV).

To Papa,

You would be so proud of who I have become.

ACKNOWLEDGEMENTS

First and foremost, I would like to thank my thesis advisor, Dr. Luis Rodrigues. I could not have accomplished this important goal without your guidance and immense support. Also, I would like to thank the professors and the administrative and technical support staff in the department who have also played a vital role in my success.

I had the honour of sharing the same space and time both in and outside the lab with some fantastic colleagues. I would like to thank them all for everything they have done.

- Behzad Samadi
- Chris Johnston
- Samer Shehab
- Henrique Paiva
- Behnam Gholitabar
- Kyungjae Baik
- Xiaoxi Huang
- Narendra Gollu
- Navid Dadkhah
- Ralf Endress
- Nastaran Nayebpanah
- Ali Reza Mehrabian
- Ralph Koyess
- Giancarlo Luglio

Moreover, I would like to thank all my friends for distracting me from this thesis when I needed a break and also for constantly reminding me that my perseverance would finally get me to the finish line. Last but not least, I would like to thank my family for their affection and support throughout my graduate degree.

"Many of the great achievements of the world were accomplished by tired and discouraged men who kept on working."

- Unknown Source

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS

State Variables

Symbol	Unit	Definition
u	m/s	Translational Velocity along x_b
v	m/s	Translational Velocity along y_b
ψ	rad	Yaw Angle
$\beta = \arctan \frac{v}{u}$	rad	Sideslip Angle
r	rad/s	Yaw Rate
y	m	Inertial y position

Control Inputs

Symbol	Unit	Definition
δ_{ped}	N	Tail Rotor Pedal Input
δ_m	$N - m$	Differential Main Rotor Control Input

Aircraft Data

Symbol	Unit	Definition
m	kg	Aircraft Mass
I_{zz}	$kg - m^2$	Mass Moment of Inertia
l_t	m	Tail Rotor Hub Location Behind C.G.
k_t	$-$	Tail Rotor Blade Constant
k_v	$-$	Slide Damping Coefficient
k_r	$-$	Yaw Damping Coefficient

Forces, Moments and Accelerations

Symbol	Unit	Definition
X	N	Total Axial Force
Y	N	Total Side Force
N	$N-m$	Total Yawing Moment
$g = 9.81$	m/s^2	Acceleration due to Gravity

Acronyms

Label	Definition
PWL	Piecewise Linear
PWA	Piecewise Affine
UG	Uniform Grid
SLP	Set of Linearisation Points
MAV	Micro Air Vehicle
UAV	Unmanned Air Vehicle
DOF	Degrees Of Freedom
BMI	Bilinear Matrix Inequality

Chapter 1

Introduction

1.1 Motivation

To understand complex dynamical systems, approximations are often made by a linearisation about an operating point of interest. The drawback of this linear approximation is that it only describes the system locally, around the operating point. One possible solution to overcome this drawback is to approximate the complex nonlinear dynamical system with a PWA system. This approximation is very important in system theory where one is interested in simplifying the analysis and numerical simulation of a complex dynamical system [10]. PWA modelling is a very powerful tool to represent nonlinear systems as a collection of a finite number of linear systems. A PWA system is the result of approximating the nonlinearities of the original system into a finite number of designated regions. The accuracy of this approximation can be increased by increasing the number of regions. The main advantage of approximating a nonlinear dynamical system as a PWA system is the simplicity of its theoretical formulation, which makes it easy to apply to complex systems [11]. Moreover, in many engineering cases, nonlinearities such as Coulomb friction and actuator saturation can be modelled exactly by PWA characteristics. PWA systems are a subclass of hybrid systems where the discrete dynamics are associated with the designated regions and each region has a particular set of continuous dynamics. Existing literature on approximating complex nonlinear dynamics systems as PWA systems generally has the potential for a large number of regions. This fact motivates the use of a new approach to approximate the nonlinear dynamics in order to reduce the number of regions in the approximation.

Based on the above motivation, the goal of this thesis is to develop a new methodology for obtaining PWA models. This thesis will investigate the use of a SLP and the Voronoi partition to approximate nonlinear systems with smooth vector fields. This methodology will be applied to several case studies including a rotorcraft MAV. In the following two sections, the previous work on modelling complex dynamical systems is reviewed. The first section describes previous work done on PWA modelling in general, while the second section will focus on MAV modelling in particular.

1.2 Literature Survey

1.2.1 Piecewise Affine Modelling

This section will be broken into two parts. The first part will review the literature of PWA systems and the second part will review the literature of Voronoi partitions.

Piecewise Affine Systems

PWA systems have received a great amount of attention since the pioneering work of Andronov [12] on oscillations in nonlinear systems in the late 1940's. In the early 1980's, Sontag [13] presented the analysis of discrete-time piecewise linear (PWL) systems using polyhedral partitions. Later, Julian et al. [14] proposed an algorithm to approximate a nonlinear dynamical system as a PWL system over a simplicial partition of a UG. Rantzer and Johansson [15, 16] developed a systematic method for continuous-time PWL systems to improve nonlinear approximations around an equilibrium point by refining the number of polytopic cells of a simplicial partition. Inspired by the work in [15], Morari et al. [17] extended the existing continuous-time approach to discrete-time PWL systems. Other works using a simplicial partition of a uniform rectangular grid to approximate nonlinear systems as PWL and PWA systems include [10, 18, 19, 20] and [5, 6, 21, 22, 23, 24, 25, 26], respectively. Simplicial partitions over a non-UG have also been used. For instance, Khargonekar et al. [27] obtain a non-uniform PWL approximation using a least squares algorithm that minimises a functional. Recently, Samadi and Rodrigues [28] developed a MATLAB and SIMULINK[1] toolbox capable of approximating a continuous time nonlinear system and synthesising PWA controllers using a uniform rectangular grid. This toolbox can perform anchored approximations, which ensures that the linearisation of the PWA approximation is equivalent to linearisation of the original nonlinear function at the equilibrium point of the system. As a result of these developments, Yue et al. [23] derived the first PWA model of a three DOF aerobatic helicopter. The largest drawback to the PWA helicopter model in [23] is the use of 30 regions leading to 30 different controllers. Although [28] considers a single linearisation point (at the equilibrium point) in the PWA anchored approximation, a SLP is not considered.

Voronoi Partitions

There exists two main dual techniques to partition the domain of a finite point set into polytopic regions: Delaunay triangulation and Voronoi partitions. A Voronoi partition can always be used to partition the domain of a point set consisting of n points, even when the points are collinear. Therefore, only Voronoi partitions are considered in this thesis. Voronoi partitions have received a great amount of attention for modelling purposes. Voronoi partitions have modelling applications in many

[1] MATLAB and SIMULINK are trademarks of The Mathworks, Inc

areas including engineering [8, 29], natural and health sciences [30, 31] and physical sciences [32, 33] to name a few. These partitions are even observed in nature [34]. Despite the wealth of applications that apply the Voronoi partition, to the best of the authors knowledge, there has not been any formal work dedicated to partitioning the domain of a nonlinear system with a Voronoi partition to obtain a PWA system.

1.2.2 Micro Air Vehicle Modelling

This section will be broken into two parts. The first part will give a brief background of MAVs and the second part will review the literature of MAVs.

Background

A MAV is a small, lightweight aircraft characterised by its small aspect ratio and low Reynolds number (Re). Typically, the size of a MAV (wingspan, chord and height) is smaller than 15 centimeters as defined by DARPA[2] with a cruise speed of 40 kph or less [35][36]. This kind of aircraft has been receiving an increasing amount of research within the past few years because of its advantages of being a low-cost multi-functional aircraft [37]. The military envision the use of a MAV as a tactical tool that a single soldier will be able to use. With this light, low radar profile, and easily deployable MAV, a soldier is able to obtain vital information either through reconnaissance or sensing-type missions by launching it from a compact tube [1][38][39]. In terms of civilian use, MAVs can be useful for weather and atmosphere monitoring, search and rescue, surveillance, controlling harmful insect populations, air sampling, surveying wildlife, providing recreation and real-estate photography as well as many other applications [40][41]. Figure 1.1 taken from [1] presents a comparison between common aircraft and wildlife as a function of the operating weight and Reynolds number. There are currently three types of MAV designs (as seen in Figure 1.2) :

[2] The Defense Advanced Research Projects Agency

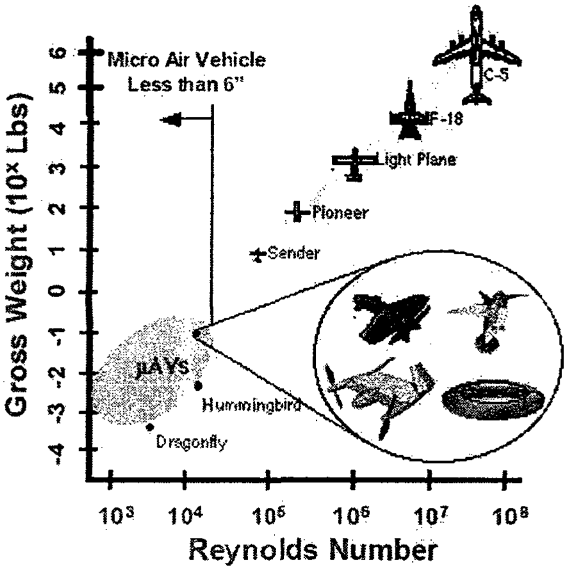


Figure 1.1: Comparison between a MAV and various aircraft and wildlife [1].

flapping wing, rotorcraft and fixed wing. The flapping wing MAV, also known as an ornithopter, is the most difficult to design due to need for a complex mechanical structure and an efficient power supply for the flight endurance. Its ability to hover and fly in close quarters to obstacles makes this design an invaluable choice to perform certain missions [42]. The rotorcraft MAV has the innate ability to hover, which makes it more maneuverable around obstacles at the expense of requiring additional power to keep the main lifting surfaces operational. Fixed wing MAVs hold the potential to generate the fastest vehicles and currently have a very promising future as there is some research focusing into combining the rotorcraft's and ornithopter's ability to hover with the fixed wing MAVs higher speed and smaller energy consumption [43].

Literature Review

The majority of the models for MAVs in the literature are linear. This comes as no surprise as linear aircraft models have proved to be quite accurate over the years and much easier to be used in control design than a nonlinear model. Nonlinear models are, thus far, much less popular in terms of implementation. Despite the

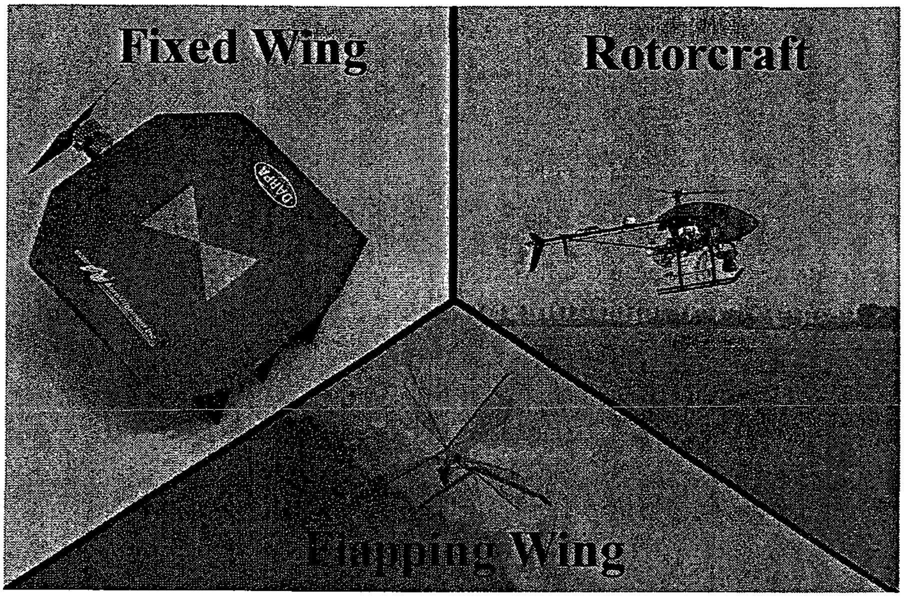


Figure 1.2: Types of MAVs (adapted from [2, 3, 4]).

type of model, there are four main approaches to obtain the unknown aerodynamic parameters that characterise the system: wind tunnel testing, analytical relations, computational fluid dynamics (CFD) and system identification.

- Wind tunnel testing is a powerful visual tool because the physical MAV wing assembly can be tested and its aerodynamic properties can be measured and observed directly [44][40][45][46][47][48][47][49][50][51][52][53][54].
- The analytical procedure follows basic assumptions (such as level flight and thin airfoil theory) to simplify the process of determining unknown aerodynamic parameters. Alongside wind tunnel test data, analytical relations are obtained based on the trends observed [55][40][48][49][56][52][57][58][53][43][59][60][61].
- A CFD approach uses a multitude of 2D [62][63][64] and 3D [50][55][65][66][67] [68][69][70][71][72] methods to numerically predict the aerodynamic parameters.
- System or model identification requires that input-output data be obtained directly from flight testing before a model can be created. Part of the recorded

data is used to obtain the model while the other part is used for validation [73] [74][75][76][77][78].

Table 1.1: Summary of MAV Data

University	Sys/ID	Wind/Tunn	CFD	Analyt.	Linear	Nonlinear
Florida	X	X	X	X	X	
Sydney		X			X	
Notre-Dame		X		X	X	
Colorado			X		X	
Concordia		X	X	X	X	
Wuhan and City	X	X			X	
Brigham				X	X	X
Glasgow				X		X
Arizona		X	X	X	X	
Linköping		X	X		X	
Rochester		X		X	X	
Drexel				X	X	
Kanpur and Kharagpur			X			
Cranfield			X	X	X	X
I.I.S		X		X	X	
City and Tsinghua	X				X	

According to Table 1.1, the majority of the MAV models developed so far were obtained using the wind tunnel test data in conjunction with an analytical procedure and a CFD approach. The last method, system identification, is the least common approach. It is obviously favourable to use each method in developing a model but as seen in Table 1.1, only one of the universities has combined all of the approaches. Furthermore, only three sources have nonlinear models. This fact motivates the research being done in this thesis. The next section will state the objectives and the contributions of the thesis.

1.3 Objective and Contributions

The main objective of this thesis is to propose a new methodology for obtaining PWA models using Voronoi partitions. The advantages of the proposed methodology are:

- *Smaller number of regions.* A SLP with a Voronoi partition requires significantly less regions than the UG method in the literature to meet a desired approximation error requirement.

- *Effective global analysis and synthesis tool.* Analysis of a system or a designed PWA controller can be performed in a formal and systematic manner as a set of convex optimisation problems. For example, a Lyapunov function can be determined to prove the stability of the closed-loop system. Furthermore, less regions imply less controllers.

The main contributions of this thesis are the following:

1. To propose a new methodology for obtaining PWA models using a SLP and Voronoi partitions.

2. To successfully apply the newly proposed approximation methodology to case studies including a MAV.

3. To compare the SLP approximation method with the current UG method in the literature and to show that the proposed method yields a smaller number of regions for the same desired model accuracy in several case studies.

1.4 Structure of the Thesis

The thesis is structured as in Figure 1.3. In Chapter 2, a general PWA system is defined followed by two methods that can be employed to partition the state space. Next, the algorithm to obtain a PWA model for smooth curves using a SLP is pro-

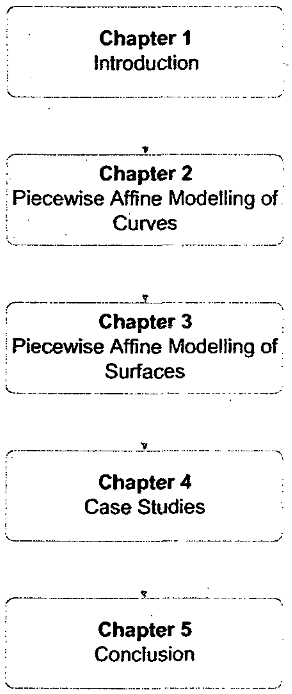


Figure 1.3: Structure of the Thesis

posed and then compared against the UG approximation method in the literature. Subsequently, the SLP approximation method is extended to smooth surfaces using a Voronoi partition in Chapter 3. Next, Chapter 4 applies the methodology presented in Chapter 3 to three case studies: a mechanical system with nonlinear spring and damper elements, a 3 DOF rotorcraft model, and a simplified model of a MAV. Finally, conclusions are drawn in Chapter 5.

Chapter 3 and part of Chapter 4 are mainly based on the following papers:

- Scott Casselman and Luis Rodrigues. Piecewise Affine Modelling of a Micro Air Vehicle using Voronoi Partitions. In *Proceedings of the 10th European Control Conference*, pages 3857-3862, 2009.

- Scott Casselman and Luis Rodrigues. "A New Methodology for Obtaining Piecewise Affine Models Using Voronoi Partitions," accepted for publication in *Proceedings of the 48th IEEE Conference on Decision and Control*, Shanghai, China, December 16-18, 2009.

Chapter 2

Piecewise Affine Modelling of Curves

2.1 Introduction

In this chapter, a general PWA system is defined. First, two methods will be described to partition the state space of a nonlinear system and approximate it by a PWA system. The method reviewed in Section 2.3 is the UG method. This method is currently being widely used in the literature of PWL [14, 15, 16, 10, 18, 19] and PWA [6, 22, 23, 24, 26] approximations PWA approximations. The UG method uses a simplicial partition for the approximation because any continuous function can be approximated uniformly by a PWA function on compact domains over simplicial partitions [6]. Alternatively, a new approximation method called the SLP method will be proposed in Section 2.4. To the best of the author's knowledge, there has not been any formal work dedicated to approximating smooth nonlinear control systems as PWA systems using a SLP and comparing it to other alternatives. This is one of the main contributions of this thesis. As a first step, an automated procedure to approximating smooth curves is presented using a SLP. Finally, the two approximation methods in Sections 2.3 and 2.4 are compared using the algorithm proposed in Section 2.5.

2.2 Piecewise Affine Systems

In Chapter 4, we will see that an aircraft model can be written in the form

$$\dot{x} = Ax + f(x) + Bu \tag{2.1}$$

where $x \in \mathbb{R}^n$ is the state vector and $A \in \mathbb{R}^{n \times n}$ and $B \in \mathbb{R}^{n \times m}$ are constant matrices. Moreover, it is assumed that $f(x)$ is a known smooth (or at least of class C^2) nonlinear function of the state. The reasons for restricting the class of nonlinear functions become apparent later in Section 2.4.1. The objective in this thesis is to obtain a global approximation of systems in the form (2.1) that is both simpler than system (2.1) and able to predict the behaviour of the original system. A PWA system is described by

$$\dot{x} = A_i x + a_i + B_i u, \text{ for } x \in \mathcal{R}_i \tag{2.2}$$

where $A_i \in \mathbb{R}^{n \times n}$, $a_i \in \mathbb{R}^n$, $B_i \in \mathbb{R}^{n \times m}$ for $\mathcal{R}_i$, $i \in \mathcal{I} = \{1, ..., M\}$. The regions $\mathcal{R}_i$ partition the state space such that each region is constructed as the intersection of a finite number (p_i) of half spaces defined by

$$\mathcal{R}_i = \{x \mid \bar{E}_i \bar{x} \succ 0\} \tag{2.3}$$

where $\bar{E}_i = \begin{bmatrix} E_i & e_i \end{bmatrix} \in \mathbb{R}^{p_i \times (n+1)}$ and $\bar{x} = \begin{bmatrix} x & 1 \end{bmatrix}^T$. Any two polytopic cells sharing a common facet are called *level-1* neighbouring cells. Let $\mathcal{N}_i = \{$*level-1* neighbouring cells of $\mathcal{R}_i\}$. In addition, we assume that there exists vectors $h_{ij} \in \mathbb{R}^n$ and scalars g_{ij} such that the facet boundary between cells $\mathcal{R}_i$ and $\mathcal{R}_j$ is contained in the hyperplane described by $\{x \in \mathbb{R}^n \mid h_{ij}^T x + g_{ij} = 0\}$ for $i = 1, ..., M$, $j \in \mathcal{N}_i$. As a result, the parametric description of the boundaries can be obtained as [5] (refer to Figure 2.1)

$$\overline{\mathcal{R}_i} \cap \overline{\mathcal{R}_j} \subseteq \left\{ x \mid \bar{x} = \bar{F}_{ij}\bar{s} \mid s \in \mathbb{R}^{n-1}, \bar{s} = \begin{bmatrix} s \\ 1 \end{bmatrix} \right\} \tag{2.4}$$

where $\overline{\mathcal{R}_i}$ indicates the closure of $\mathcal{R}_i$, $\bar{F}_{ij} = \begin{bmatrix} F_{ij} & l_{ij} \\ 0 & 1 \end{bmatrix}$, $F_{ij} \in \mathbb{R}^{n\times(n-1)}$ is a full rank matrix whose columns span the null space of h_{ij} and $l_{ij} \in \mathbb{R}^n$ can be written as $l_{ij} = -h_{ij}(h_{ij}^T h_{ij})^{-1} g_{ij}$.

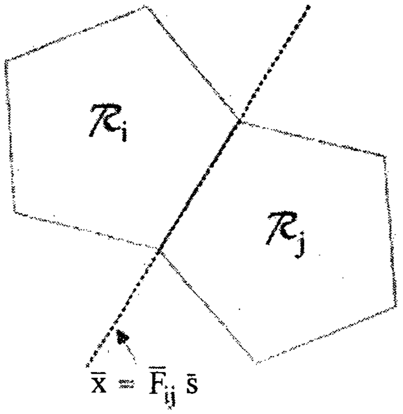


Figure 2.1: Polytopic Regions $\mathcal{R}_i$ and $\mathcal{R}_j$ and Boundary (adapted from [5])

In order to approximate System (2.1) by (2.2), the nonlinearity $f(x)$ needs to be approximated by a PWA function. To proceed, a method to partition the space of variables in the domain of the nonlinearity into polytopic regions is required as shown in Figure 2.2. The next two sections will describe two methods that can be used to partition the domain of $f(x)$ and hence obtain an approximation of $f(x)$ as a PWA function. Section 2.3 will review the UG method. Subsequently, Section 2.4 will propose a new partitioning method that uses a SLP.

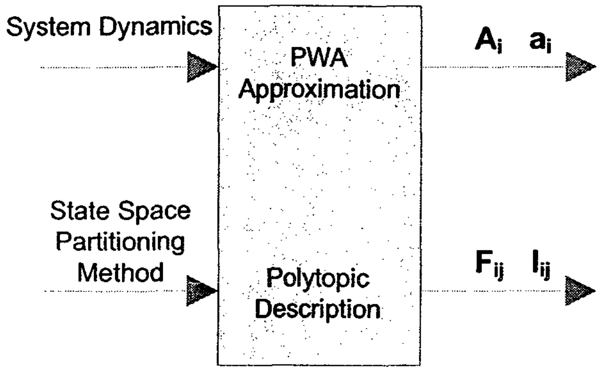


Figure 2.2: PWA Approximation Method (adapted from [6])

2.3 Method 1: Uniform Grid

The first method uses a uniform rectangular grid to partition the space of variables in the domain of the nonlinearity. A UG is the most widely used method in the literature of PWL [14, 15, 16, 10, 18, 19] and PWA [6, 22, 23, 24, 26] approximations. A simplicial partition is used to partition the state space into polytopic cells. A simplex is defined as follows

Definition 2.3.1 *A simplex in $\mathbb{R}^n$ is defined as the convex hull of $n+1$ affinely independent points. The convex hull of a set $\mathcal{S}$ is the smallest convex set that contains $\mathcal{S}$.*

For example, a simplex in $\mathbb{R}^2$ is a triangle, and in $\mathbb{R}^3$ is a tetrahedron. The procedure for computing a PWA approximation proposed in [6] is as follows

1. Order all the vertices of the chosen uniform rectangular grid.

2. Group the vertices into simplicial cells.

3. Obtain a polytopic description for each cell.

4. Obtain a parametric description for each boundary

5. Determine the PWA approximation of $f(x)$ and the piecewise constant approximation of the input matrix B within each cell.

After $f(x)$ from the nonlinear system (2.1) is replaced by its PWA approximation, a PWA system of the form (2.2) is obtained. The greatest disadvantage of the UG method is the potential large amount of regions required to obtain a desired accuracy for systems with more than one variable in the domain of the nonlinearity. The next section will describe the SLP method to approximate and partition the space of the nonlinearity in order to reduce the number of regions in the partition.

2.4 Method 2: Set of Linearisation Points

The method proposed in this thesis to partition the space of the nonlinearity and obtain a PWA approximation is called SLP. The idea is to compute several linearisations of (2.1), each linearisation being valid for a particular region. The SLP method is based on the first order Taylor series. A first order Taylor series approximation involves the use of the function value and its derivative at a single point X_i. Given a function $f(X)$ and a set $\mathcal{X} = \{X_1, ..., X_n\}$ of n distinct points, the linear approximation $f_{L_i}(X)$ is

$$f_{L_i}(X) = f(X_i) + \left.\frac{\partial f}{\partial X_i}\right|_{X_i} \cdot (X - X_i) \text{ for } i = 1, ...n \tag{2.5}$$

In order to select the linearisation points for an approximation, it is proposed in this thesis to use the curvature of the nonlinearity. This choice is made due to the correlation between the curvature and the gradient of a function. A function with zero curvature is linear, as well will see in Section 2.4.2. Before attempting to approximate a smooth surface (which is the case as we will see in the aircraft model that will be presented in Chapter 4), it is necessary to begin with a simpler case, ie a smooth curve. The approximation of a smooth curve using (2.5) and its curvature will be explained in the next section.

2.4.1 A Smooth Curve in $\mathbb{R}^2$ and its Curvature

In order to motivate the approximation of a smooth curve using a SLP and its curvature, consider the aircraft model (4.14), which will be presented later in Chapter 4. Assume that there is no side slip ($v = 0, \dot{v} = 0$ and $\beta = 0$). The system can now be simplified and written as

$$\begin{aligned}\dot{r} &= -\frac{k_r}{I_{zz}}r + \frac{Q_m}{I_{zz}} - \frac{T_t l_t}{I_{zz}} \\ \dot{\psi} &= r \\ \dot{y} &= u_0 \sin(\psi)\end{aligned} \tag{2.6}$$

Evidently the system of equations is nonlinear in only a single variable, ψ. In order to approximate the nonlinear function $\sin(\psi)$, it is proposed to investigate the curvature property of smooth curves. A smooth curve in $\mathbb{R}^2$ is defined as follows [79]:

Definition 2.4.1 *A smooth curve in $\mathbb{R}^2$ is a smooth function $f(x)$ that is continuously differentiable over a specified interval on x (Figure 2.3). The curve can be reparameterised by the position vector $\vec{r}(x)$ such that $\vec{r}(x) = (x, f(x))$. The velocity,*

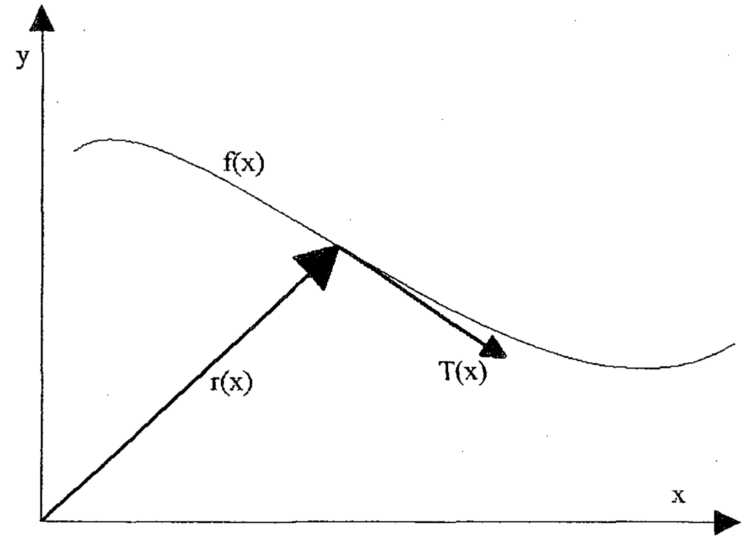


Figure 2.3: Smooth Curve in $\mathbb{R}^2$

speed and acceleration of the curve can now be defined as:

$$\dot{\vec{r}}(x) = (1, \dot{f}(x)) \tag{2.7}$$

$$\left\|\dot{\vec{r}}(x)\right\| = \sqrt{1 + \dot{f}(x)^2} \tag{2.8}$$

$$\ddot{\vec{r}}(x) = (0, \ddot{f}(x)) \tag{2.9}$$

where $\|\cdot\|$ is the Euclidean norm. The curvature of a smooth curve is defined as the length of the acceleration vector computed relative to the arc length when $\vec{r}(x)$ traces the curve at a constant unit speed. In other words, the curvature $k(x)$ can be written as

$$k(x) = \left\|\frac{d\vec{T}(x)}{ds(x)}\right\| \tag{2.10}$$

$$= \left\|\frac{d\vec{T}(x)}{dx} \cdot \frac{dx}{ds(x)}\right\| \tag{2.11}$$

$$= \frac{\left\|\frac{d\vec{T}(x)}{dx}\right\|}{\left\|\frac{ds(x)}{dx}\right\|} \tag{2.12}$$

where $\vec{T}(x)$ and $s(x)$ are the unit tangent vector and arc length, respectively, defined by

$$\vec{T}(x) = \frac{\dot{\vec{r}}(x)}{\left\|\dot{\vec{r}}(x)\right\|} \tag{2.13}$$

$$s(x) = \int_0^x \left\|\dot{\vec{r}}(x)\right\| \, dx \tag{2.14}$$

Finally, substituting (2.13)-(2.14) into (2.12) gives

$$k(x) = \frac{\left\|\dot{\vec{T}}(x)\right\|}{\left\|\dot{\vec{r}}(x)\right\|} \tag{2.15}$$

Since the $k(x)$ is proportional to $\left\|\dot{\vec{T}}(x)\right\|$, the second derivative must exist and be continuous. Hence $f(x)$ must be smooth or at the very least of class C^2. The next section will focus on a strategy to automate the approximation of a smooth curve by a PWA function, which will be extended to the approximation of surfaces later in Chapter 3.

2.4.2 Automated PWA Approximation Procedure of a Smooth Curve in $\mathbb{R}^2$ using a SLP

As mentioned previously in Section 2.4.1, the curvature will be used as a reference to selecting linearisation points. The approximation algorithm is structured in Figure 2.4.

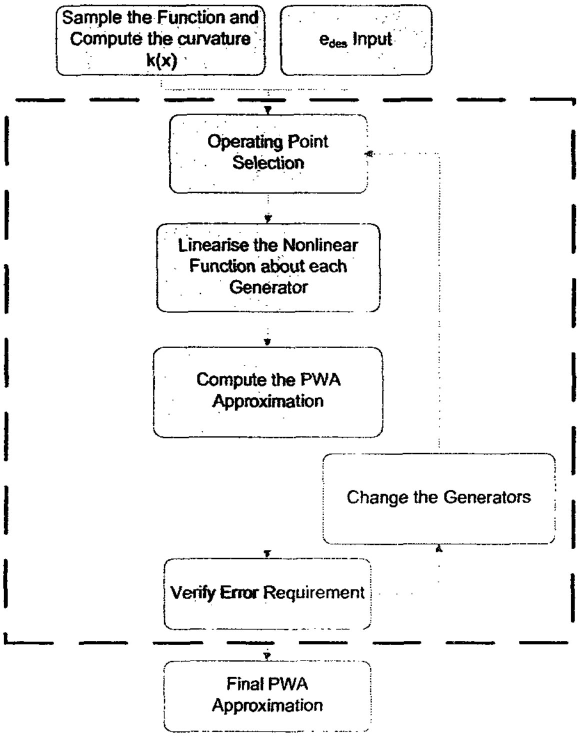

Sample the Function and Compute the curvature k(x)
e_{des} Input
Operating Point Selection
Linearise the Nonlinear Function about each Generator
Compute the PWA Approximation
Change the Generators
Verify Error Requirement
Final PWA Approximation

Figure 2.4: Smooth Curve Approximation Procedure

1. **User Input**: First, the smooth curve is sampled using a UG and the curvature $k(x)$ is computed. Additionally, an error requirement, e_{des}, is chosen to determine the accuracy of the approximation.

2. **Initial Point Selection**: The initial selection of linearisation points is selected based upon the curvature. Three different cases arise from inspecting the curvature.

(a) **Nonzero constant curvature**: The first case is nonzero constant curvature. If the curvature is constant, then an initial generator is placed at the centroid of the grid. The spacing between each subsequent linearisation point is chosen to be inversely proportional to the curvature. Assuming that the constant curvature is k, then the spacing Δ_x is (refer to Figure 2.5)

$$\Delta_x = \frac{c}{k} \tag{2.16}$$

where $c : (0, 1]$ is a constant that is used to fine tune the spacing of the points. The initial approximation will take $c = 1$, which corresponds to the coarsest approximation. If there is no nonzero constant mean curvature found, the algorithm checks for case 2.

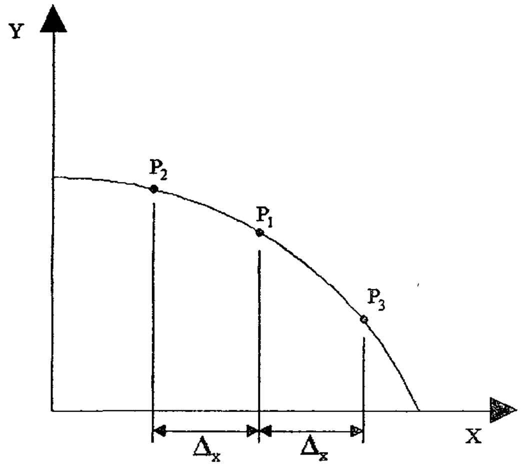


Figure 2.5: Placement of Linearisation Points on a Curve with Nonzero Constant Curvature

(b) **Zero curvature**: The second case is that of zero curvature. If the calculated curvature is zero for all x in the sampled grid, then a single linearisation point is required and it is placed at the center of the sampled grid. If the calculated curvature is zero for some of the sampled points, then a linearisation point is placed at each of these points.

(c) **Neither zero nor nonzero constant curvature exists**: If case 1 and case 2 fail, then linearisation points are placed at the locations corresponding to the minimum of the absolute value(s) of the curvature.

3. **Linearisation**: The nonlinear function $f(x)$ is then linearised about each linearisation point using (2.5).

4. **PWA Approximation Computation**: Subsequently, the PWA approximation is computed and therefore the maximum approximation error, e_{max}, is obtained. Here e_{max} is defined as

$$e_{max} = \max_i \left[\sup_{x \in \mathcal{R}_i} \left(\left\| f(x) - (A_i x + a_i) \right\| \right) \right] \tag{2.17}$$

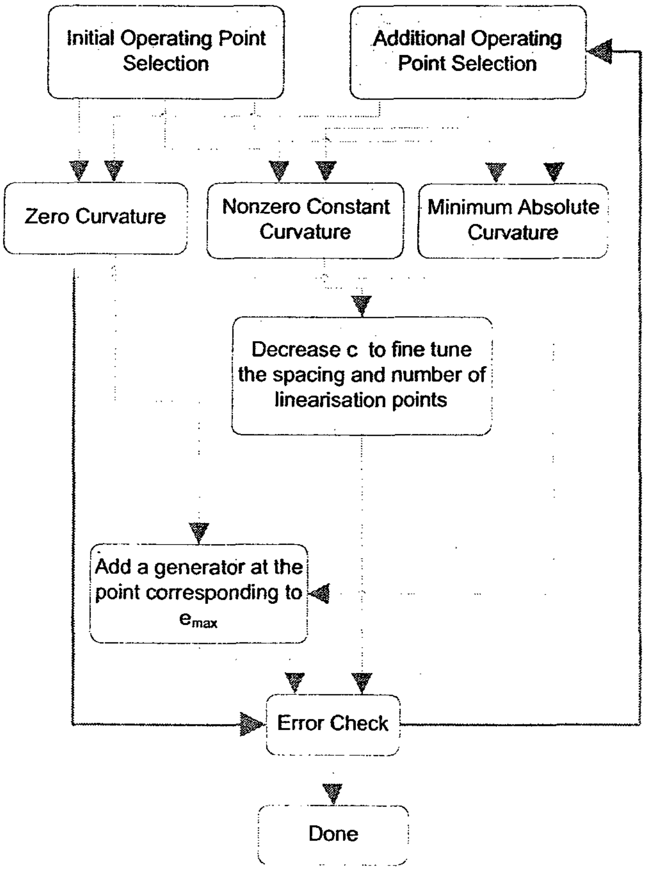

Initial Operating Point Selection
Additional Operating Point Selection
Zero Curvature
Nonzero Constant Curvature
Minimum Absolute Curvature
Decrease c to fine tune the spacing and number of linearisation points
Add a generator at the point corresponding to e_{max}
Error Check
Done

Figure 2.6: Methodology for adding new generators.

5. **Error Requirement Verification**: The approximation error is then compared to the desired error requirement, e_{des}. If the e_{des} requirement is met, then the PWA model is obtained.

6. **Addition of Operating Points**: If the e_{des} requirement is not met, the approximation procedure repeats itself by including additional operating points, as follows (refer to Figure 2.6):

   (a) **Case 1**: For the case of nonzero constant curvature, additional points are added by decreasing c in order to have a finer approximation.

   (b) **Case 2 and 3**: For cases 2 and 3, additional points are added where e_{max} occurs until the approximation meets the error requirement.

Before applying the proposed methodology to system (2.6), a few simple cases will be used to verify the algorithm. Figures 2.7 and 2.8 depict the results of approximating a straight line and a circular section, respectively, with varying error requirements. For the first case, the straight line is approximated exactly using one linearisation point at the centroid of the sampled grid. The second case involves the approximation

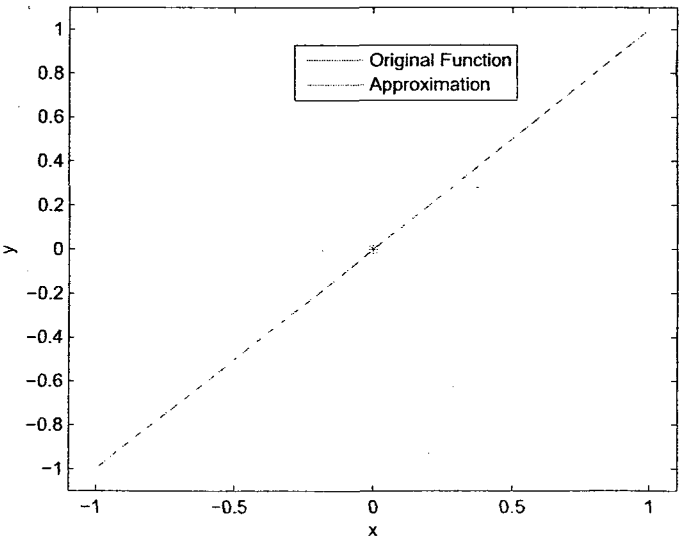
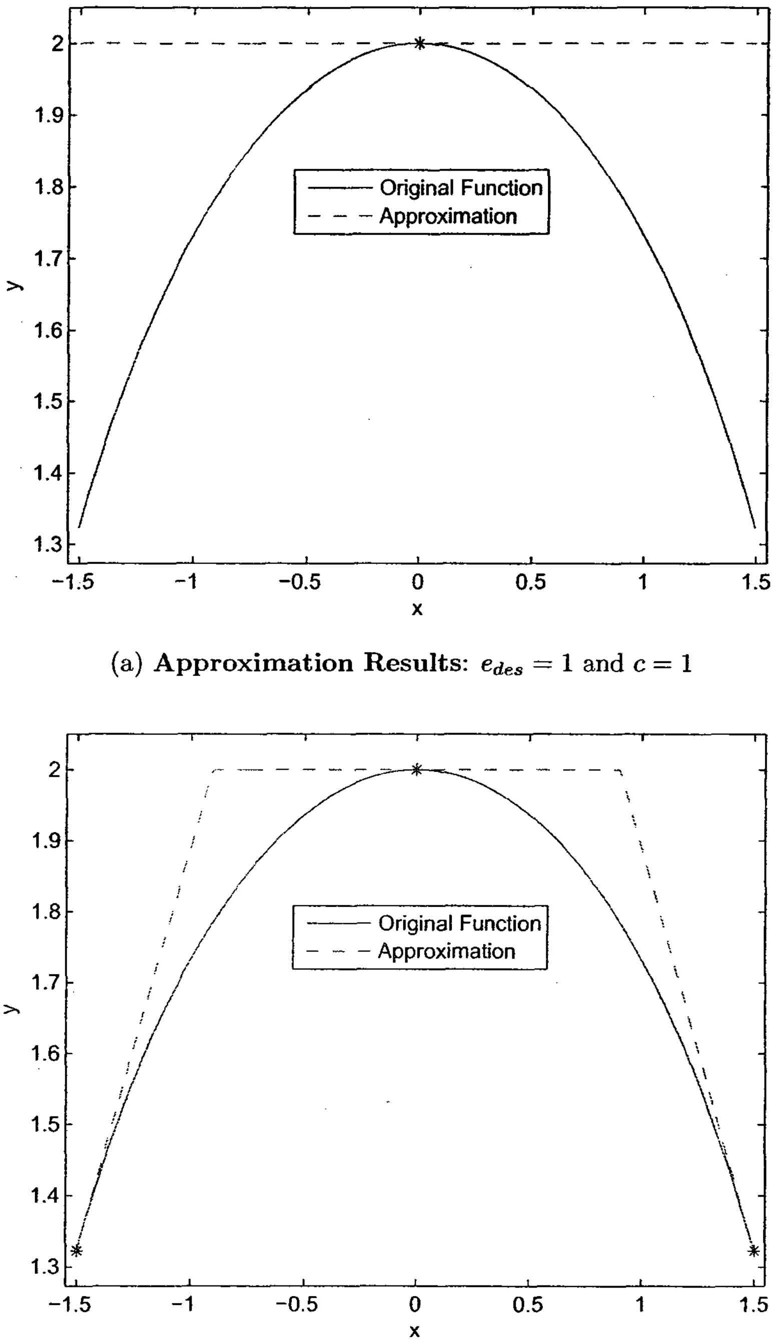


Figure 2.7: Approximation of the Straight Line $y = x$.

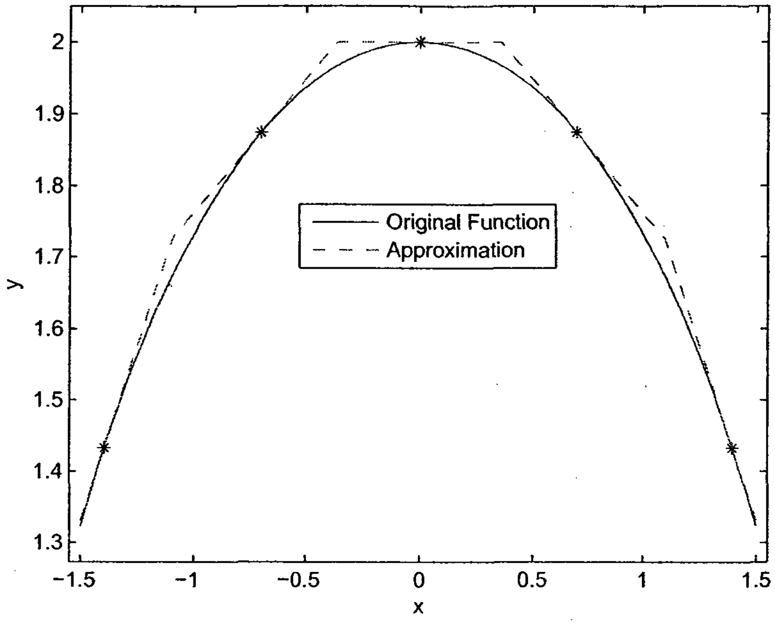


(a) **Approximation Results**: $e_{des} = 1$ and $c = 1$

(b) **Approximation Results**: $e_{des} = 0.5$ and $c = 0.75$

Figure 2.8: Approximation of a Circular Section of $x^2 + y^2 = 4$.

of a circular curve. Figures 2.8a-2.8c show the approximation results over a range of error requirements. As expected, as the error requirement decreases, c decreases in order to allow for a finer approximation. With two simple cases verified, the algorithm can now be applied to the the nonlinearity $\sin(\psi)$ in system (2.6). The approximation of f, $\hat{f}$, is shown for three different error requirements in Figure 2.9. The first approximation consists of linearisation points P_1 and P_2, which are the zero curvature points found within the sampled domain. The second approximation is obtained by adding points P_3-P_5 to the original two points. Notice that these new linearisation points are indeed placed at the points of maximum error as indicated in the algorithm. The third approximation is a result of adding points P_6-P_9 to the previous five linearisation points. As in the secondary approximation, the new points are added at the locations corresponding to the maximum error. With the algorithm verified for smooth curves, the SLP and UG approximation methods will be compared in the next section.



(c) **Approximation Results**: $e_{des} = 0.075$ and $c = 0.35$

Figure 2.8: (Continued)

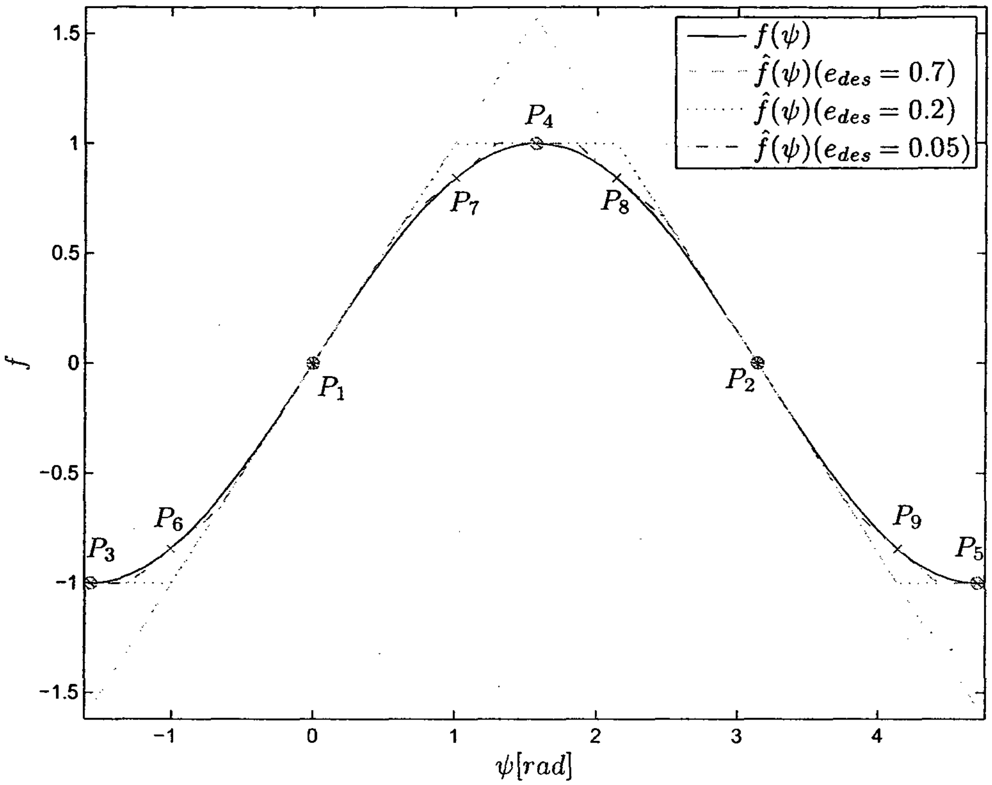


Figure 2.9: SLP Approximation of $f(\psi) = \sin(\psi)$ over $-\frac{\pi}{2} \leq \psi \leq \frac{3\pi}{2}$.

2.5 Comparison Algorithm

To compare the two approximation methods, the following algorithm is proposed and shown in Figure 2.10:

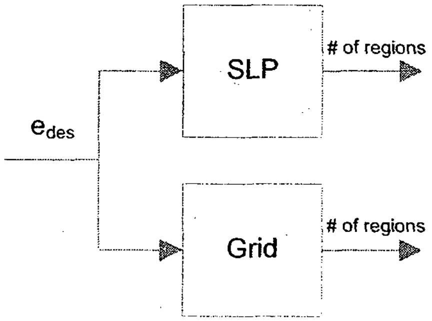


Figure 2.10: Block Diagram of the Comparison Algorithm.

1. Choose a desired approximation error, e_{des}.

2. (a) SLP: Compute the PWA approximation and the corresponding maximum error, e_{max}, using (2.5) and (2.17), respectively.

   (b) UG: Compute the PWA approximation and e_{max} for a given UG using the algorithm in section 2.3.

3. (a) SLP: Repeat step two while $e_{max} > e_{des}$ by varying the number and location of points using the algorithm in Section 2.4.2.

   (b) UG: Repeat step 2 while $e_{max} > e_{des}$ using the Toolbox in [28].

4. Compare the number of regions that each method needed to meet the desired error requirement.

2.5.1 Curve Approximation Comparison and Results

The objective of this section is to compare the SLP and UG methods using the algorithm in Section 2.5 applied to system (2.6) for two cases. In the first case, the UG and SLP methods will be compared over $-\frac{\pi}{2} \leq \psi \leq \frac{3\pi}{2}$ while the second case will use a range of $-\pi \leq \psi \leq \pi$.

- Case 1: $-\frac{\pi}{2} \leq \psi \leq \frac{3\pi}{2}$

  The comparison results for case 1 can be seen in Table 2.1 for three different e_{des} as in Section 2.4.2. Moreover, Figures 2.9 and 2.11 depict the approximation results using a SLP and a UG, respectively. The main result of case

Table 2.1: Curve Approximation Results for Case 1

	SLP Approximation		UG Approximation	
e_{des}	e_{max}	Number of Regions	e_{max}	Number of Regions
0.7	**0.57**	2	**0.57**	2
0.20	**0.15**	5	**0.13**	5
0.05	**0.046**	9	**0.031**	9

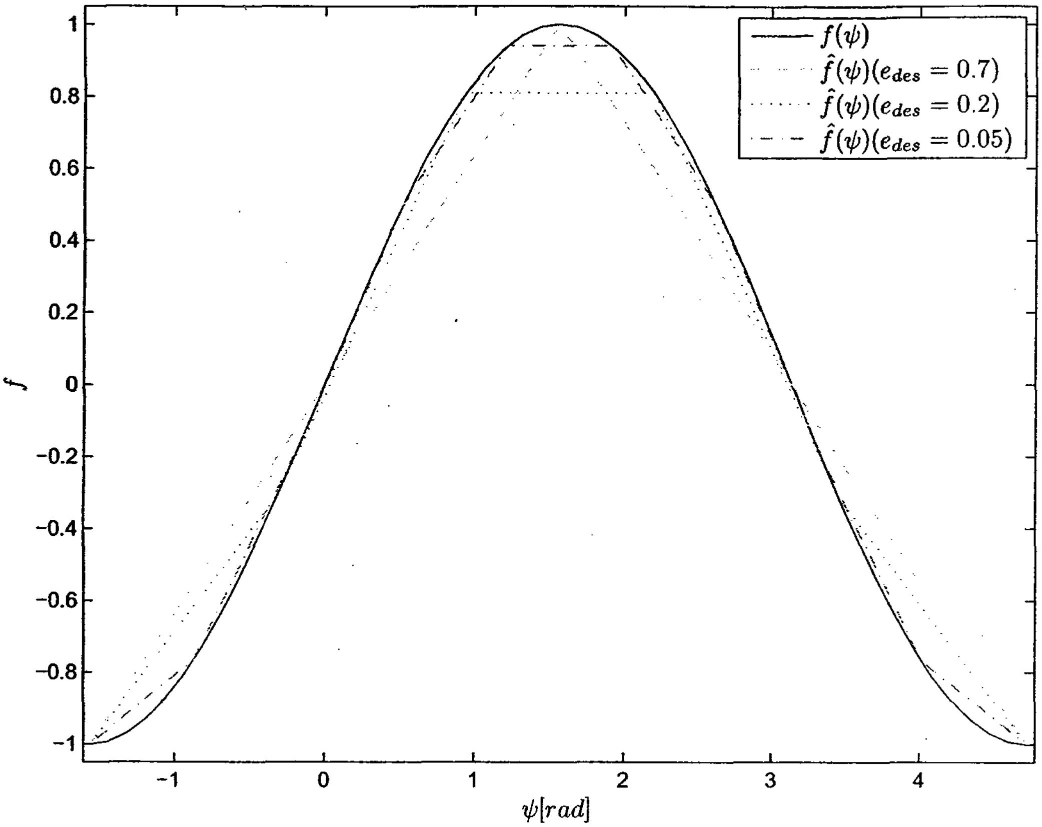


Figure 2.11: UG Approximation of $f(\psi) = \sin(\psi)$ over $-\frac{\pi}{2} \leq \psi \leq \frac{3\pi}{2}$.

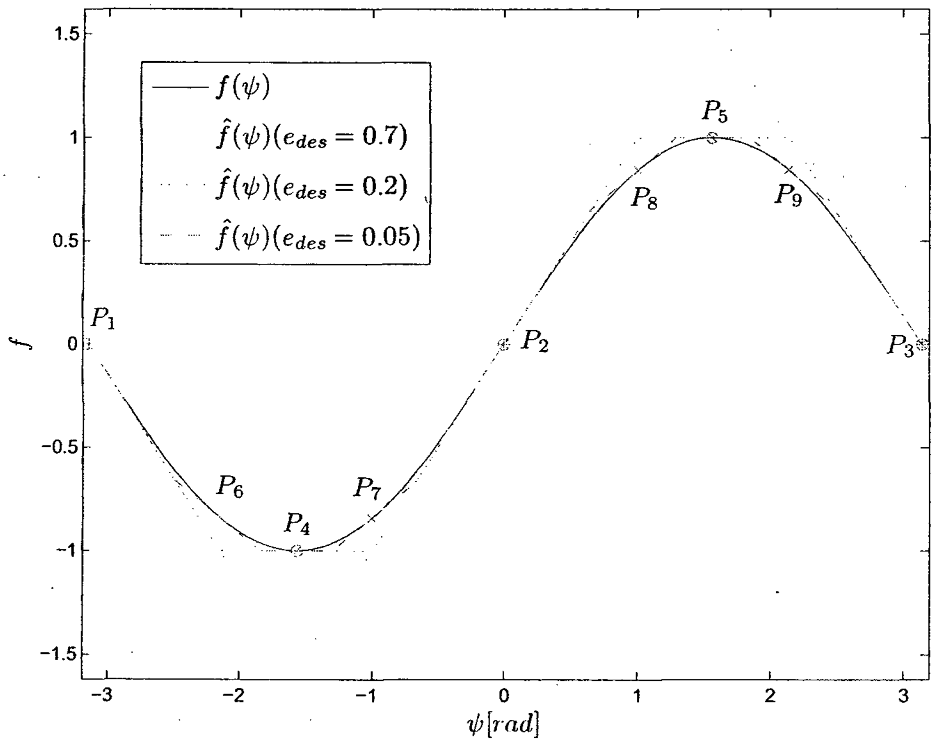
1 is that the SLP and UG approximation methods use the same number of regions. The difference between the two approximation methods is the e_{max} for the second and third e_{des}. Although the UG approximation has a smaller maximum approximation error than the SLP approximation, the SLP method guarantees that the slope and value of the PWA approximation are equivalent to the nonlinear function at each linearisation point.

- Case 2: $-\pi \leq \psi \leq \pi$

The comparison results for case 2 can be seen in Table 2.2 for the same e_{des} as in case 1. Moreover, Figure 2.12 depicts the approximation results using a SLP and a UG. It is interesting to note that the SLP approximation results for case 2 are consistent with the results presented for case 1 in that the e_{max} are the same for each e_{des}. Additionally, with the exception of $e_{des} = 0.7$, the same number of regions are required in the SLP approximation for cases 1 and 2. The UG approximation results for case 2 do not have the same consistency in e_{max} than the SLP approximation results. Moreover, the UG approximation for the last e_{des} requires one more region than the SLP approximation.

Table 2.2: Curve Approximation Results for Case 2

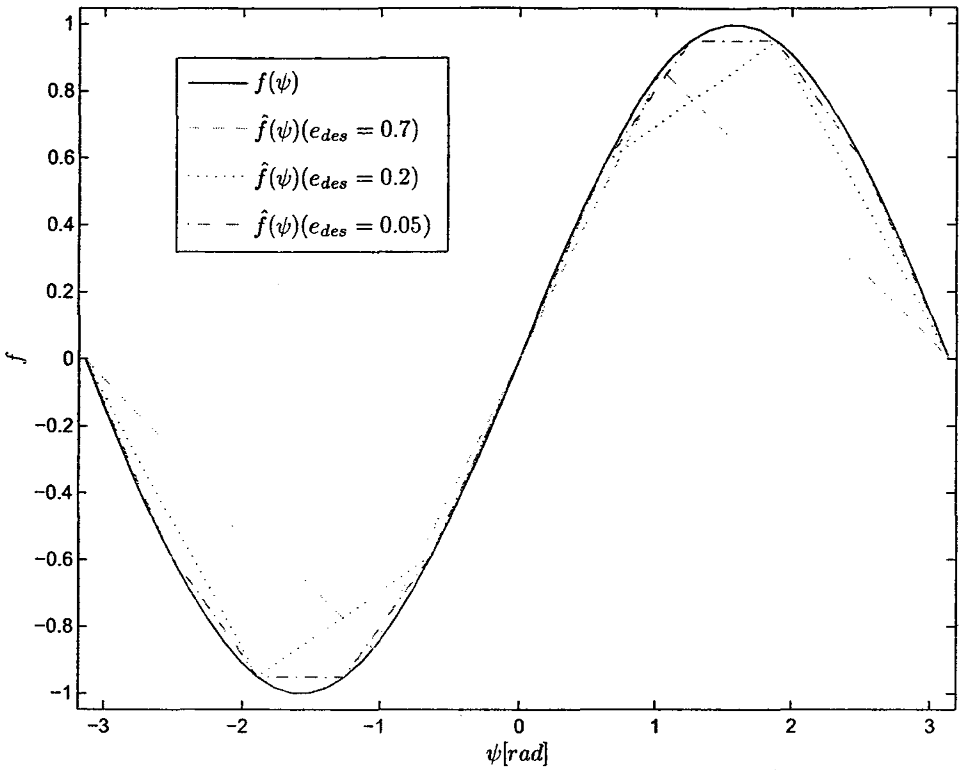
	SLP Approximation		UG Approximation	
e_{des}	e_{max}	Number of Regions	e_{max}	Number of Regions
0.7	**0.57**	**3**	**0.437**	**3**
0.2	**0.15**	**5**	**0.18**	**5**
0.05	**0.046**	**9**	**0.047**	**10**



(a) **SLP Approximation Results**

Figure 2.12: SLP and UG Approximation of $f(\psi) = \sin(\psi)$ over $-\pi \leq \psi \leq \pi$.

It should be noted that the goal of this thesis is to reduce the complexity of PWA models that have more than one variable in the domain of the nonlinearity.



(b) **UG Approximation Results**

Figure 2.12: (Continued)

With the SLP approximation method successfully applied to system (2.6), the focus can be shifted to extending the algorithm in Section 2.4.2 to include smooth surfaces, which is the topic of Chapter 3.

2.6 Summary

In summary, this chapter develops the framework needed to obtain a PWA model using the SLP method. First, the existing UG approximation method is reviewed and a new approximation technique called SLP is proposed. Second, the SLP approximation method is applied to smooth curves and then validated for several examples in Section 2.4.2. Finally, the UG and SLP methods are compared in Section 2.5.1 using the algorithm proposed in Section 2.5. The main result of this comparison is that the SLP method gives consistent approximation results while the UG method gives inconsistent approximation results for a nonlinear function with two different sampled domains.

Chapter 3

Piecewise Affine Modelling of Surfaces

3.1 Introduction

In this chapter, the SLP procedure proposed in Chapter 2 for smooth curves is extended to smooth surfaces. Moreover, the Voronoi partition is proposed to partition the domain of a nonlinear function. Subsequently, the SLP and UG approximation methods are compared using a modified version of the algorithm proposed in Section 2.5.

3.1.1 A Smooth Surface in $\mathbb{R}^3$ and its Curvature

In order to compute the curvature of a smooth surface in $\mathbb{R}^3$, one would expect, as with curves, to study the changes in the tangent planes. However, since the orientation of a tangent plane can be described by its normal vector, we can simply compute how the normal vector field is changing at each point on the surface [7]. Curvature of surfaces can also be extended to $\mathbb{R}^n$. For the purpose of this thesis, only smooth surfaces in $\mathbb{R}^3$ are considered since the function $f(\psi, v)$ in (4.14) is nonlinear in two variables, as we will see later in Chapter 4. The change in the normal vector field is obtained from the Weingarten map, L, which is defined as follows (refer to Figure 3.1) [79]:

Definition 3.1.1 *A smooth surface in $\mathbb{R}^3$ is a smooth function $f(x, y)$ that is continuously differentiable over a specified interval on x and y. Let $\mathcal{M} \subset \mathbb{R}^3$ be a smooth surface parameterised such that $\mathcal{M} = (p, f(p))$ where p is a point $\in \mathcal{M}$ with coordinates (x, y). The Weingarten map of $\mathcal{M}$ at p is the linear map $L : T_p\mathcal{M} \rightarrow T_p\mathcal{M}$ written as*

$$L(p) = -\nabla N(p) \tag{3.1}$$

where T_p is the tangent space of $\mathcal{M}$ at p and N is the unit normal vector field defined by

$$N = \frac{\frac{\partial \mathcal{M}}{\partial x} \times \frac{\partial \mathcal{M}}{\partial y}}{\left\| \frac{\partial \mathcal{M}}{\partial x} \times \frac{\partial \mathcal{M}}{\partial y} \right\|} \tag{3.2}$$

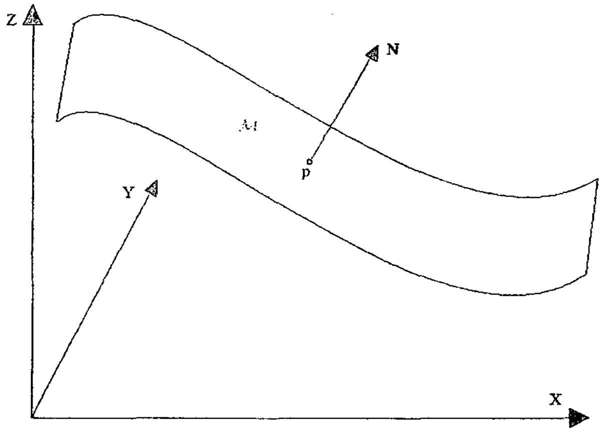


Figure 3.1: Smooth surface $\mathcal{M}$ and its unit normal vector at point p.

In order to assign a single value of curvature to each point on $\mathcal{M}$, a single value is required from the Weingarten map. This assignment can be done by either obtaining the determinant or the trace of L and they are defined in the following [7, 79]:

Definition 3.1.2

1. *The determinant $K(p)$ is called the Gaussian curvature where*

$$K(p) = \det L = k_1(p)k_2(p) \tag{3.3}$$

2. *The average value $H(p)$ is called the mean curvature where*

$$H(p) = \frac{1}{2}Tr(L) = \frac{k_1(p) + k_2(p)}{2} \tag{3.4}$$

3. *A point p on a surface is called*

elliptic	*if* $K(p) > 0$
hyperbolic	*if* $K(p) < 0$
parabolic	*if* $K(p) = 0$ *and* $H(p) \neq 0$
a level point	*if* $K(p) = H(p) = 0$
Special Case	
umbilic	*if* $k_1(p) = k_2(p) \neq 0$

where $k_1(p)$ and $k_2(p)$ are the principal curvatures of $\mathcal{M}$ at p.

Examples of elliptic, hyperbolic and parabolic surfaces can be seen in Figure 3.2. An ellipsoid has only elliptic points, while a hyperboloid has only hyperbolic points. On the other hand, a circular cylinder only has parabolic points, a sphere has only umbilic points, while a plane consists only of level points. The reader is referred to [79, 80, 7] for more detailed explanations. The next step is to extend the approximation algorithm in Section 2.4.2 to surfaces. Recall that there exists two measures of curvature for surfaces: Gaussian and mean curvature. For the purpose of this thesis, the Gaussian curvature is disregarded and the mean curvature will be taken as the curvature of the nonlinearity. The reason for this selection is based on the approximation algorithm in Section 2.4.2 and the property of parabolic surfaces. For example, assume that the surface to be approximated is parabolic and the Gaussian curvature was selected instead of the mean curvature. Then, according to the approximation algorithm and the definition of a parabolic surface (the Gaussian curvature

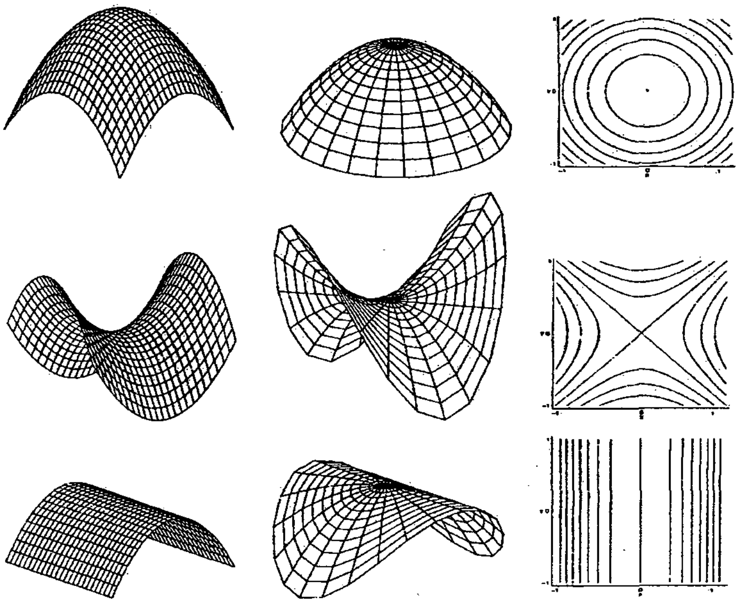


Figure 3.2: Elliptic (1st row), hyperbolic (2nd row) and parabolic (3rd row) surfaces with level curves [7].

is zero), a single linearisation point would be placed at the centroid of the sample grid. Of course this would not be a good approximation because the curvature is varying in a single direction. It is proposed in this thesis to use the Voronoi partition to partition the domain of the nonlinearity based on the set of linearisation points $\mathcal{X}$ (defined at the beginning of Section 2.4) for surfaces. To the best of the author's knowledge, there has not been any formal work dedicated to partitioning the domain of a nonlinear system with a Voronoi partition to obtain a PWA system. This is one of the main contributions of this thesis.

3.1.2 Voronoi SLP Approximation: Automated Generation Procedure

This section will describe in detail the proposed algorithm that will be used in this thesis to automatically approximate a given smooth surface. The approximation algorithm is structured in Figure 3.3. The only subtle difference between the curve and surface approximation algorithms is the addition of the Voronoi partition computation

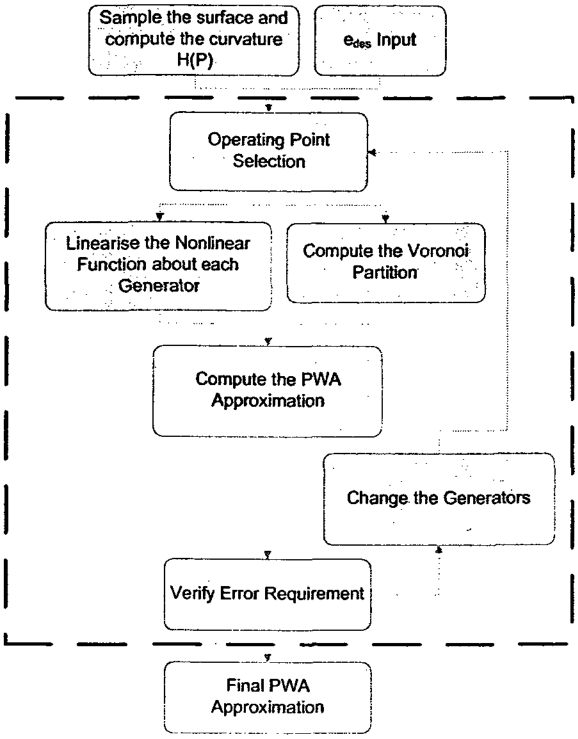

Sample the surface and compute the curvature H(P)
e_{des} Input
Operating Point Selection
Linearise the Nonlinear Function about each Generator
Compute the Voronoi Partition
Compute the PWA Approximation
Change the Generators
Verify Error Requirement
Final PWA Approximation

Figure 3.3: The Voronoi Partition Modelling Procedure

in Figure 3.3 as compared to Figure 2.4. Also, mean curvature as defined by (3.4) replaces curvature in this algorithm. In order to proceed it is necessary to review Voronoi partition theory.

The Voronoi Partition

A Voronoi partition or diagram is defined as follows [81]:

Definition 3.1.3 *Given a domain $\mathcal{S} \subset \mathbb{R}^n$ and $\mathcal{X} \subset \mathcal{S}$ of n distinct points (often called generators), the Voronoi partition of $\mathcal{S}$ is the subdivision of the domain into n cells defined by*

$$\mathcal{V}(X_i) = \left\{ q \in \mathcal{S} \middle| \, \|q - X_i\| < \|q - X_j\| \right\}, \, \forall X_j \in \mathcal{X} \text{ with } j \neq i \tag{3.5}$$

where $\mathcal{V}(X_i)$ is the Voronoi cell corresponding to generator X_i.

Figure 3.4 (adapted from [8]) illustrates a two dimensional Voronoi partition (denoted by the bold lines) as well as its geometric dual called Delaunay triangulation (denoted by the lighter lines). From this figure, it can clearly be seen that the outer Voronoi cells $(\mathcal{V}(P_4), \mathcal{V}(P_5), \mathcal{V}(P_6), \mathcal{V}(P_7), \mathcal{V}(P_8), \mathcal{V}(P_9))$ are unbounded. For the purpose of this thesis, since all physical variables are bounded, a bounding polytope will be added to enclose the outer Voronoi cells. The bounding polytope $\mathcal{B}$ is defined by its vertices as

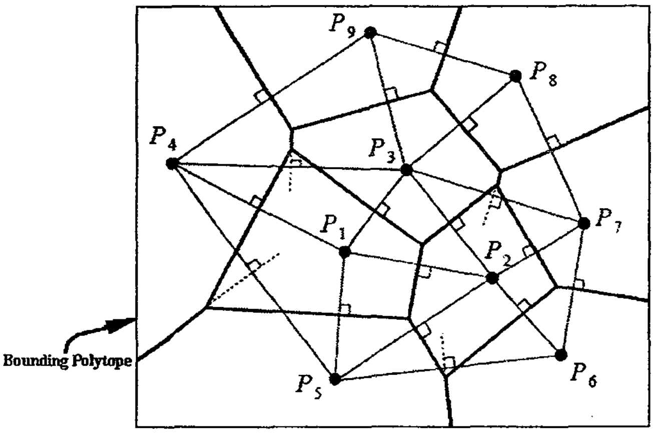


Figure 3.4: Sample Voronoi partition (adapted from [8]).

$$\mathcal{B} = \left\{ x \in \mathbb{R}^n \mid x = \sum_{i=1}^{\nu_p} \alpha_i V_p^{(i)},\ 0 \leq \alpha_i \leq 1,\ \sum_{i=1}^{\nu_p} \alpha_i = 1 \right\} \tag{3.6}$$

where $V_p^{(i)}$ is the i-th vertex of $\mathcal{B}$ and ν_p is the total number of vertices. The strategy that will be adopted in this thesis is to make the generators be the linearisation points.

1. **User Input**: In the first step of the surface approximation algorithm, the surface is sampled using a UG and the mean curvature is computed using (3.1) and (3.4). Additionally, an error requirement, e_{des}, is chosen to determine the accuracy of the approximation.

2. **Initial Generator Selection**: Subsequently, the initial Voronoi generators are chosen based upon the curvature. As with curves in Section 2.4.2, the same three cases arise from inspecting the curvature for surfaces.

(a) **Nonzero constant mean curvature**: Recall from Section 3.1.1 that nonzero constant mean curvature can exist for surfaces that have umbilic and parabolic points. To distinguish between the two, the principal curvatures, $k_1(p)$ and $k_2(p)$, are computed. If both principal curvatures are identical, the surface is umbilic. Otherwise, the surface is parabolic. To approximate either surface, a similar approach to approximating curves with nonzero constant curvature is proposed i.e, the spacing between generators will be inversely proportional to the principal curvature. For convenience, we assume that the coordinate system is chosen such that the x-axis is aligned with the direction along which $k_1(p)$ is changing and the y-axis is aligned with the direction along which $k_2(p)$ is changing. There is no loss of generality in doing so because we can simply modify the coordinate system to align the axes with the direction of curvature change.

i. $k_1(p) \neq k_2(p)$: Assume that $k_1(p)$ is zero and that $k_2(p)$ is nonzero. This assumption suggests that the curvature is invariant in the x direction while it is variant in the y direction. For example, Figure 3.5 shows a parabolic surface whose curvature varies only in the y direction. The first generator, P_1, is chosen to be the centroid (x_c, y_c) of the sampled grid. Since the curvature does not vary in the x direction, the x coordinate for additional linearisation points (ie P_2) will remain x_c. The y coordinate, however, will vary and the distance between each successive point along y is Δ_y defined by

$$\Delta_y = \frac{c}{k_2(p)} \tag{3.7}$$

where c is defined in Section 2.4.2.

ii. $k_1(p) = k_2(p)$: If both principal curvatures are identical, then the surface is umbilic. As with the case of a parabolic surface, the first generator is selected to be the centroid of the sampled grid. The

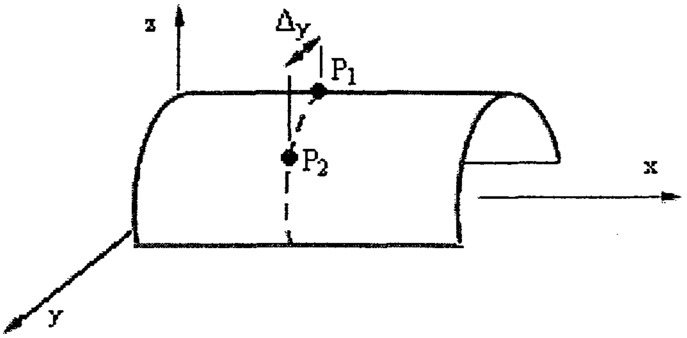


Figure 3.5: Placement of Linearisation Points on a Parabolic Surface

locations of successive points will now vary in the x and y direction according to (3.7). Figure 3.6 shows how successive points are added to the surface. If there is no nonzero constant mean curvature found, the algorithm checks for case 2.

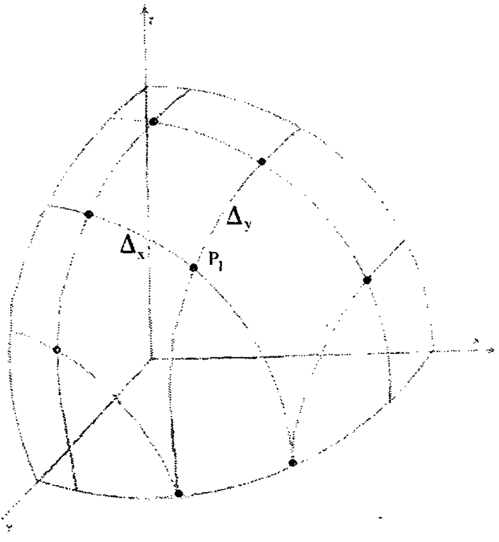


Figure 3.6: Placement of Linearisation Points on an Umbilic Surface

(b) **Zero mean curvature**: If the mean curvature is zero ie, both principal curvatures are zero, then a single linearisation point is placed at the centroid of the sampled grid. Otherwise, if there are points whose mean curvature is zero, then generators are placed at each of those points.

(c) **Neither zero nor nonzero constant mean curvature exists**: If case 1 and case 2 fail, then linearisation points are placed at the locations corresponding to the minimum of the absolute value(s) of the mean curvature.

3. **Linearisation**: The next step is to linearise the nonlinear function about each generator using (2.5).

4. **Voronoi Partition Computation**: To avoid potential numerical problems due to the domain of the sampled nonlinear function being thin (for example, the range of ψ is twice that of the range of v in system (4.14), which we will see later in Section 3.2.1), the Voronoi partition will be normalised. The normalisation of the Voronoi generators is computed by

$$\mathcal{X}_N = \left\{ \frac{X_1}{X_{1_{max}}}, ..., \frac{X_n}{X_{n_{max}}} \right\} \tag{3.8}$$

where $\mathcal{X}_N$ is the set of normalised generators and $X_{i_{max}}$ is the maximum value that a generator can take within the specified range of X_i, $i = 1, ...n$. Finally, following the rules written in (3.5), (3.6) and (3.8), the Voronoi partition is computed using the Multi Parametric Toolbox (MPT) [82].

5. **PWA Approximation Computation**: the PWA approximation can be computed and therefore the maximum approximation error, e_{max}, is obtained using equation (2.17).

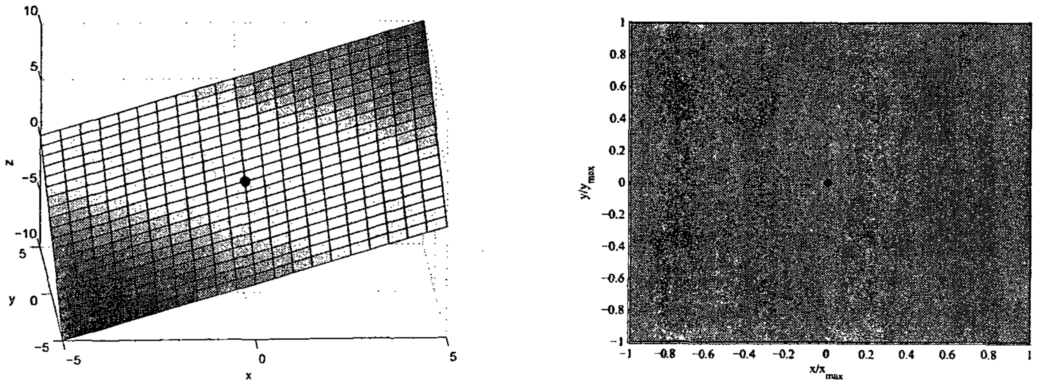
6. **Error Requirement Verification**: The approximation error is then compared to the desired error requirement, e_{des}. If the e_{des} requirement is met, then the PWA model is obtained.

7. **Addition of Operating Points**: If the e_{des} requirement is not met, the approximation procedure repeats itself by including additional operating points, as follows (refer to Figure 2.6 in Section 2.4.2):

(a) **Case 1**: For the case of nonzero constant mean curvature, additional points are added by decreasing c in order to have a finer approximation.

(b) **Case 2 and 3**: For cases 2 and 3, additional points are added where e_{max} occurs until the approximation meets the error requirement.

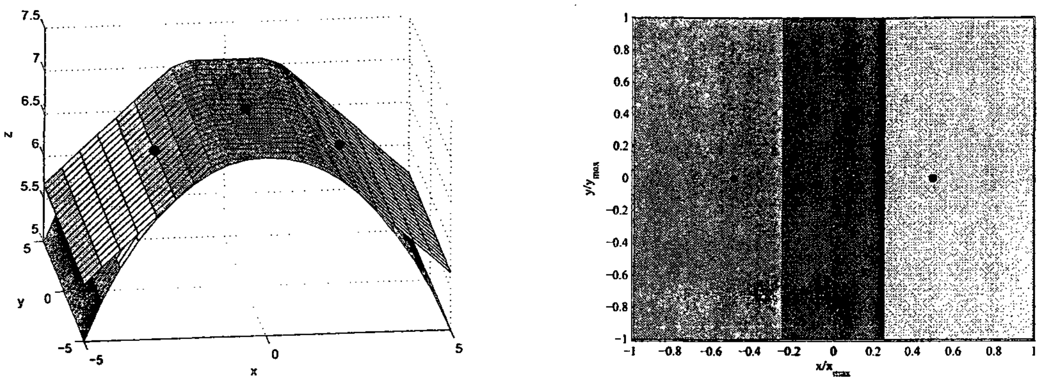
Before applying the proposed methodology to system (4.14), a few simple cases will be used to verify the algorithm. Figures 3.7 and 3.8 depict the results of approximating a plane and a parabolic section, respectively. For the first case, the plane is approximated exactly using one linearisation point at the centroid of the sampled grid. The second case involves the approximation of a parabolic surface. Figure 3.8 depicts the approximation results for $e_{des} = 0.5$ and its accompanying Voronoi partition. As desired, the algorithm has placed additional linearisation points only along the x direction. The partition of the domain can also be seen from the normalised Voronoi partition. With two simple cases verified, the following section will describe an algorithm to compare the SLP and UG methods for surfaces.



(a) Approximation Results.

(b) Voronoi Partition of the Approximation.

Figure 3.7: Approximation of the Plane $z = x + y$.



(a) **Approximation Results**: $e_{des} = 0.5$ and $c = 0.75$

(b) **Voronoi Partition of the Approximation.**

Figure 3.8: Approximation of a Parabolic Section

3.2 Comparison Algorithm

To compare the two approximation methods, the algorithm proposed in Chapter 2 is modified to include surfaces (refer to Figure 2.10):

1. Choose a desired approximation error, e_{des}.

2. (a) SLP: Compute the PWA approximation, the description of the regions and the corresponding maximum error, e_{max}, using (2.5), (3.5)-(3.6) and (2.17), respectively.

   (b) UG: Compute the PWA approximation, the description of the regions and e_{max} for a given UG using the algorithm in section 2.3.

3. (a) SLP: Repeat step two while $e_{max} > e_{des}$ by varying the number and location of points using the algorithm in Section 3.1.2.

   (b) UG: Repeat step 2 while $e_{max} > e_{des}$ using the Toolbox in [28].

4. Compare the number of regions that each method needed to meet the desired error requirement.

3.2.1 Surface Approximation Comparison and Results

In this section, the SLP and the UG methods are applied to the nonlinear aircraft model (4.14) that will later be presented in Chapter 4. Both approximation methods are compared using the algorithm in Section 3.2. We will show that the SLP approximation will require significantly less regions than the UG approximation. The nonlinear function in the model is equations (3.9)-(3.10)

$$f(\psi, v) = u_0 \sin(\psi + \beta) + v \cos(\psi + \beta) \tag{3.9}$$

$$\beta = \arctan \frac{v}{u_0} \tag{3.10}$$

To evaluate $f(\psi, v)$, three physical parameters are required: u_0, ψ and v. The forward velocity, u_0, is assumed to be constant while ψ and v are allowed to vary as in [23] and are shown in Table 3.1. The following results are obtained and shown in Table 3.2 for two desired error requirements: $e_{des} = 0.50$ and 0.25. Notice that the Voronoi partition in Figure 3.9 is normalised as mentioned in the approximation algorithm in Section 3.1.2. The main result of this comparison is that the SLP approximation uses a significantly smaller number of regions to approximate $f(\psi, v)$ than the UG

Table 3.1: Physical parameters used to approximate $f(\psi, v)$

Parameter	Value/Range
$u_0(m/s)$	0.7
$\psi(rad)$	$\frac{-\pi}{2} < \psi < \frac{\pi}{2}$
$v(m/s)$	$-0.8 < v < 0.8$

Table 3.2: Surface Approximation Results

	SLP Approximation		UG Approximation	
e_{des}	e_{max}	Number of Regions	e_{max}	Number of Regions
0.50	**0.27**	**9**	**0.31**	**18**
0.25	**0.24**	**13**	**0.201**	**50**

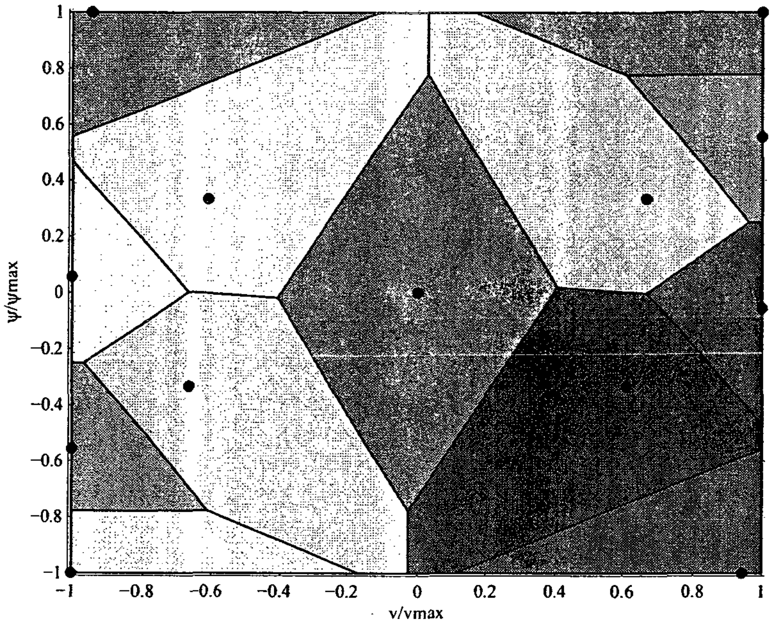


Figure 3.9: Voronoi Partition for the PWA approximation with $e_{des} = 0.25$

approximation. The SLP approximation required fifty percent less regions than the UG approximation for $e_{des} = 0.50$. Moreover, only thirteen regions were required in the SLP approximation to achieve $e_{des} = 0.25$ while the UG approximation needed 50 regions. These results are extremely useful because if one were to control (2.2) using the state feedback law (3.11)

$$u = \bar{K}_i \bar{x}, \text{ for } x \in \mathcal{R}_i \tag{3.11}$$

where $\bar{K}_i = \begin{bmatrix} K_i & m_i \end{bmatrix}$ as done in [23], one would need a controller for each region. The drawback to the Voronoi approximation is due to the potential discontinuities at the boundaries of each region as seen in Figure 3.10. These discontinuities can lead to a non-smooth control signal. In order to overcome this problem, a condition to ensure control input continuity will be used in the controller design. With the SLP approximation method successfully applied to surfaces, the focus can now be shifted to applying the newly proposed method to various case studies, which is the topic of the next chapter.

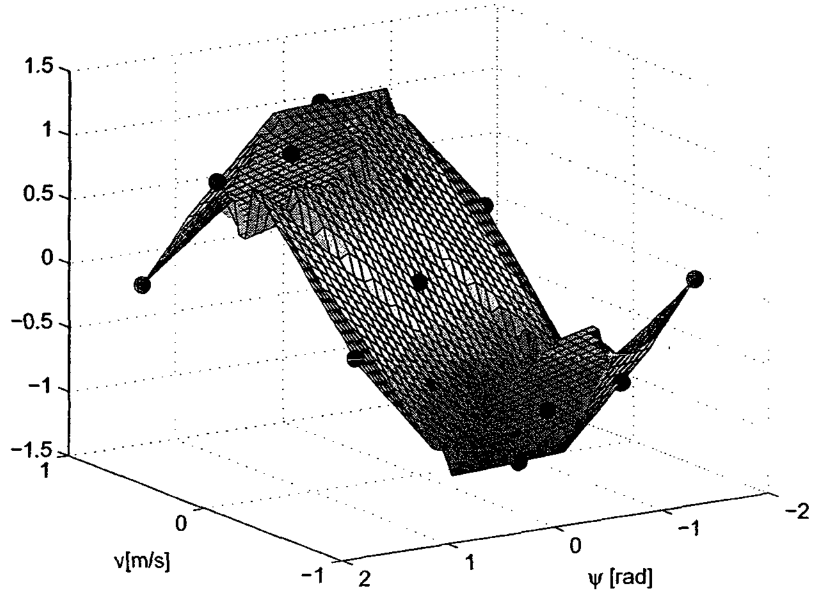


Figure 3.10: Voronoi PWA Approximation of $f(\psi, v)$ for $e_{des} = 0.25$

3.3 Summary

In summary, this chapter extends the SLP approximation method proposed in Chapter 2 to smooth surfaces using Voronoi partitions to partition the domain of the nonlinearity. The SLP approximation method is then validated for several examples in Section 3.1.2. Subsequently, the UG and SLP methods are compared in Section 3.2.1 using the algorithm proposed in Section 2.5. The main result of this comparison is that the number of regions in the SLP approximation is significantly less than the number of regions for the UG approximation.

Chapter 4

Case Studies

4.1 Introduction

The purpose of this chapter is to apply the SLP approximation methodology presented in Chapter 3 to case studies. The first case study is the modelling of a mechanical system with a nonlinear spring and damper. The second and third case studies involve the modelling and control of a rotorcraft UAV and MAV, respectively.

4.2 Case Study 1: Modelling of a Nonlinear Spring and Damper Mechanical System

The first case study is a mass-spring-damper mechanical system with nonlinear spring and damper elements [16]. Assuming a constant unity mass, the dynamics that describe this mechanical system can be written as follows:

$$\dot{x_1} = x_2 \tag{4.1}$$

$$\dot{x_2} = -b(x_2)x_2 - k(x_1)x_1 \tag{4.2}$$

where $b(x_2) = |x_2|$ and $k(x_1) = (1 + x_1^2)$ are the nonlinear damping and spring coefficients, respectively, and $|\cdot|$ is the absolute value. Equations (4.1)-(4.2) can be written in the state space form of (2.1) as

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ f(x_1, x_2) \end{bmatrix} \tag{4.3}$$

$$f(x_1, x_2) = -|x_2|x_2 - x_1^3 \tag{4.4}$$

Note that the second derivative of $f(x_1, x_2)$ is not continuous for $x_2 = 0$ and therefore the surface approximation algorithm developed in Section 3.1.2 cannot be immediately applied. To make the second derivative of $f(x_1, x_2)$ continuous (which makes the curvature continuous), the absolute value function in (4.4) is replaced with the following function [10]:

$$|x_2| \approx \frac{2x_2}{\pi} \arctan(ax_2) \tag{4.5}$$

where $a \gg 0$ is a parameter that tunes the smoothness of (4.5). The algorithm in Section 3.1.2 is now applied to system (4.3) with $e_{des} = 2$, $a = 100$ and $-2 \leq x_1 \leq 2$, $-2 \leq x_2 \leq 2$ as in [16]. The resulting approximation of $f(x_1, x_2)$ and the Voronoi partition can be seen in Figure 4.1. A total of 20 Voronoi cells are used in the SLP approximation while a uniform grid consisting of 32 simplicial regions is considered in [16]. Figure 4.2 compares the open loop system trajectories of the nonlinear, the SLP and the grid PWA systems for the initial conditions $x_0 = \begin{bmatrix} 1.5 & 1 \end{bmatrix}^T$ (Top) and $\begin{bmatrix} -1.5 & 2 \end{bmatrix}^T$ (Bottom) over a time period of 35 seconds. The main result is that the open loop response of the SLP PWA system follows the nonlinear trend more closely than the grid PWA system. This result is easily seen when the system begins to oscillate to its equilibrium point $\begin{bmatrix} 0 & 0 \end{bmatrix}^T$. The grid approximation reaches the equilibrium point after 10 seconds while the SLP approximation continues to oscillate with the nonlinear system. The advantage that the UG method has in this example over the SLP method is that it does not require that the absolute function be approximated by (4.5).

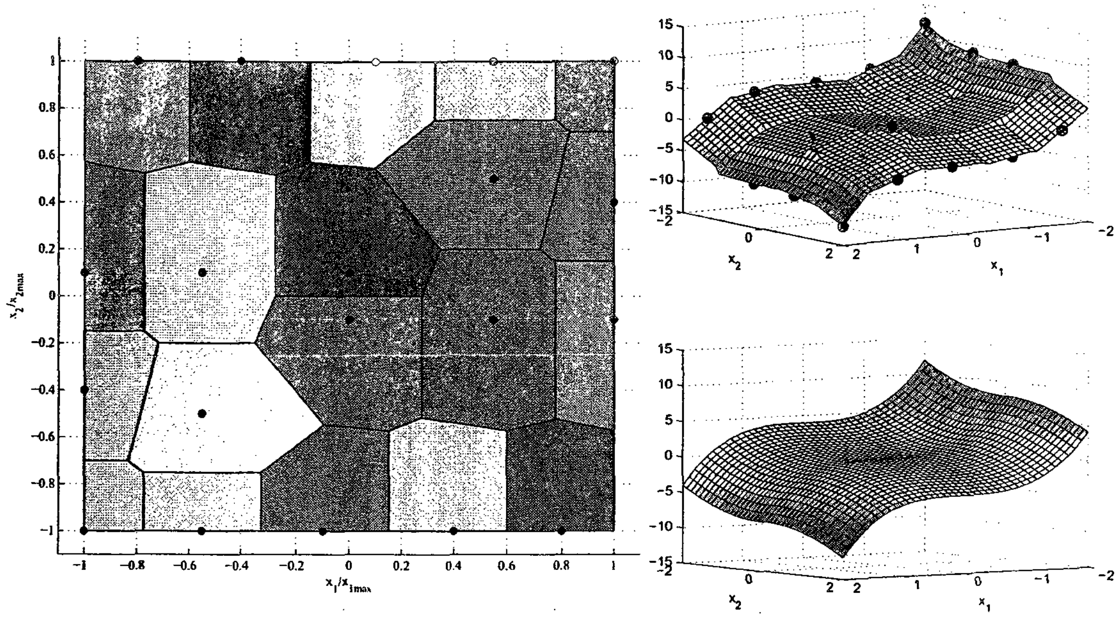


Figure 4.1: Voronoi Partition and PWA Approximation of $f(x_1, x_2)$.

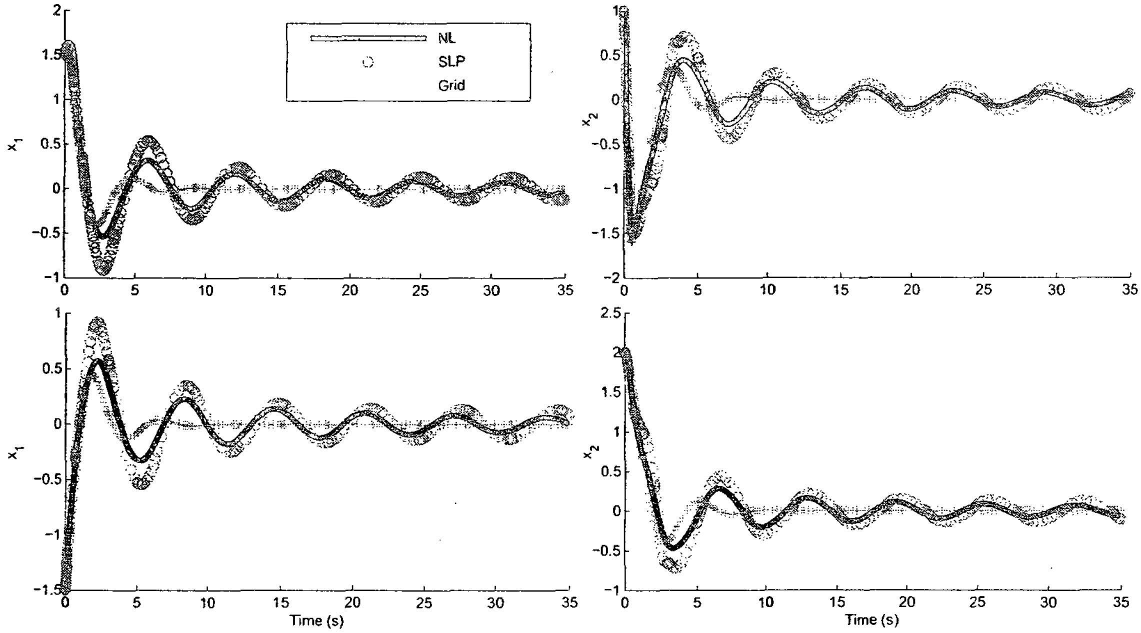


Figure 4.2: Nonlinear, SLP and Grid Open Loop Responses.

4.3 Case Studies 2 and 3: Modelling and Control of UAV and MAV Helicopters

The second and third case studies are the modelling and control of rotorcraft UAV and MAV. To proceed, Section 4.3.1 will present the nonlinear equations of motion for an aircraft. Next, a simplified nonlinear lateral model of a rotorcraft is obtained and the lateral model is linearised about an operating point. Subsequently, a PWA controller design methodology will be reviewed in Section 4.3.2 for systems of the form of (2.1). Section 4.3.3 will present the approximation and simulation results for the rotorcraft UAV and MAV. As a first step, the aircraft model (4.14) will be validated through comparison of simulation results using the software in [9]. Next the results are shown for the rotorcraft UAV followed by the rotorcraft MAV.

4.3.1 Aircraft Modelling

Equations of Motion

In this section, a simplified nonlinear rotorcraft model that was derived in [9] is considered. The motion of the rotorcraft will be restricted to the lateral plane as seen in Figures 4.3 and 4.4 for the UAV and MAV, respectively. Therefore, the equations of motion can be written as follows:

$$\begin{aligned} \dot{u} &= vr + \frac{X}{m} \\ \dot{v} &= -ur + \frac{Y}{m} \\ \dot{r} &= \frac{N}{I_{zz}} \\ \dot{\psi} &= r \end{aligned} \tag{4.6}$$

where u and v are translational velocites, r is the yaw rate, ψ is the yaw angle, m is the rotorcraft mass and I_{zz} is the mass moment of inertia. The resulting 3 DOF model is highly nonlinear. The difficulty present in the model arises when one needs to evaluate the forces (X, Y) and moment (N). These unknowns are highly nonlinear functions that depend upon the state variables and control inputs. Moreover, these forces and moment must be evaluated over a complex arrangement of five interacting subsystems: main rotor, tail rotor, horizontal stabiliser, vertical fin and fuselage. The goal of this thesis is not to address the issue of evaluating these forces and moment. The reader is referred to [9] for a detailed description of each subsystem, including their interactions and the forces and moments they generate. Instead we will assume that these forces and moment are known.

Control Objective and Simplified Model

The control objective in this thesis is to design a controller to drive the rotorcraft to follow the straight line $y = 0$ at a constant forward velocity u_0. In other words, the following variable is set to zero: $\dot{u} = 0$. Therefore, system (4.6) becomes:

$$\dot{v} = -u_0 r + \frac{Y}{m} \tag{4.7}$$

$$\dot{r} = \frac{N}{I_{zz}} \tag{4.8}$$

$$\dot{\psi} = r \tag{4.9}$$

- UAV Model

For the UAV, the model complexity is reduced further by simplifying the force Y and the moment N. Equations (4.7)–(4.9) now become (refer to [9] for all the relevant details):

$$\dot{v} = -u_0 r - \frac{k_v}{m} v \tag{4.10}$$

$$\dot{r} = -\frac{k_r}{I_{zz}} r + \frac{Q_m}{I_{zz}} - \frac{T_t l_t}{I_{zz}} \tag{4.11}$$

$$\dot{\psi} = r \tag{4.12}$$

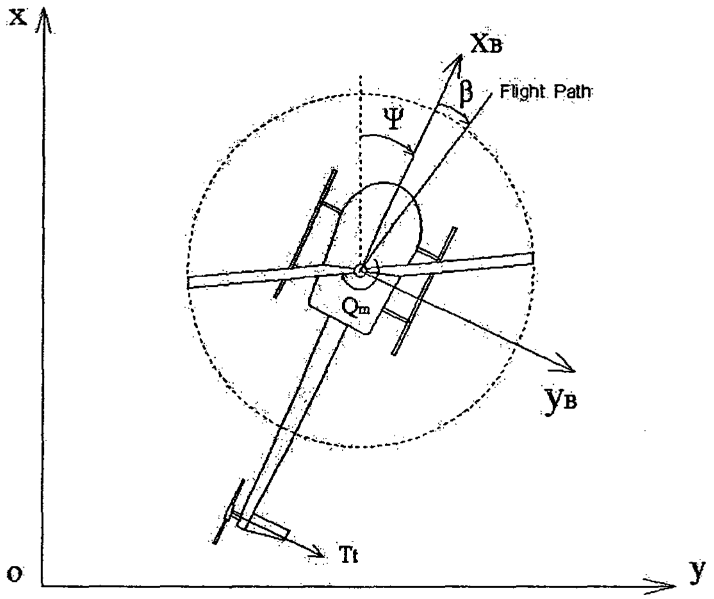


Figure 4.3: Lateral Rotorcraft UAV Model([9])

where k_v and k_r are slide damping and yaw damping coefficients, respectively, Q_m is the torque produced by the main rotor, l_t is the distance from the center of gravity to the hub of the tail rotor and $T_t = k_t\delta_{ped}$ is the tail rotor thrust with k_t and δ_{ped} being the tail blade constant and the tail rotor pedal control input, respectively. To meet the control objective, the velocity equation on y_b is included as

$$\dot{y} = u_0 \sin(\psi + \beta) + v\cos(\psi + \beta) \tag{4.13}$$

where β is the sideslip angle. Combining (4.10)−(4.13) gives the lateral rotorcraft UAV model written in the form of (2.1) as

$$\begin{bmatrix} \dot{\psi} \\ \dot{r} \\ \dot{v} \\ \dot{y} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -\frac{k_r}{I_{zz}} & 0 & 0 \\ 0 & -u_0 & -\frac{k_v}{m} & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \psi \\ r \\ v \\ y \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{Q_m}{I_{zz}} \\ 0 \\ f(\psi, v) \end{bmatrix} + \begin{bmatrix} 0 \\ -\frac{k_t l_t}{I_{zz}} \\ 0 \\ 0 \end{bmatrix} u \tag{4.14}$$

$$f(\psi, v) = u_0 \sin(\psi + \beta) + v \cos(\psi + \beta) \tag{4.15}$$

$$\beta = \arctan \frac{v}{u_0} \tag{4.16}$$

- MAV Model

The Micro Mosquito MAV (as seen in Figure 4.4) has a pair of counter-rotating main rotors that controls hovering and turning of the helicopter. The tail rotor controls the pitch of the helicopter and hence the forward (and reverse) motions. This is different from the UAV in that the tail rotor was used to control the hovering and turning. Furthermore, there is no vertical fin and therefore, the total yawing moment, N, is produced by the main and tail rotors. For the purpose of this thesis, the pitch angle of the MAV helicopter is assumed to be small. The model can then be approximated

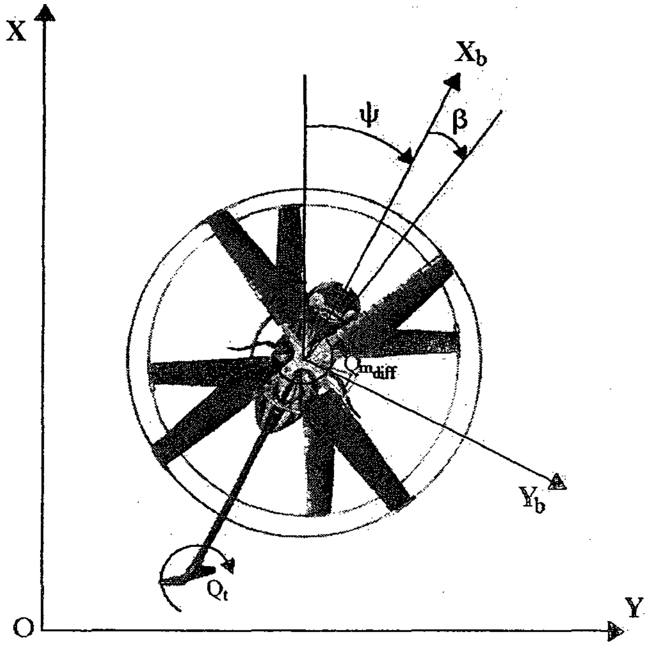


Figure 4.4: Lateral Rotorcraft MAV Model.

by (4.17)-(4.19)

$$\begin{bmatrix} \dot{\psi} \\ \dot{r} \\ \dot{v} \\ \dot{y} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & -u_0 & -\frac{k_v}{m} & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \psi \\ r \\ v \\ y \end{bmatrix} + \begin{bmatrix} 0 \\ -\frac{Q_t}{I_{zz}} \\ 0 \\ f(\psi, v) \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{I_{zz}} \\ 0 \\ 0 \end{bmatrix} Q_{m_{diff}} \tag{4.17}$$

$$f(\psi, v) = u_0 \sin(\psi + \beta) + v \cos(\psi + \beta) \tag{4.18}$$

$$\beta = \arctan \frac{v}{u_0} \tag{4.19}$$

where Q_t is the torque produced by the tail rotor and $Q_{m_{diff}} = k_m \delta_m$ is the differential main rotor torque with the main rotor torque constant k_m and the differential main rotor control input δ_m.

The next section will focus on linearising systems (4.14) and (4.17) about the equilibrium condition $x_0 = \begin{bmatrix} \psi_0 & r_0 & v_0 & y_0 \end{bmatrix}^T$.

Linearised Lateral Model

A linear system of differential equations can be described in state space form by the following expression:

$$\dot{x} = Ax + Bu \tag{4.20}$$

where $x \in \mathbb{R}^n$ is the state vector, $u \in \mathbb{R}^m$ is the control input, A and B are matrices $\in \mathbb{R}^{nxn}$ and $\mathbb{R}^{n \times m}$ respectively. To obtain the A and B matrices, the nonlinear system (4.20) is linearised about the operating condition x_0. The Jacobian matrix, J, is a matrix whose elements are all first-order partial derivatives (evaluated at the equilibrium point) of a multivariate[3] function. Suppose that there are n multivariate functions $(y_1 ... y_n)$ which are functions of m variables $(x_1 ... x_m)$. The corresponding

[3] involves more than one variable

$n \times m$ Jacobian matrix can be written as

$$J = \begin{bmatrix} \frac{\partial y_1}{\partial x_1} & \cdots & \frac{\partial y_1}{\partial x_m} \\ \vdots & \ddots & \vdots \\ \frac{\partial y_n}{\partial x_1} & \cdots & \frac{\partial y_n}{\partial x_m} \end{bmatrix} \tag{4.21}$$

For the case of the rotorcraft, $x_0 = \begin{bmatrix} \psi_0 & r_0 & v_0 & y_0 \end{bmatrix}^T$. To obtain the A matrix, a 4×4 Jacobian matrix must be determined because there are 4 multivariate functions (one for each state equation) that involves 4 variables (one for each state variable). However, the only nonlinear multivariate function is $f(\psi, v)$ while all other elements are linear. Therefore, only the variables involved in $f(\psi, v)$ are taken into account. Moreover, the B matrix is constant and the Jacobian procedure does not need to be applied. The linearised system of (4.14) can now be represented by the following expression:

$$\begin{bmatrix} \dot{\psi} \\ \dot{r} \\ \dot{v} \\ \dot{y} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -\frac{k_r}{I_{zz}} & 0 & 0 \\ 0 & -u_0 & -\frac{k_v}{m} & 0 \\ \left.\frac{\partial \dot{y}}{\partial \psi}\right|_{x_0} & 0 & \left.\frac{\partial \dot{y}}{\partial v}\right|_{x_0} & 0 \end{bmatrix} \begin{bmatrix} \psi \\ r \\ v \\ y \end{bmatrix} + \begin{bmatrix} 0 \\ -\frac{k_t l_t}{I_{zz}} \\ 0 \\ 0 \end{bmatrix} u \tag{4.22}$$

The only difference between the linearised system of (4.17) and (4.22) is that the elements in the second row are all zero. The next section will review the control design methodology proposed in [21, 28].

4.3.2 Piecewise Affine Controller Design

This section will focus on the development of a PWA controller to stabilise the rotorcraft models (4.14) and (4.17) that were previously derived. In order to meet the control objective, a globally quadratic control Lyapunov function will be searched for. Next, the set of constraints imposed on the controller design is presented. Finally, the control synthesis problem is formulated and solved. This section is a review of the work in [21, 28].

Controller Design Methodology

The objective is to design a PWA controller to enable the rotorcraft to follow the straight line $y = 0$. In order to do this, the system's equilibrium point $x_{cl} = \begin{bmatrix} 0 & 0 & 0 & 0 \end{bmatrix}^T$ will be stabilised using the state feedback control law (3.11). Substituting (3.11) into (2.2) leads to

$$\dot{\bar{x}} = (\bar{A}_i + \bar{B}_i\bar{K}_i)\bar{x} \tag{4.23}$$

where $\bar{A}_i = \begin{bmatrix} A_i & a_i \\ 0 & 0 \end{bmatrix}$ and $\bar{B}_i = \begin{bmatrix} B_i \\ 0 \end{bmatrix}$. To design the controller, a Lyapunov based method is used. The candidate Lyapunov function to be searched for will be globally quadratic defined by

$$V(x) = \bar{x}^T\bar{P}\bar{x} \tag{4.24}$$

where $\bar{P} = \begin{bmatrix} P & -Px_{cl} \\ -x_{cl}^TP^T & r \end{bmatrix}$ and $P = P^T > 0$. The following constraints are imposed in order to design the stabilising controller[21, 28]:

1. **Positive definitiveness of the candidate Lyapunov function**: The candidate Lyapunov function is positive definite if

$$V(x) > 0, \text{ for } x \neq x_{cl} \tag{4.25}$$

To satisfy the inequality, $\bar{P}$ must be positive definite. Therefore,

$$\bar{P} > 0 \tag{4.26}$$

2. **Decay of the candidate Lyapunov function over time**: The candidate Lyapunov function is decreasing with time if

$$\dot{V}(x) < -\alpha V(x) \tag{4.27}$$

where $\alpha \geq 0$ is the decay rate of the Lyapunov function. Sufficient conditions for satisfying inequality (4.27) for each region $\mathcal{R}_i$ can be shown using the polytopic description of the cells (2.3) and the $\mathcal{S}$-procedure [83] to determine the existence of matrices Λ_i with nonnegative entries that satisfies

$$\bar{P}\left(\bar{A}_i + \bar{B}_i\bar{K}_i\right) + \left(\bar{A}_i + \bar{B}_i\bar{K}_i\right)^T \bar{P} + \alpha\bar{P} + \bar{E}_i^T\Lambda_i\bar{E}_i < 0 \qquad (4.28)$$

3. **Continuity of the control input**: Recall from Section 3.2.1 that the SLP approximation can be discontinuous. Therefore, to ensure that the control input is continuous across region boundaries, we need to have $u_i(x) = u_j(x)$ for $x \in \bar{\mathcal{R}}_i \cap \bar{\mathcal{R}}_j$ or in other words

$$\bar{K}_i\bar{x} = \bar{K}_j\bar{x} \text{ for } \bar{x} = \bar{F}_{ij}\bar{s} \qquad (4.29)$$

Therefore, the constraint to satisfy the continuity of the control signal is

$$(\bar{K}_i - \bar{K}_j)\bar{F}_{ij} = 0, \text{ for } j = \mathcal{N}_i \qquad (4.30)$$

The PWA controller design used in this thesis consists of the following steps [5]:

1. **Linear controller design**: The first step consists of designing a local linear controller for region $\mathcal{R}_{i^*}$ containing the equilibrium point i^* to meet a stability or performance requirement. To proceed, consider the dynamics of the system within this region to be given by

$$\dot{x} = A_{i^*}x + a_{i^*} + B_{i^*}u, \text{ for } x \in \mathcal{R}_{i^*} \qquad (4.31)$$

Next we assume that there exists a vector m_{i^*} such that

$$A_{i^*}x_{cl} + a_{i^*} + B_{i^*}m_{i^*} = 0 \qquad (4.32)$$

With the control input given to be

$$u = K_{i^*}x + m_{i^*} \qquad (4.33)$$

Hence, the closed loop dynamics are obtained by substituting (4.33) into (4.31).

$$\dot{x} = (A_{i^*} + B_{i^*} K_{i^*})\, x \tag{4.34}$$

The feedback matrix K_{i^*} can now be designed using linear control methodologies. In this thesis, a LQR controller controller is designed for the UAV and the MAV using the following weighting parameters

$$Q_{UAV} = \begin{bmatrix} 50 & 0 & 0 & 0 \\ 0 & 500 & 0 & 0 \\ 0 & 0 & 500 & 0 \\ 0 & 0 & 0 & 50 \end{bmatrix} \tag{4.35}$$

$$R_{UAV} = 0.5 \tag{4.36}$$

$$Q_{MAV} = \begin{bmatrix} 0.91 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 2421 & 0 \\ 0 & 0 & 0 & 0.04 \end{bmatrix} \tag{4.37}$$

$$R_{MAV} = 1 \tag{4.38}$$

2. **PWA Problem formulation**: Given xcl, α, and fixing K_{i^*} and m_{i^*}, the problem can be formulated as follows

$$\begin{aligned} &\text{Find } \bar{P} \\ &\quad \text{s.t. } (4.26), (4.28), (4.30) \\ &\qquad P = P^T > 0, \Lambda_i \succ 0, \\ &\qquad -\bar{K}_{Lim} \prec \bar{K}_i \prec \bar{K}_{Lim} \\ &\qquad \text{for } i \in \mathcal{I} = \{1, ..., M\} \end{aligned} \tag{4.39}$$

where $\bar{K}_{Lim}$ is a limit imposed on the controller gains and $\prec$ and $\succ$ are component-wise inequalities.

The set of constraints (4.39) in the synthesis problem contains a set of Bilinear Matrix Inequalities (BMIs), which are nonconvex and difficult to solve. To solve the current synthesis problem, a general purpose, non-commercial software called PENBMI [84] is used in conjunction with YALMIP [85].

If a solution exists to problem (4.39), the following results can be established to prove stability of the closed-loop system [5].

Theorem 4.3.1 *Assume that the Lyapunov function is defined in $\mathcal{X} \subseteq \mathbb{R}^n$. If there exists a feasible solution to problem (4.39), then the PWA approximate closed-loop system is locally exponentially stable inside any subset of the largest level set of the control Lyapunov function (4.24) that is contained in $\mathcal{X}$.*

Proof: The reader is referred to [5] for details.

Furthermore, if the approximation error between the PWA closed-loop system and the original closed-loop system is small enough, then it follows that the original closed-loop system is locally exponentially stable inside any subset of the largest level set of the control Lyapunov function (4.24) that is contained in $\mathcal{X}$ [5]. The resulting PWA feedback control law for the UAV is presented in the Appendix at the end of this Chapter and the global control Lyapunov functions proving stability of the closed-loop UAV and MAV systems are shown in Figures 4.5 and 4.6. It should be noted that the PWA feedback control law for the UAV using the SLP method required 20 seconds to compute compared to an hour of computation time in [23] for the UG method.

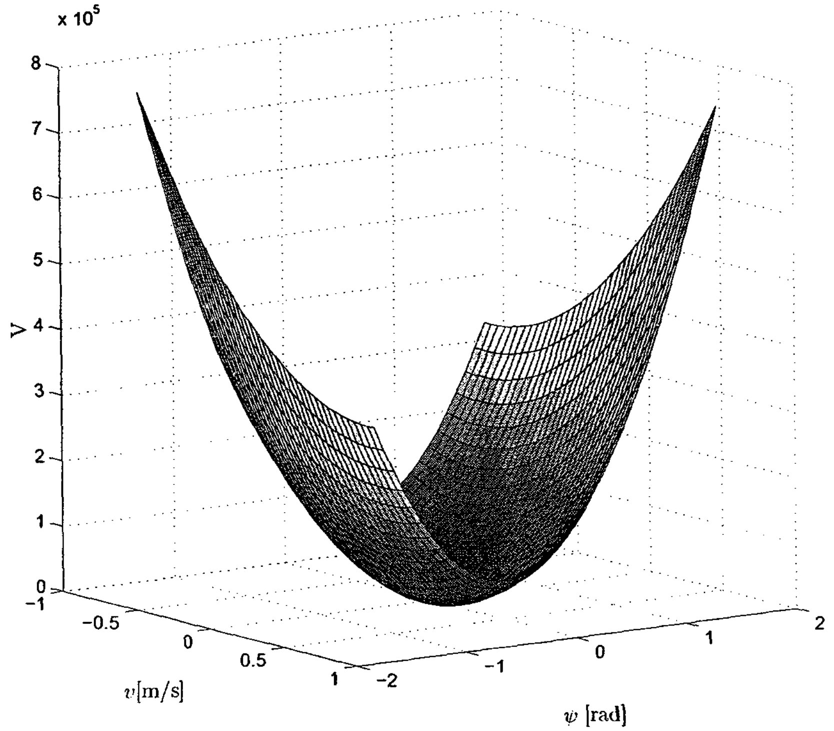


Figure 4.5: Control Lyapunov Function for the UAV.

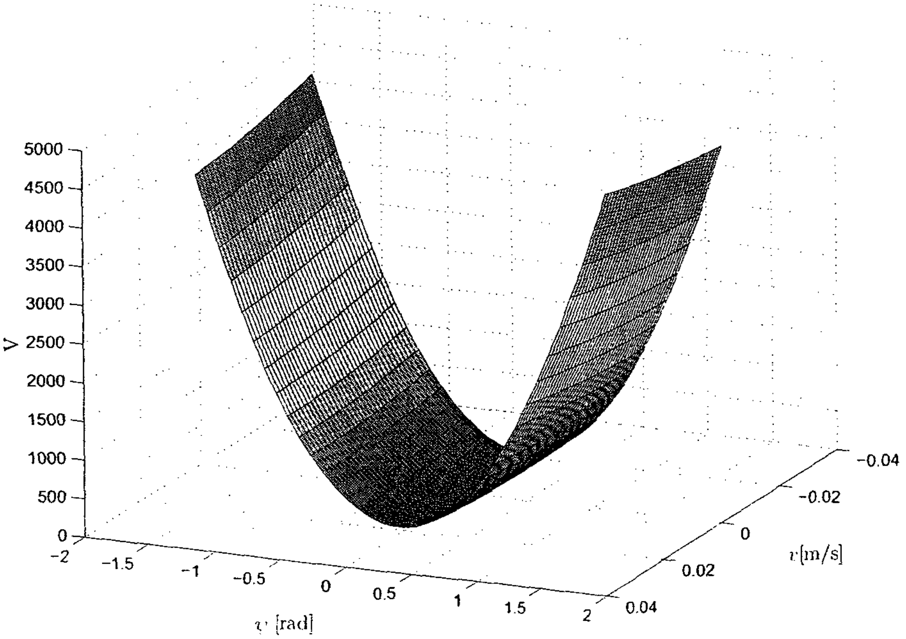


Figure 4.6: Control Lyapunov Function for the MAV.

4.3.3 Results

This section will present the PWA approximation and simulation results for case studies 2 and 3. First, the simplified 3 DOF rotorcraft model (4.14) developed in Section 4.3.1 will be validated through simulation against the software in [9]. In case study 2, the SLP and UG approximation results are revisited from Section 3.2.1. Subsequently, the closed-loop simulation results for case study 2 validate the feedback controller laws designed using the methodology reviewed in Section 4.3.2. In the third case study, a SLP PWA approximation for the MAV model (4.17) is obtained and compared to a UG approximation. Finally, a PWA controller is designed for the MAV to follow a desired trajectory.

Software Validation

This section presents the software validation of the open-loop system (4.14) by comparing with simulation results from [9]. The validation will be performed for three different cases. For the first case, the helicopter will be initially be flying at its equilibrium condition with no control input applied. The second case will verify the response of the rotorcraft subject to an initial flight condition other than the equilibrium condition without any control input. Finally, the third case will use the same initial conditions as the first case but with a doublet[4] control input. The model parameters used for the simulations are tabulated in 4.1 and the initial conditions used for the three cases are as follows:

Case One: $x_0 = \begin{bmatrix} 0 & 0 & 0 & 0 \end{bmatrix}^T$ and control input $u = 0$.

Case Two: $x_0 = \begin{bmatrix} \frac{\pi}{3} & 0 & 0 & 5 \end{bmatrix}^T$ and control input $u = 0$.

Case Three: $x_0 = \begin{bmatrix} 0 & 0 & 0 & 0 \end{bmatrix}^T$ and a doublet control input of amplitude 50. Figures 4.7, 4.8 and 4.9 compare the dynamical responses of the nonlinear open-loop system (4.14) and the one found in [9]. In each of the cases, it can clearly be seen

[4] A doublet is characterised by the input being a one period sinusoid.

that each responses is identical, thus validating the software.

Table 4.1: UAV Helicopter Simulation Parameters

Parameter	Value	Unit
I_{zz}	0.01	$kg - m^2$
m	1	kg
l_t	0.5	m
k_t	0.01	$-$
k_r	0.1	$-$
k_v	1	$-$
Q_m	1	$N - m$
u_0	0.7	m/s

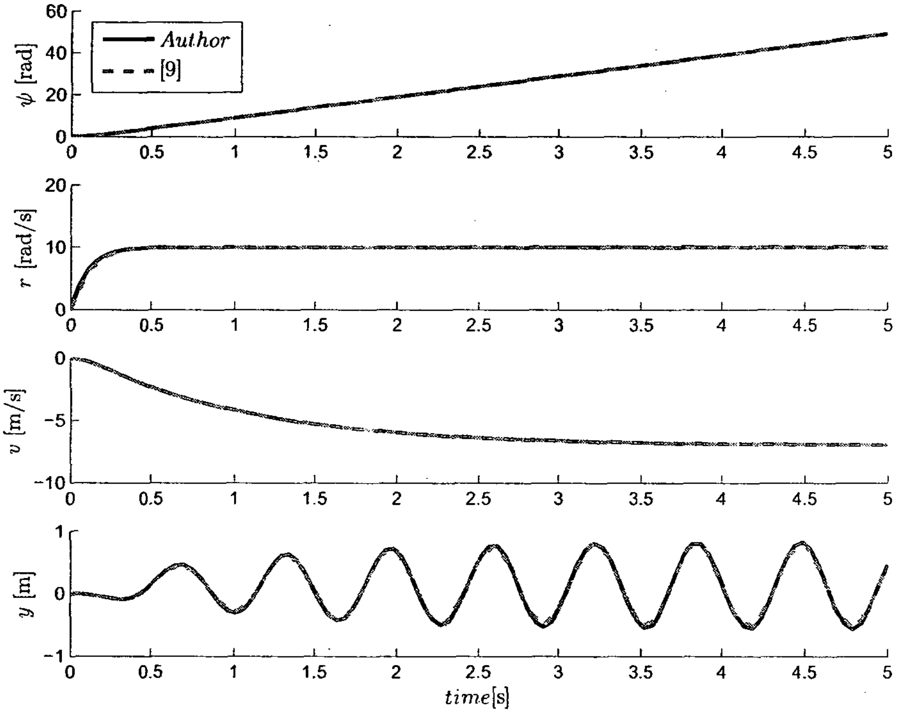


Figure 4.7: Time histories of case one.

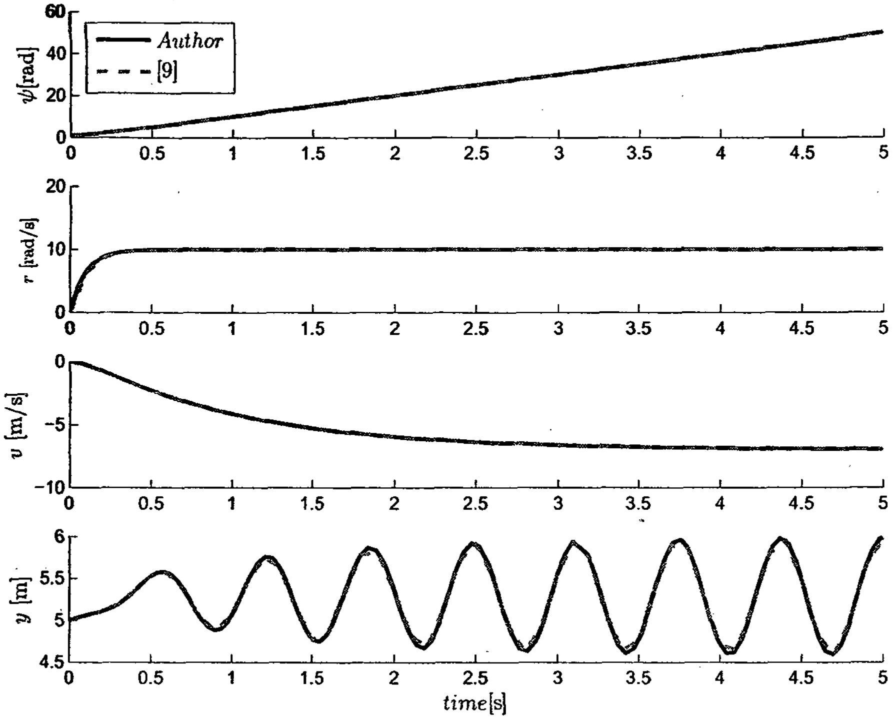


Figure 4.8: Time histories of case two.

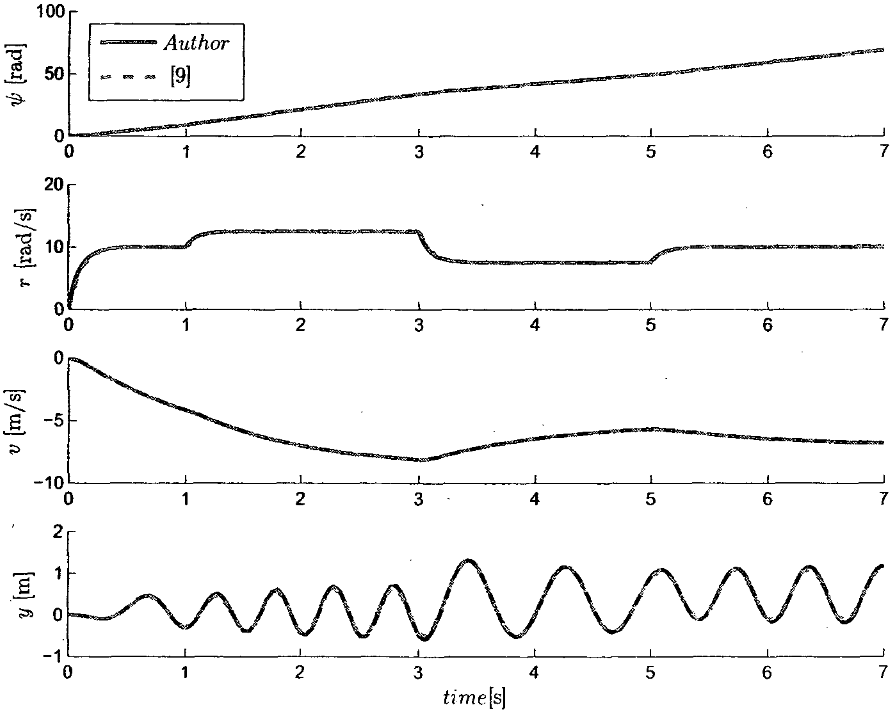


Figure 4.9: Time histories of case three.

Case Study 2: 3 DOF Rotorcraft UAV

Recall that in Chapter 3, a SLP and a UG approximation was obtained for the UAV model (4.14) that was derived earlier in Section 4.3.1 (the full PWA model of the UAV can be found in the Appendix at the end of this chapter). Both approximation methods were then compared and it was shown that the SLP approximation required significantly less regions than the UG approximation. In this section, the developed SLP PWA controller is applied directly to (4.14) as seen in the Simulink block diagram shown in Figure 4.10. Figure 4.11 shows the time histories for the nonlinear closed loop models obtained with the proposed methodology in this thesis and from previous work [23]. It can easily be seen that both PWA controllers are able to stabilise the nonlinear system. However, as mentioned previously in Section 3.2.1, less regions imply less controllers. In this case, the SLP PWA controller consists of thirteen controllers as opposed to thirty controllers in [23]. The $x - y$ trajectory of the closed loop system shown in Figure 4.12 confirms that the rotorcraft meets the control objective by following the straight line $y = 0$. Given a different set of other initial conditions, Figure 4.13 shows that the PWA controller is still able to stabilise the nonlinear system. However, if one were to apply the linear controller designed for the region containing the equilibrium point to the nonlinear system, the control objective is no longer achieved. Figures 4.15 and 4.16 show that the nonlinear system in feedback with the linear controller is not stabilised unlike with the PWA controller.

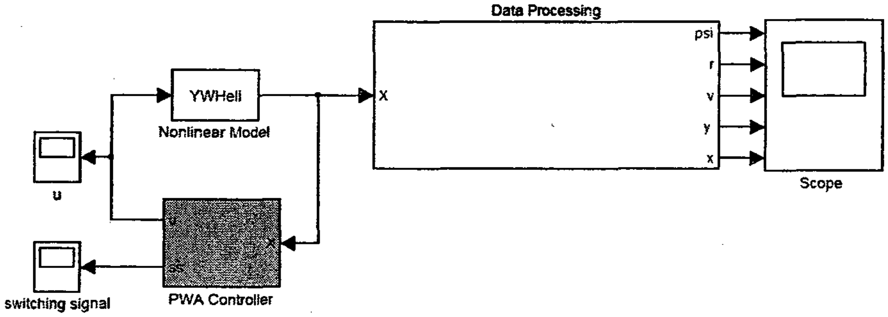


Figure 4.10: UAV Simulink Model Layout

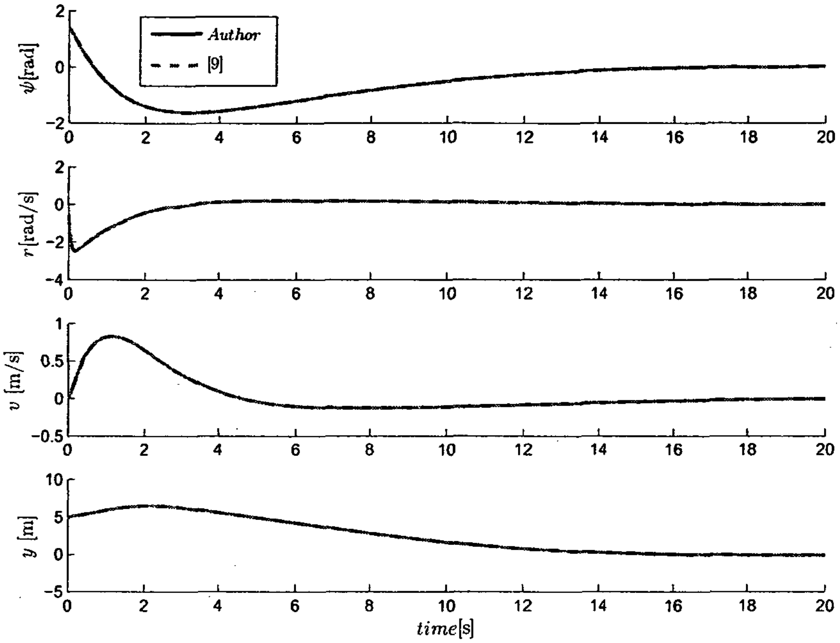


Figure 4.11: Time histories of the SLP and UG based nonlinear closed-loop systems with initial conditions $\psi_0 = \frac{\pi}{3}$, $y_0 = 5$.

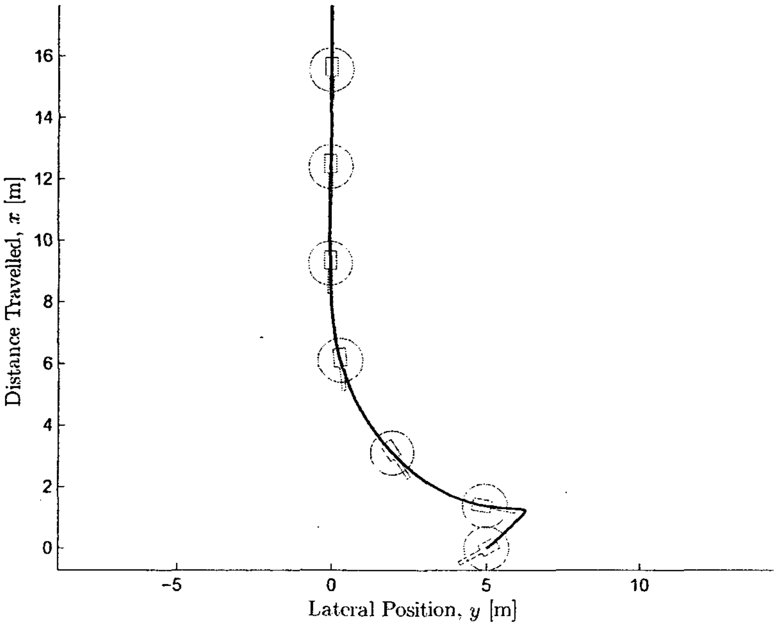


Figure 4.12: x-y trajectory of the Voronoi closed-loop system with initial conditions $\psi_0 = \frac{\pi}{3}$, $y_0 = 5$

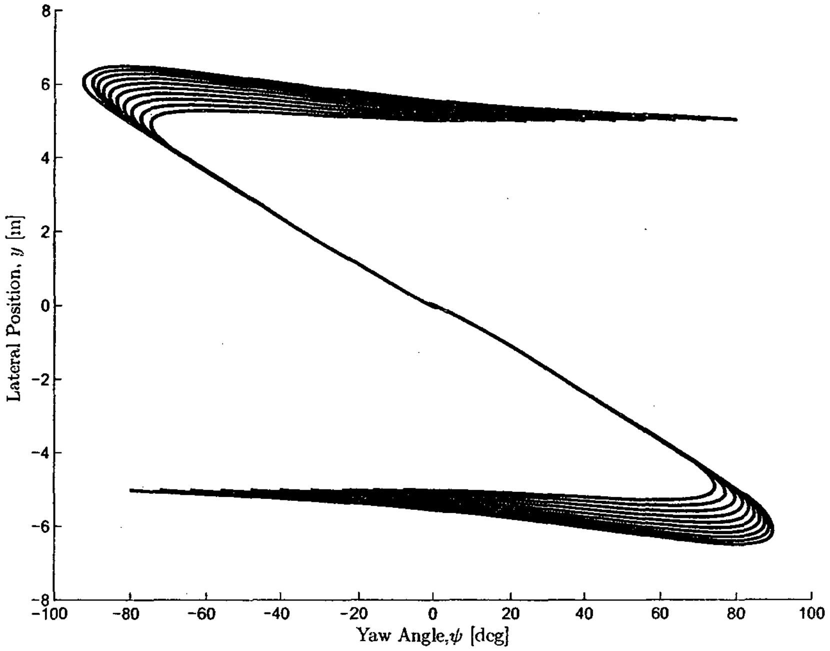


Figure 4.13: Closed-loop system trajectories with different initial conditions.

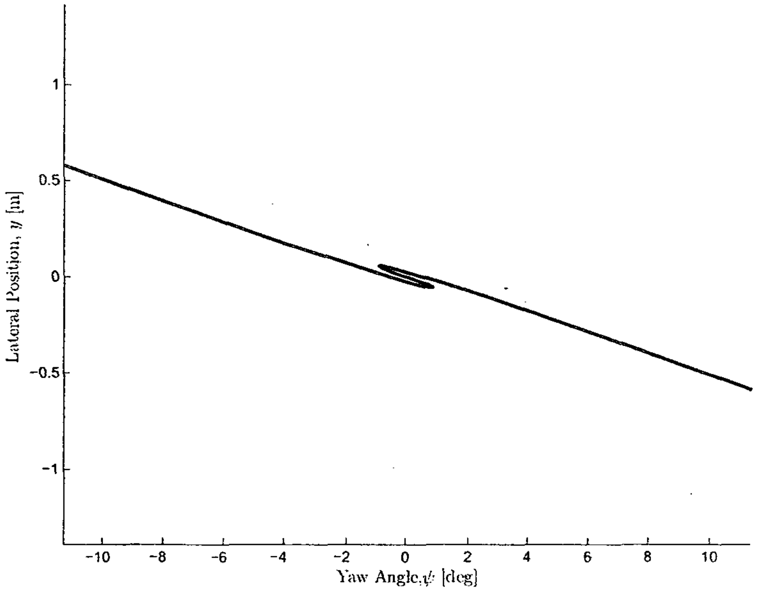


Figure 4.14: Close up of the origin of Figure 4.13

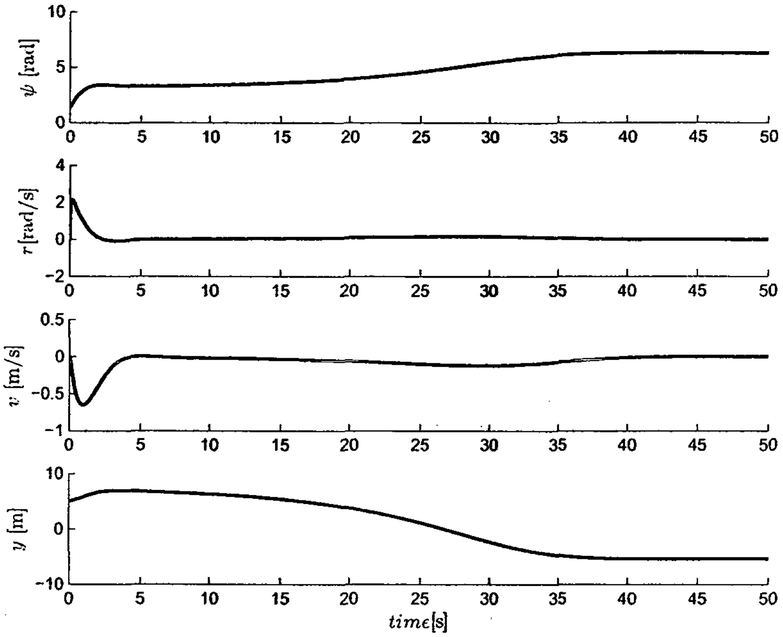


Figure 4.15: Time histories of the closed-loop system with a linear controller for initial conditions $\psi_0 = \frac{\pi}{3}$, $y_0 = 5$.

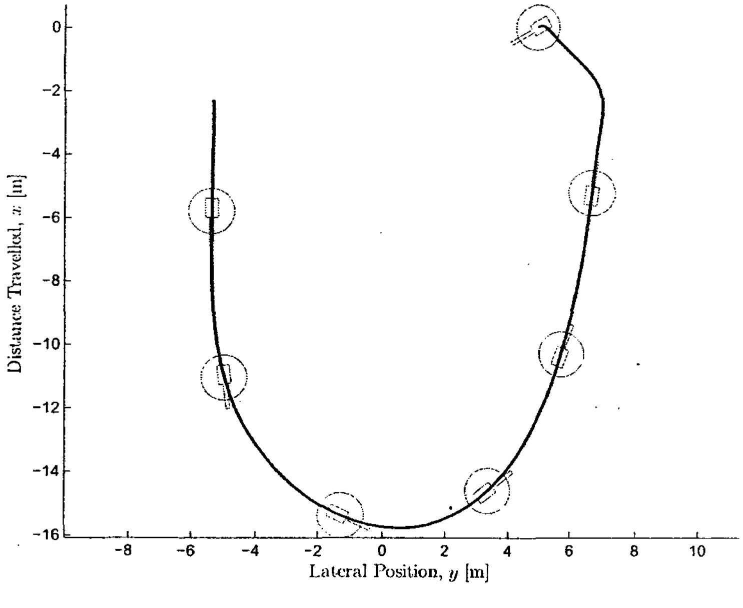


Figure 4.16: x-y trajectory of the closed-loop system with a linear controller.

Case Study 3: Rotorcraft MAV

In the third case study, the simplified lateral model derived for the Micro Mosquito MAV in Section 4.3.1 will be approximated with the SLP and UG approximation methods. Subsequently, a PWA controller is designed to make the MAV follow the straight line $y = 0$ at a constant forward speed u_0. To proceed, the nonlinear function $f(\psi, v)$ is evaluated with a desired error requirement of 0.012 and the physical parameters used both in the approximation and simulation are shown in Table 4.2[5].

The resulting SLP PWA approximation can be seen in Figure 4.17 and Table 4.3 compares the results between the SLP and UG methods using the algorithm in Section 3.2. The main result is that the SLP approximation requires 53 percent less regions than the UG approximation for $e_{des} = 0.012$. A SLP PWA controller is then designed for the rotorcraft to follow the straight line $y = 0$ using the methodology in

Table 4.2: Physical parameters used to approximate $f(\psi, v)$ for the Micro Mosquito

Parameter	Value/Range	Unit
u_0	0.0508	m/s
ψ	$\frac{-\pi}{2} < \psi < \frac{\pi}{2}$	rad
v	$-0.02 < v < 0.02$	m/s
m	0.025	kg
I_{zz}	0.001	$kg - m^2$
k_m	0.01	$-$
k_v	0.1	$-$
Q_t	0.05	$N - m$

Table 4.3: Approximation Results

	SLP Approximation		Uniform Grid Approximation	
e_{des}	e_{max}	Number of Regions	e_{max}	Number of Regions
0.012	**0.01**	**15**	**0.01**	**32**

[5] The maximum forward velocity for the MAV is 0.1016 m/s [86] and the maximum slideslip velocity, v, is assumed to be 20% of this value.

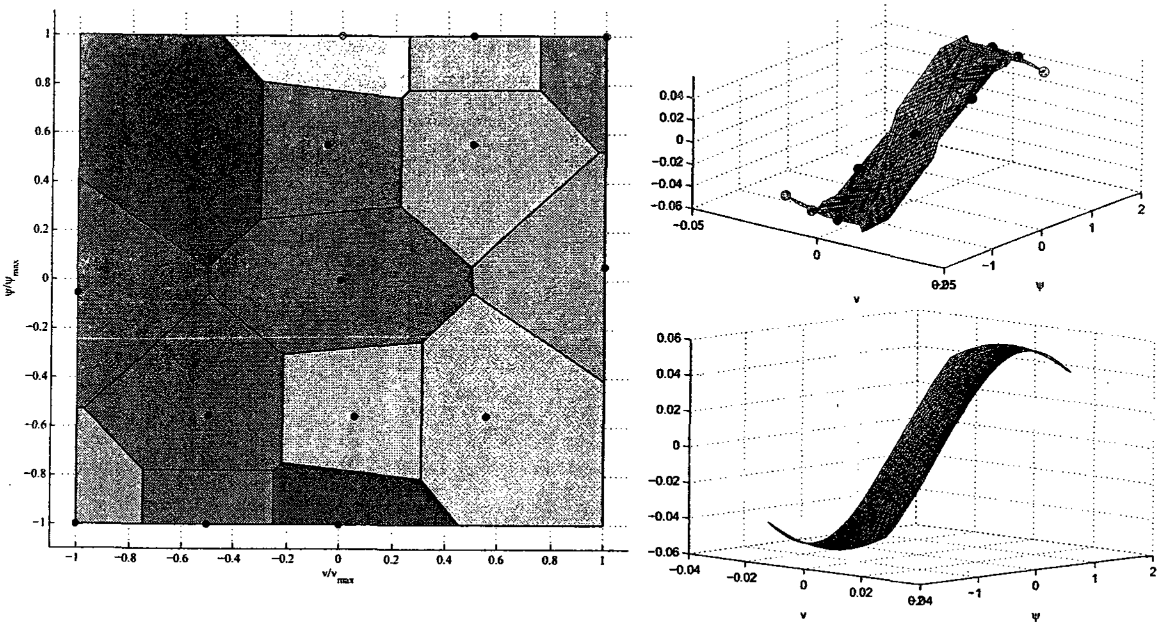


Figure 4.17: Voronoi Partition and PWA Approximation of $f(x_1, x_2)$.

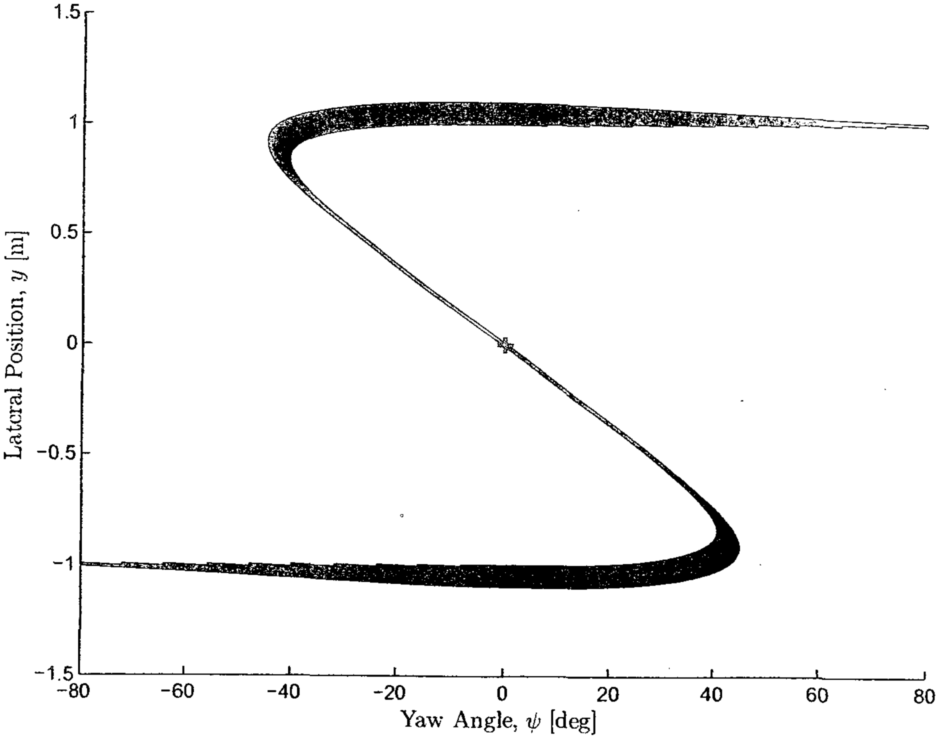


Figure 4.18: Closed-loop system trajectories with different initial conditions.

Section 4.3.2. Given a large set of initial conditions far from the origin, Figure 4.18 shows that the SLP PWA controller is able to stabilise the nonlinear system (4.17). However, if one were to apply a linear controller, the control objective is no longer achieved. Figure 4.19 shows that the nonlinear system in feedback with the linear controller is not stabilised unlike with the PWA controller.

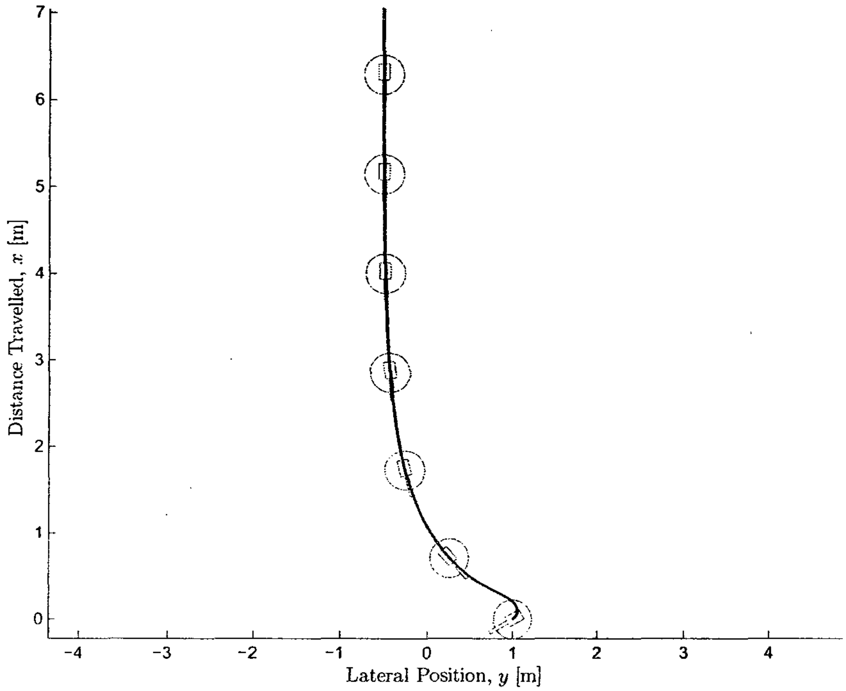


Figure 4.19: x-y trajectory of the closed-loop system with a linear controller for initial conditions $\psi_0 = \frac{\pi}{3}$, $y_0 = 1$.

4.4 Summary

In summary, this chapter presents the application of the SLP approximation methodology developed in Chapter 3 to three case studies. The first case study deals with a mass-spring-damper mechanical system with a nonlinear spring and damper. The SLP approximation is obtained and validated against the nonlinear model and a UG approximation through open loop simulations. The SLP approximation is shown to follow the nonlinear model more accurately than the UG approximation at the expense of approximating a nonlinearity to make its curvature continuous. The second case study involved a 3 DOF UAV model that was derived in Section 4.3.1. The PWA controller obtained using the methodology reviewed in Section 4.3.2 is shown to stabilise the nonlinear helicopter model in simulation for a large number of initial conditions far away from the equilibrium point. Furthermore, a linear controller in feedback connection with the nonlinear model is unable to stabilise the origin.. The final case study is the simplified lateral model of a Micro Mosquito rotorcraft MAV, which was also derived in Section 4.3.1. Similarly to the UAV, a PWA controller is designed so that the MAV can follow the straight line $y = 0$. The PWA controller is shown to stabilise the nonlinear MAV model for a large number of initial conditions far from the origin. A linear controller is, however, unable to stabilise the original nonlinear system.

4.5 Appendix

UAV PWA Model and PWA Controller

$$\dot{x} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -10 & 0 & 0 \\ 0 & -0.7 & -1 & 0 \\ 0.7 & 0 & 2 & 0 \end{bmatrix} x + \begin{bmatrix} 0 \\ 100 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ -0.5 \\ 0 \\ 0 \end{bmatrix} u \qquad \text{for } x \in \mathcal{R}_1 \tag{4.40}$$

$$\dot{x} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -10 & 0 & 0 \\ 0 & -0.7 & -1 & 0 \\ -1.05 & 0 & -1.4 & 0 \end{bmatrix} x + \begin{bmatrix} 0 \\ 100 \\ 0 \\ -2.64 \end{bmatrix} + \begin{bmatrix} 0 \\ -0.5 \\ 0 \\ 0 \end{bmatrix} u \qquad \text{for } x \in \mathcal{R}_2 \tag{4.41}$$

$$\dot{x} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -10 & 0 & 0 \\ 0 & -0.7 & -1 & 0 \\ -1.05 & 0 & -1.4 & 0 \end{bmatrix} x + \begin{bmatrix} 0 \\ 100 \\ 0 \\ 2.64 \end{bmatrix} + \begin{bmatrix} 0 \\ -0.5 \\ 0 \\ 0 \end{bmatrix} u \qquad \text{for } x \in \mathcal{R}_3 \tag{4.42}$$

$$\dot{x} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -10 & 0 & 0 \\ 0 & -0.7 & -1 & 0 \\ -0.22 & 0 & 0.18 & 0 \end{bmatrix} x + \begin{bmatrix} 0 \\ 100 \\ 0 \\ -0.87 \end{bmatrix} + \begin{bmatrix} 0 \\ -0.5 \\ 0 \\ 0 \end{bmatrix} u \qquad \text{for } x \in \mathcal{R}_4 \tag{4.43}$$

$$\dot{x} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -10 & 0 & 0 \\ 0 & -0.7 & -1 & 0 \\ 1.02 & 0 & 1.41 & 0 \end{bmatrix} x + \begin{bmatrix} 0 \\ 100 \\ 0 \\ -0.62 \end{bmatrix} + \begin{bmatrix} 0 \\ -0.5 \\ 0 \\ 0 \end{bmatrix} u \qquad \text{for } x \in \mathcal{R}_5 \tag{4.44}$$

$$\dot{x} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -10 & 0 & 0 \\ 0 & -0.7 & -1 & 0 \\ -0.22 & 0 & 0.18 & 0 \end{bmatrix} x + \begin{bmatrix} 0 \\ 100 \\ 0 \\ 0.869 \end{bmatrix} + \begin{bmatrix} 0 \\ -0.5 \\ 0 \\ 0 \end{bmatrix} u \qquad \text{for } x \in \mathcal{R}_6 \tag{4.45}$$

$$\dot{x} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -10 & 0 & 0 \\ 0 & -0.7 & -1 & 0 \\ 1.02 & 0 & 1.41 & 0 \end{bmatrix} x + \begin{bmatrix} 0 \\ 100 \\ 0 \\ 0.62 \end{bmatrix} + \begin{bmatrix} 0 \\ -0.5 \\ 0 \\ 0 \end{bmatrix} u \qquad \text{for } x \in \mathcal{R}_7 \tag{4.46}$$

$$\dot{x} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -10 & 0 & 0 \\ 0 & -0.7 & -1 & 0 \\ 0.65 & 0 & 1.62 & 0 \end{bmatrix} x + \begin{bmatrix} 0 \\ 100 \\ 0 \\ -0.095 \end{bmatrix} + \begin{bmatrix} 0 \\ -0.5 \\ 0 \\ 0 \end{bmatrix} u \qquad \text{for } x \in \mathcal{R}_8 \tag{4.47}$$

$$\dot{x} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -10 & 0 & 0 \\ 0 & -0.7 & -1 & 0 \\ 0.65 & 0 & 1.62 & 0 \end{bmatrix} x + \begin{bmatrix} 0 \\ 100 \\ 0 \\ 0.095 \end{bmatrix} + \begin{bmatrix} 0 \\ -0.5 \\ 0 \\ 0 \end{bmatrix} u \qquad \text{for } x \in \mathcal{R}_9 \tag{4.48}$$

$$\dot{x} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -10 & 0 & 0 \\ 0 & -0.7 & -1 & 0 \\ -0.05 & 0 & 0.69 & 0 \end{bmatrix} x + \begin{bmatrix} 0 \\ 100 \\ 0 \\ -0.5 \end{bmatrix} + \begin{bmatrix} 0 \\ -0.5 \\ 0 \\ 0 \end{bmatrix} u \qquad \text{for } x \in \mathcal{R}_{10} \tag{4.49}$$

$$\dot{x} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -10 & 0 & 0 \\ 0 & -0.7 & -1 & 0 \\ -0.05 & 0 & 0.69 & 0 \end{bmatrix} x + \begin{bmatrix} 0 \\ 100 \\ 0 \\ 0.5 \end{bmatrix} + \begin{bmatrix} 0 \\ -0.5 \\ 0 \\ 0 \end{bmatrix} u \qquad \text{for } x \in \mathcal{R}_{11} \tag{4.50}$$

$$\dot{x} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -10 & 0 & 0 \\ 0 & -0.7 & -1 & 0 \\ -0.9 & 0 & -0.71 & 0 \end{bmatrix} x + \begin{bmatrix} 0 \\ 100 \\ 0 \\ -1.92 \end{bmatrix} + \begin{bmatrix} 0 \\ -0.5 \\ 0 \\ 0 \end{bmatrix} u \qquad \text{for } x \in \mathcal{R}_{12} \tag{4.51}$$

$$\dot{x} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -10 & 0 & 0 \\ 0 & -0.7 & -1 & 0 \\ -0.9 & 0 & -0.71 & 0 \end{bmatrix} x + \begin{bmatrix} 0 \\ 100 \\ 0 \\ 1.92 \end{bmatrix} + \begin{bmatrix} 0 \\ -0.5 \\ 0 \\ 0 \end{bmatrix} u \qquad \text{for } x \in \mathcal{R}_{13} \tag{4.52}$$

where $x = \begin{bmatrix} \psi & r & v & y \end{bmatrix}^T$

$$\mathcal{R}_1 = \{x \in \mathcal{V}(X_1),\ \ X_1 = (0,0)\} \tag{4.53}$$
$$\mathcal{R}_2 = \{x \in \mathcal{V}(X_2),\ \ X_2 = (-1,-1)\} \tag{4.54}$$
$$\mathcal{R}_3 = \{x \in \mathcal{V}(X_3),\ \ X_3 = (1,1)\} \tag{4.55}$$
$$\mathcal{R}_4 = \{x \in \mathcal{V}(X_4),\ \ X_4 = (-0.67,-0.33)\} \tag{4.56}$$
$$\mathcal{R}_5 = \{x \in \mathcal{V}(X_5),\ \ X_5 = (-0.94,1)\} \tag{4.57}$$
$$\mathcal{R}_6 = \{x \in \mathcal{V}(X_6),\ \ X_6 = (0.67,0.33)\} \tag{4.58}$$
$$\mathcal{R}_7 = \{x \in \mathcal{V}(X_7),\ \ X_7 = (0.94,-1)\} \tag{4.59}$$
$$\mathcal{R}_8 = \{x \in \mathcal{V}(X_8),\ \ X_8 = (-0.61,0.33)\} \tag{4.60}$$
$$\mathcal{R}_9 = \{x \in \mathcal{V}(X_9),\ \ X_9 = (0.61,0.33)\} \tag{4.61}$$
$$\mathcal{R}_{10} = \{x \in \mathcal{V}(X_{10}),\ \ X_{10} = (-1,0.056)\} \tag{4.62}$$
$$\mathcal{R}_{11} = \{x \in \mathcal{V}(X_{11}),\ \ X_{11} = (1,-0.056)\} \tag{4.63}$$
$$\mathcal{R}_{12} = \{x \in \mathcal{V}(X_{12}),\ \ X_{12} = (-1,-0.56)\} \tag{4.64}$$
$$\mathcal{R}_{13} = \{x \in \mathcal{V}(X_{13}),\ \ X_{13} = (1,0.56)\} \tag{4.65}$$

$$\bar{K}_1 = \bar{K}_4 = \bar{K}_6 = \bar{K}_8 = \bar{K}_9 = \bar{K}_{10} = \bar{K}_{11} = \begin{bmatrix} 40.28 & 18.96 & 15.39 & 10 & 200 \end{bmatrix} \tag{4.66}$$
$$\bar{K}_2 = \begin{bmatrix} 25.18 & 18.96 & 0.0015 & 10 & 173.82 \end{bmatrix} \tag{4.67}$$
$$\bar{K}_3 = \begin{bmatrix} 20.9 & 18.96 & -4.36 & 10 & 233.6 \end{bmatrix} \tag{4.68}$$
$$\bar{K}_5 = \begin{bmatrix} 38.04 & 18.7 & 3.01 & 4.75 & 184.95 \end{bmatrix} \tag{4.69}$$
$$\bar{K}_7 = \begin{bmatrix} 40.57 & 18.96 & 15.08 & 10 & 199.5 \end{bmatrix} \tag{4.70}$$
$$\bar{K}_{12} = \begin{bmatrix} 23.51 & 18.96 & 9.69 & 10 & 184.85 \end{bmatrix} \tag{4.71}$$
$$\bar{K}_{13} = \begin{bmatrix} 20.9 & 18.96 & 8.8 & 10 & 217.51 \end{bmatrix} \tag{4.72}$$

Chapter 5

Conclusion

This chapter summarises the main conclusions that can be drawn from this work based on the contributions of the thesis that are stated in Chapter 1. Subsequently, potential future work is proposed that can extend the present research. The focus of this thesis has been to develop a new methodology for obtaining PWA models using a SLP. The main conclusions are stated in the following:

- To develop a new approximation method to obtain PWA models using a SLP and Voronoi partitions.

  Chapter 2 proposed a new approximation method using a SLP. In this method, the approximation of a smooth curve is obtained by using its curvature as a measure for selecting linearisation points. Subsequently, the proposed approximation method is extended to smooth surfaces in Chapter 3, where the domain of the surface is partitioned with a Voronoi partition and each Voronoi generator is chosen to be a linearisation point. A useful extension to the proposed approximation algorithm is to accommodate more precision at certain regions of interest (e.g. for stability purposes). Another interesting extension would be to change the error function to be related to the open loop trajectories of the nonlinear system instead of the approximation error.

Conclusions

- The SLP method requires that the smooth nonlinear curve or surface has a continuous second derivative.
- The SLP method has the potential for generating a discontinuous PWA approximation while the UG method is continuous.

- To successfully apply the SLP approximation method to case studies and to compare it to the UG approximation method in the literature.

  In Chapter 4, the SLP approximation method is successfully applied to three case studies. The first case study is a mechanical system with a nonlinear spring and damper. The SLP PWA system is able to follow the open loop response of the nonlinear system better in comparison to the UG PWA system. The second case involved a 3 DOF UAV model that was derived in Section 4.3.1. The UAV model was previously approximated as a PWA system using a SLP and a UG in Section 3.2.1 using the comparison algorithm proposed in Section 3.2. The result of this comparison is that the SLP PWA model requires a significantly less amount of regions than the UG PWA approximation for a given desired error requirement. A PWA controller is then designed using the methodology reviewed in Section 4.3.2. The PWA controller is found to stabilise the nonlinear system in simulation for a large number of initial conditions far away from the equilibrium point. Also, a linear controller in feedback connection with the nonlinear model is unable to stabilise the origin. The third case applied the SLP approximation algorithm to a simplified Micro Mosquito MAV model that was derived in Section 4.3.1. The SLP approximation requires 53 percent less regions than the UG approximation using the comparison algorithm in Section 3.2. A PWA controller is designed with the same control objective as the UAV. Simulation results show that the PWA controller stabilises the nonlinear MAV plant, while a linear controller is unable to do so. A useful extension to this work would be to extend the SLP approximation method to smooth nonlinear functions of more than two variables.

Conclusions

- In all surface approximation examples, the SLP method requires significantly less regions than the UG method.
- In case 1, the UG method is able to handle the absolute value function while the SLP method required the function to be approximated.

Based on previous observations, potential future work that can be done to improve and continue this research includes:

1. To formulate an optimal approach to selecting the approximation points.
2. To guarantee the continuity of the approximation.
3. To accommodate more precision at certain regions of interest.
4. To change the error function to be related to the open loop trajectories of the nonlinear system instead of the maximum approximation error.
5. To extend the SLP approximation method to include smooth nonlinear functions of more than two variables.

Approximating nonlinear systems with smooth vector fields using a set of linearisation points thus seems to be a very rich field of study.

Bibliography

[1] Arthur F. Huber. Death by a thousand cuts : Micro-air vehicles in the service of air force missions. Technical report, Maxwell Air Force Base, 2001.

[2] Joel M. Grasmeyer and Matthew T. Keennon. Development of the Black Widow micro air vehicle. Technical report, AeroVironment, 2000.

[3] Delft Center for Systems and Control. Automatic autorotation of a rotorcraft uav. World Wide Web, http://www.dcsc.tudelft.nl/Education/ThesisProposals/proposal-5642.html, 2009.

[4] Gallery TUDelft. Delfly. World Wide Web, http://gallery.citg.tudelft.nl/Delfly/DelFly_2, 2009.

[5] Luis Rodrigues. *Dynamic Output Feedback Controller Synthesis For Piecewise-Affine Systems.* PhD thesis, Stanford University, 2002.

[6] Luis Rodrigues and Jonathan P. How. Automated Control Design for a Piecewise-Affine Approximation of a Class of Nonlinear Systems. In *Proceedings of the American Control Conference*, pages 3189–3194, 2001.

[7] Wolfgang Kühnel and Bruce Hunt. *Differential Geometry Curves-Surfaces-Manifolds.* The American Mathematical Society, 2nd edition edition, 2006.

[8] Kikuo Fujita, Noriyasa Hirokawa, and Tomoya Tachikawa. Voronoi diagram based cumulative approximation for engineering optimization. *The American Insitute of Aeronautics and Astronautics*, 2000.

[9] Wei Yue. Modelling and piecewise-affine control of an aerobatic helicopter. Master's thesis, Concordia University, 2005.

[10] Marco Storace and Oscar De Feo. Piecewise-linear approximation of nonlinear dynamical systems. *IEEE Transactions on Circuits and Systems*, Vol. 51(No. 4):830–842, 2004.

[11] M. Storace and O. De Feo. Pwl approximation of nonlinear dynamical systems, part-i: Structural stability. *Journal of Physics: Conference Series*, Vol. 22:208–221, 2005.

[12] A. Andronov and C. Chaikin. *Theory of Oscillations.* Princeton University Press, Princeton, New Jersey, 1949.

[13] Eduardo D. Sontag. Nonlinear regulation: The piecewise linear approach. *IEEE Transactions on Automatic Control*, Vol. AC-26(Number 2), April 1981.

[14] Pedro Julian, Alfredo Desages, and Osvaldo Agamennoni. High-Level Canonical Piecewise Linear Representation Using a Simplicial Partition. *IEEE Transactions on Circuits and Systems - I: Fundamental Theory and Applications*, Vol. 46(Number 4):463–480, 1999.

[15] Anders Rantzer and Mikael Johansson. Piecewise Linear Quadratic Optimal Control. *IEEE Transactions on Automatic Control*, Vol. 45(Number 4):629–637, 2000.

[16] Mikael Johansson. *Piecewise Linear Control Systems.* Springer, Berlin, 2003.

[17] Domenico Mignone, Giancarlo Ferrari-Trecate, and Manfred Morari. Stability and stabilization of piecewise affine and hybrid systems: An LMI approach. In *39th IEEE Conference on Decision and Control*, pages 504–509, 2000.

[18] Gang Feng. Controller design and analysis of uncertain piecewise-linear systems. *IEEE Transactions on Circuits and Systems - I: Fundamental Theory and Applications*, Vol. 49(No. 2):224–232, 2002.

[19] Jorge M. Gonçalves, Alexandre Megretski, and Munther A. Dahleh. Global analysis of piecewise linear systems using impact maps and surface lyapunov functions. *IEEE Transactions on Automatic Control*, Vol. 48(No. 12):2089–2106, 2003.

[20] E. Asarin, T. Dang, and A. Girard. *Hybrid Systems: Computation and Control*, pages 20–35. Springer, 2003.

[21] Luis Rodrigues and Jonathan P. How. Synthesis of piecewise-affine controllers for stabilization of nonlinear systems. In *Conference on Decision and Control*, pages 2071–2076, 2003.

[22] Samer Shehab and Luis Rodrigues. Preliminary results on UAV path following using piecewise-affine control. In *Proceedings of the IEEE on Control Applications*, pages 358–363, 2005.

[23] Wei Yue, Luis Rodrigues, and Brandon Gordon. Piecewise-affine control of a three DOF helicopter. In *Proceedings of the 2006 American Control Conference*, pages 3924–3929, 2006.

[24] S. Lebel and L. Rodrigues. Path Following of a Wheeled Mobile Robot Combining Piecewise-Affine Synthesis and Backstepping Approaches. In *IEEE American Control Conference*, pages 4518–4523, 2007.

[25] Shun ichi Azuma, Jun ichi Imura, and Toshiharu Sugie. Lebesgue piecewise affine approximation of nonlinear systems and its application to hybrid system modeling and biosystems. In *45th IEEE Conference on Decision and Control*, pages 2128–2133, 2006.

[26] Daniele Corona and Bart De Schutter. Adaptive cruise control for a SMART car: A comparison benchmark for MPC-PWA control methods. *IEEE Transactions on Control Systems Technology*, Vol. 16(Number 2):365–372, 2008.

[27] Richard E. Groff, Pramod P. Khargonekar, and Daniel E. Koditschek. A local convergence proof for the minvar algorithm for computing continous piecewise linear approximations. *SIAM Journal on Numerical Analysis*, Vol. 41(Number 3):87–93, 2003.

[28] Behzad Samadi and Luis Rodrigues. Piecewise-affine controller synthesis based on a local linear controller: Toolbox for MATLAB using PENBMI solver. Technical report, Concordia University, 2005.

[29] Scott A. Houser. Medial surface transformations for rapid approximation of casting solidification. Master's thesis, Virginia Polytechnic Institute and State University, 1996.

[30] P.R. Sibbald and P. Argos. Weighting aligned protein or nucleic acid sequences to correct for unequal representation. *Journal of Molecular Biology*, Vol. 216:813–818, 1990.

[31] Marina Piccinelli, Lorenzo Botti, Bogdan Ene-Iordache, Andrea Remuzzi, Alessandro Veneziani, and Luca Antiga. Link between vortex structures and Voronoi diagram in cerebral aneurysms. In *16th Congress of the European Society of Biomechanics*, 2008.

[32] Mar Serrano, Gianni De Fabritiis, Pep Espanõl, Eirik G. Flekkøy, and Petter V. Coveney. Mesoscopic dynamics of Voronoi fluid particles. *Journal of Physics A: Mathematical and General*, Vol. 35(Number 7):1605–1621, 2002.

[33] N. Sukumar, B. Moran, and T. Belytschko. The natural element method in solid mechanics. *International Journal for Numerical Methods in Engineering*, Vol. 43:839–887, 1998.

[34] G. W. Barlow. Hexagonal territories. *Animal Behaviour*, Vol. 22:876–878, 1974.

[35] Ryan Scott Causey. Lateral vison-based control autopilot for micro air vehicles using a horizon detection approach. Master's thesis, University of Florida, 2003.

[36] C. Wilson, J. Nutbean, and I. Bond. Aerodynamic and structural design of a solar powered micro unmanned air vehicle. *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, Vol. 214:97–105, 2000.

[37] Jung xia Zhan, Wei jun Wang, Zhe Wu, and Jin jun Wang. Wind tunnel experimental investigation on a fix-wing micro air vehicle. *Journal of Aircraft*, Vol. 43(Number 1):279–282, 2006.

[38] G. La Rosa, G. Mirone, and A. Risitano. Preliminary design and wind tunnel tests of a micro air vehicle for surveillance and sensor-bearing applications. *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, Vol. 219:1–10, 2005.

[39] Anthony M. DeLuca, Mark F. Reeder, Jacob Freeman, and Michael V. Ol. Flexible and rigid-wing micro air vehicle: Lift and drag comparison. *Journal of Aircraft*, Vol. 43(Number 2):572–575, 2006.

[40] Gabriel E. Torres. *Aerodynamics of Low Aspect Ratio Wings at Low Reynolds Numbers with Applications to Micro Air Vehicle.* PhD thesis, University of Notre-Dame, 2002.

[41] J. L. Lin, C. Y. Wei, and C. Y. Lin. Aerodynamic performance of thin wings at low reynolds numbers. *Aircraft Engineering and Aerospace Technology: An International Journal*, Vol. 79(Number 3):245–253, 2007.

[42] Rafal Zbikowski. Fly like a fly. *IEEE Spectrum*, November:46–51, 2005.

[43] Przemyslaw Marek and Ladislav Smrcek. Development of Dart MAV : Fixed wing hover-capable micro aerial vehicle. In *Advanced Engineering Design Conference*, 2006.

[44] Jason Jackowski, Kenneth Boothe, Roberto Albertani, Rick Lind, and Peter Ifju. Modeling the flight dynamics of a micro air vehicle. Technical report, University of Florida, 2004.

[45] Gabriel Torres and Thomas J. Mueller. Micro aerial vehicle development: Design, components, fabrication, and flight-testing. Technical report, University of Notre-Dame, 2001.

[46] Thomas J. Mueller. *Fixed and Flapping Wing Aerodynamics for Micro Air Vehicle Applications.* American Institute of Aeronautics and Astronautics, 2001.

[47] T. Spoerry and K. C. Wong. Design and development of a micro air vehicle (μAV) concept: Project Bidule. Technical report, School of Aerospace, Mechanical and Mechatronic Engineering, University of Sydney, 2006.

[48] Davide Prella, Alessandra Pollifrone, Francis Bé, and Tanya Potoreyko. Capstone final report: Micro air vehicle. Technical report, Concordia University, 2006.

[49] Sergey Shkarayev, William Null, and Matthew Wagner. Development of micro air vehicle technology with in-flight adaptive wing structure. Technical Report CR-2004-213271, NASA, 2004.

[50] Tamaki Yanagita, Tom Lind, Bharani Malladi, Dmytro Silin, Deva Coopamah, Travis Berka, Bo Han, Stephanie Rodriguez, and Sergey Shkarayev. Simulation, wind tunnel testing, and design of fixed and flapping wing micro air vehicles. *American Institute of Aeronautics and Astronautics*, 2006.

[51] W. Null and S. Shkarayev. Effect of camber on the aerodynamics of adaptive-wing micro air vehicles. *Journal of Aircraft*, Vol. 42(Number 6):1537–1541, 2005.

[52] Jason Grow, Julie Jones, Chris Szachta, Drew Walter, Bob Williams, Kelly Lagulli, and Matt DeHaven. Development of Thnikkaman micro air vehicle at rochester insitute of technology. In *Proceedings of KGCOE-MD2004: Multi-Disciplinary Engineering Design Conference*, 2004.

[53] M. Meenakshi and Seetharama Bhat M. Robust fixed-order H_2 controller for micro air vehicle - design and validation. *Optimal Control Applications and Methods*, 2006.

[54] Dong Sun, Huaiyu Wu, Chi Ming Lam, and Rong Zhu. Development of a small air vehicle based on aerodynamic model analysis in the tunnel tests. *Mechatronics*, Vol. 16(Number 1), 2006.

[55] Yongsheng Lian, Wei Shyy, Dragos Viieru, and Baoning Zhang. Membrane wing aerodynamics for micro air vehicles. *Progress in Aerospace Sciences*, pages 425–465, 2003.

[56] C. Roberts, M. Vaughn, and W. J. Bowman. Development of a solar powered micro air vehicle. *American Institute of Aeronautics and Astronautics*, 2002.

[57] William E. Green and Paul Y. Oh. A fixed-wing aircraft for hovering in caves, tunnels, and buildings. In *Proceedings of the American Control Conference*, 2006.

[58] William E. Green and Paul Y. Oh. A MAV that flies like an airplane and hovers like a helicopter. In *Proceedings of the IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, 2005.

[59] Randal W. Beard. Equations of motion for micro air vehicles. Technical report, Brigham Young University, 2005.

[60] Nathan B. Knoebel, Stephen R. Osborne, Joshua S. Matthews, Andrew M. Eldredge, and Randal W. Beard. Computationally simple model reference adaptive control for miniature air vehicles. In *Proceedings of the 2006 American Control Conference*, 2006.

[61] S. A. Ansari, R. Zbikowski, and K. Knowles. Nonlinear unsteady aerodynamic model for insect-like flapping wings in the hover: Part 2: Implementation and validation. *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, Vol. 220:169–186, 2006.

[62] T. K. Sengupta, V. Vikas, and A. Johri. An improved method for calculating flow past flapping and hovering airfoils. *Theoretical Computational Fluid Dynamics*, Vol. 19(Number 6):417–440, 2005.

[63] Jaime Alberto Escobar G. Aerodynamic analysis and design of a MAV's wing. Technical report, Concordia University, 2006.

[64] M. Drela and Michael B. Giles. Viscous-inviscid analysis of transonic and low Reynolds number airfoils. *AIAA Journal*, Vol. 25(Number 10):1347–1355, 1987.

[65] Roberto Albertani, Frank Boria, Scott Bowman, Dan Claxton, Aaron Crespo, Carlo Francis, Peter Ifju, Baron Johnson, Kyo Ho Lee, Mike Morton, and Mike

Sytsma. Development of reliable and mission capable micro air vehicles. Technical report, University of Florida, 2005.

[66] B. Stanford, R. Albertani, and P. Ifju. Static finite element validation of a flexible micro air vehicle. *Experimental Mechanics*, pages 283–294, 2007.

[67] Dragos Viieru, Robertyo Albertani, Wei Shyy, and Peter G. Ifju. Effect of tip vortex on wing aerodynamics of micro air vehicles. *American Institute of Aeronautics and Astronautics*, 2004.

[68] Yongsheng Lian, Wei Whyy, Peter G. Ifju, and Erwan Verron. Membrane wing model for micro air vehicles. *AIAA Journal*, Vol. 41(Number 12):2492–2494, 2003.

[69] K. Mohseni, D. Lawrence, D. Gyllhem, M. Culbreth, and P. Geuzaine. Flow simulation around a micro air vehicle in a plume characterization scenario. *American Institute of Aeronautics and Astronautics*, 2004.

[70] J. Hall, K. Mohseni, and D. Lawrence. Investigation of variable wing-sweep for applications in micro air vehicles. Technical report, University of Colorado, 2005.

[71] Pablo Kalika Burdman, Matthew Guest, Florian Nieto, Fabien Pradeloux, and Joel Senabre. Design of a micro air vehicle (MAV). Technical report, Institute of Technology, Linköping University, 2004.

[72] Rafal Zbikowski, Salman A. Ansari, and Kevin Knowles. On mathematical modelling of insect flight dynamics in the context of micro air vehicles. *Bioinspiration and Biomimetics*, Vol. 19(Number 6):26–37, 2006.

[73] Mujahid Abdulrahim. Dynamic characteristics of morphing micro air vehicles. Master's thesis, University of Florida, 2004.

[74] Mujahid Abdulrahim and Rick Lind. Modeling and control of micro air vehicles with biologically-inspired morphing. In *Proceedings of the 2006 American Control Conference*, 2006.

[75] Huaiyu Wu, Dong Sun, and Zhaoying Zhou. Model identification of a micro air vehicle in loitering flight based on attitude performance evaluation. *IEEE Transactions on Robotics*, Vol. 20(Number 4), August 2004.

[76] Huaiyu Wu, Dong Sun, and Zhaoying Zhou. Micro air vehicle: Configuration, analysis, fabrication, and test. *IEEE/ASME Transactions on Mechatronics*, Vol. 9(Number 1), March 2004.

[77] Rong Zhu, Dong Sun, and Zhaoying Zhou. Integrated design of trajectory planning and control for micro air vehicles. *Journal of Mechatronics*, Vol. 17:245–253, 2007.

[78] Huai-yu Wu, Zhao-wing Zhou, and Dong Sun. Autonomous hovering control and test for micro air vehicle. In *Proceedings of the 2003 IEEE International Conference on Robotics and Automation*, 2003.

[79] Ethan D. Bloch. *A First Course in Geometric Topology and Differential Geometry.* Birkhäuser, 1997.

[80] I.D. Faux and M.J. Pratt. *Computational Geometry for Design and Manufacture.* John Wiley & Sons, 1979.

[81] Mark de Berg, Marc van Kreveld, Mark Overmars, and Otfried Schwarzkopf. *Computational Geometry: Algorithms and Applications*, chapter 7. Springer-Verlag, 2nd revised edition, 2000.

[82] M. Kvasnica, P. Grieder, M. Baotic, and F.J. Christophersen. Multi-parametric toolbox (MPT). Technical report, Swiss Federal Institute of Technology, 2006.

[83] Stephen Boyd, Laurent El Ghaoui, Eric Feron, and Venkataramanan Balakrishnan. *Linear Matrix Inequalities in System and Control Theory.* SIAM, 1994.

[84] Michal Kočvara and Michael Stingl. PENBMI user's guide (version 2.1). World Wide Web, http://www.penopt.com/penbmi.html, 2006.

[85] J. Löfberg. YALMIP : A toolbox for modeling and optimization in MATLAB. In *Proceedings of the CACSD Conference*, Taipei, Taiwan, 2004. http://control.ee.ethz.ch/~joloef/yalmip.php.

[86] Radio Shack. Interactive toy concepts indoor Micro Mosquito helicopter. World Wide Web, http://www.radioshack.com, 2008.