4-15-2014

Continuous Polynomial Adaptive Estimator for Nonlinearly Parameterized Systems

Haitham B. Felemban
University of Connecicut, haitham.felemban@gmail.com

Continuous Polynomial Adaptive Estimator for Nonlinearly Parameterized Systems

Haitham Bahraldeen M Felemban

B.S., Northern Illinois University, 2012

A Thesis

Submitted in Partial Fulfillment of the

Requirements for the Degree of

Master of Science

at the

University of Connecticut

2014

APPROVAL PAGE

Masters of Science Thesis

Continuous Polynomial Adaptive Estimator for Nonlinearly Parameterized Systems

Presented by

Haitham Bahraldeen M Felemban, B.S.

Major Advisor _____
Prof. Chengyu Cao

Associate Advisor _____
Prof. Robert Gao

Associate Advisor _____
Prof. Jiong Tang

University of Connecticut

2014

ACKNOWLEDGMENTS

First, I would like to acknowledge my major advisor, Prof. Chengyu Cao, for his guidance and support over the past two years. Specifically, I appreciate his patience and encouragement, especially since I did not have experience in research. He has helped a lot during my Master's program.

In addition, I would like to thank my associate advisors, Prof. Robert Gao and Prof. Jiong Tang for their advice and encouragement.

I would like to extend my gratitude to my lab group members: Dr. Jie Luo, John Cooper, Michael Santone, Jiaxing Che, Dr. Fulai Yao, Yuqian Liu, Cyuansi Shih, Ma Tong, Robert Herman, Yushuo Niu, Siwei Han and Dr. Jianxin Zhou with who helped to make my research an enjoyable experience. Among these members, I would like to give special thanks to Jiaxing Che, whose support and kindness has helped me to overcome hard days and to build confidence in my myself and research.

A special thanks to the Majewski family, Kurt, Sherry, Ryan and his girlfriend Amanda, Kyle and Hannah who became my family in the USA. Likewise, I want to thank Grandma Janet and Grandpa David Syens for everything. In addition, a special thanks to Jhonattan and Jessica Yandun and their families for their kindness and support. Also, I would like to thanks my fellow Saudi brothers and their families: Faisal Klufah, Naseh Algehainy, Badar Al Marri, Sultan Al Yami, and Mohsen Abusaq who consistently helped me adapt to life in the USA and ensured I was never homesick.

This dissertation is dedicated to my parents and my brother and sister who have continuously given me love, support, and encouragement in my life.

Contents

ABSTRACT

This thesis presents the continuous polynomial adaptive estimator (CPAE) which estimates a nonlinear parameter in nonlinearly parameterized (NLP) systems. It combines the multiple region law with the companion adaptive system presented in [1] to come up with the CPAE. Stability is discussed and a general definition of persistence-of-excitation (PE) condition is proposed for parameter convergence. Simulation is included to illustrate the parameter convergence using the CPAE. As an application, the CPAE was successfully used to estimate the airspeed in presence of airspeed sensor failure on a developed academic aircraft model. As part of *Loss of Control Prevention through Adaptive Reconfiguration* project supported by NASA, the IMU theory method, which estimates airspeed using data from the inertial measurement unit (IMU) and the global positioning system (GPS), is presented and applied on the generic transport model (GTM). Conclusions and future work for aforementioned topics were presented at the end of this thesis.

Chapter 1

Introduction

1.1 Motivation

The analysis for any physical or mechanical systems can be done using empirical methods, where different input signals can be applied on the system. After this, the response can then be studied and analyzed. If the response does not meet the desired behavior, some factors will be adjusted or a compensator will be attached to change the response's behavior. However, this approach becomes limited especially when the physical system is too complicated, too expensive, or too dangerous to apply different input signals. Also, it requires experience which is gained through of trial and error studies. Therefore, it is preferred to capture the physical system in a mathematical model. This enables easy analysis, control design, and simulation.

However, mathematical models based on physical laws are usually qualitative instead of quantitative. There are many factors contributing to the mismatch between theoretical models and experimental data such as approximations and simplifications

during the derivation, various disturbances, and omitted physical processes. Nevertheless, these qualitative models are still very valuable because they provide a rough model structure as a starting point. This significantly reduces the set of models to be focused on from an almost infinite number of choices. To bridge the gap between qualitative model and experimental data, free parameters are added which gives the model the ability to match reality while still keeping useful structures based on physical laws. Hence, the qualitative model will be transformed into a parameterized one with the potential to be corrected to match experimental data. The next step is to correct the model by finding approximate parameter values to match experimental input/output data using different parameters estimations techniques.

In general, parameter estimation can be defined as the process of calculating model parameters based on input and output data, in this thesis adaptive estimation is the focuses.

1.2 Thesis Contribution

This thesis focuses on parameter estimation in nonlinearly parameterized (NLP) systems where the unknown parameter occurs nonlinearly. Chapter 2 will introduce the multiple region law and combine it with the companion adaptive system in [1] to come up with the continuous polynomial estimator (CPAE). This chapter will also introduce the general definition of persistence-of-excitation (PE) condition to guarantee parameter convergence. Those results will be submitted to *System and control letters* journal. In addition, Chapter 3 successfully applies CPAE to estimate the airspeed in the presence of a sensor failure for an academic aircraft model. Those results were

published in *SciTech 2014: Guidance, Navigation and Control* conference [2]. Chapter 4 estimates airspeed using data from both inertial measurement units (IMU) and global positioning systems (GPS) in the generic transport model (GTM) provided by NASA.

1.3 Thesis Outline

This thesis is organized as follows, Chapter 2 will theoretically introduce the CPAE structure and discuss the stability using the Lyapunov stability criteria. Moreover, a general PE condition will be introduced for parameter convergence along with an example and simulation. Chapter 3 will be an application of the CPAE, where it was applied on a developed academic aircraft model to estimate the airspeed in the presence of sensor failure. Finally in Chapter 4, a practical approach to estimate airspeed using data from the IMU and GPS is presented and applied to the GTM as part of Loss of Control Prevention through Adaptive Reconfiguration project supported by NASA. Chapter 5 will state conclusions and future work.

Chapter 2

The Continuous Polynomial Adaptive Estimator (CPAE)

2.1 Introduction

Most systems in nature can be treated as nonlinearly parameterized (NLP) systems, where parameters occur nonlinearly. For instance, the Hill equation for modeling system biology is highly composed of NLP functions [3], and most kinetic models have nonlinear rates in biochemical models [4]. In addition, several dynamical models such as friction dynamics [5], uncertainties in robot manipulators [6] and others [7] can be treated as NLP systems.

NLP systems are an intensively explored area in today's literatures. In adaptive control, [8, 9] focused on designing adaptive control for nonlinear convex/concave parameterizations systems. [10] developed an adaptive control for a class of first-order nonlinearly parameterized plants. [11] presented an error model approach which de-

scribes the relation between the tuning and parameter error and uses the minmax optimization procedure to make the error go to zero. In [12], an adaptive control based on results in [9] is proposed, where it is extended to n^{th} order system with triangular structure. [13, 14] introduce a smooth and nonsmooth framework for global adaptive control of a significant class of NLP systems with uncontrollable unstable linearization. By separating the unknown nonlinear parameter from the nonlinear function and adding a power integrator, a feedback domination design approach was developed. [15] presented an adaptive control for a system with NLP fuzzy approximation, where the radial basis fuction (RBF) used to construct fuzzy approximation and the adaptive control will tune all the RBF parameters to improve the control performance by reducing approximation error. Furthermore, [16] designed a control law by introducing a nonlinear biasing vector function into parameter estimation in order to link both system dynamics to estimate error dynamics. That leads to a new Lyapunov function and a set of conditions to achieve global asymptotic stability. Nevertheless, [17] designed a smooth adaptive state-feedback controller for high-order stochastic NLP systems. [18] addressed the problem of output tracking for NLP systems with unstabilizable linearization, by employing the idea of universal adaptive control and adding a power integrator combined with the technique of changing supply rate for input-to-state stable (ISS) Lyapunov function systems. Also, [19] introduced an adaptive tracking for periodically time-varying NLP systems by combining multilayer neural networks (MNN) and fourier series expansions (FSE) into a novel approximator. Then, combine the dynamic surface control (DSC) approach and integral-type Lyapunov function (ILF) technique to design the control algorithm. In addition, [19] focused on adaptive backstepping fuzzy-control (ABFC) approach for NLP systems with periodic disturbances, where a novel approximator function

based on fuzzy-logic system (FLS) and Fourier series expansion (FSE) is proposed to approximate the unknown system functions. Then, dynamic-surface-control (DSC) approach for strict-feedback and periodically time-varying systems with unknown control-gain functions was developed. In [20], an adaptive observer for NLP class of nonlinear MIMO systems is constructed under well-defined persistent excitation condition for guaranteed convergence.

Despite of the fact that NLP is being intensively explored, most of the previous literatures focused on the control aspect of NLP system. In this chapter, a pure parameter estimation problem is in focused. There are some literatures focused specifically on parameter estimation for NLP systems. For instance, in [21] a new technique for the adaptive parameter estimation in NLP systems was introduced, where an uncertainty set-update approach is proposed that makes the uncertainty set around the true value to vanish. Also [22], considers a class of systems influenced by perturbations that are NLP by unknown constant parameters, and constructs an update law to asymptotically invert a nonlinear equation. However, the previous introduced literatures fail in some practical scenarios and subject to various restrictions.

This chapter introduces a new class of parameter estimation in NLP systems. For this parameter estimation problem in NLP systems, all states are measured and inputs are known. The continuous polynomial adaptive estimator (CPAE) is an extension of [23, 24] and based on the work presented in [1, 25, 26]. The CPAE consists of two parts; a companion adaptive system and the multiple region law. The companion adaptive system can be applied on any NLP system which can be approximated using a piecewise linear function and with all states measured. The companion adaptive system consists of two parts, a companion model and an adaptive law. Here, the companion model is deterministic without any unknown information, thus, the

multiple region approach is established to estimate the unknown nonlinear parameter. The general structure of the CPAE can be seen in Figure (2.1.1). Nevertheless, the scalar case for the nonlinear parameter is considered here with a potential to extend the CPAE to cover the vector form as future work.

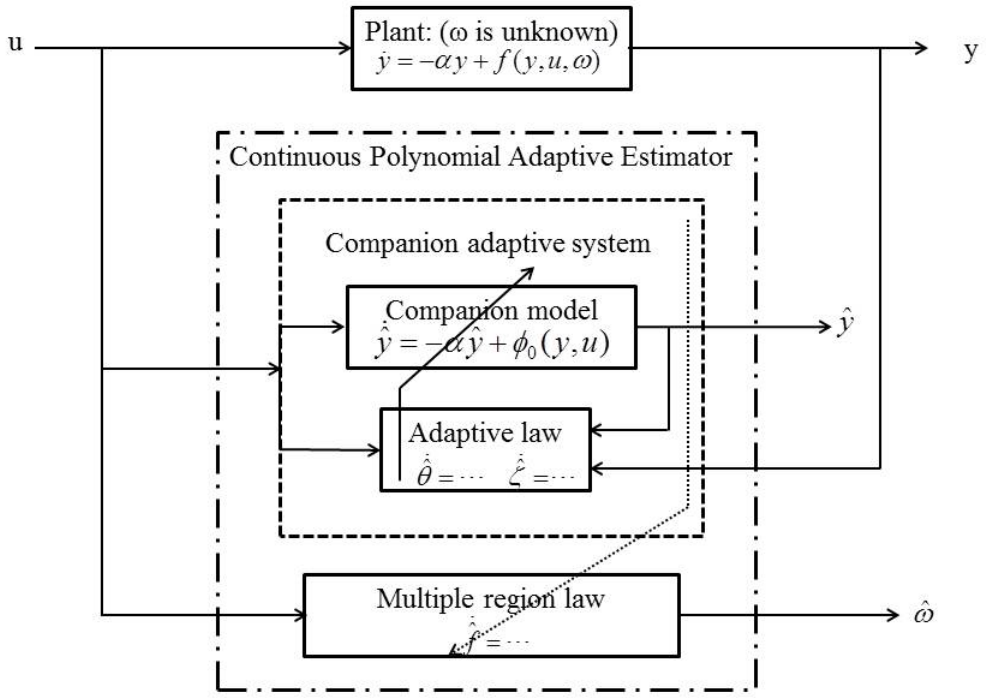

FIGURE 2.1.1: General structure of the CPAE

The rest of this Chapter organized as follow, Section 2.2 will state the problem formulation. In Section 2.3, the CPAE algorithm and the stability analysis are discussed. In Section 2.4, general persistence-of-excitation (PE) condition is proposed. An example along with simulation results are presented in Section 2.5. Finally, a summary about this section is stated in Section 2.6.

2.2 Problem Formulation

Consider the scalar case of the following system

$$\dot{y} = -\alpha y + f(y, u, \omega) \tag{2.2.1}$$

Where $\alpha > 0$ is a known constant, $y \in \Re$ is the state variable, $u \in \Re^m$ is the input signal, ω is the unknown nonlinear parameter. Two assumptions were made:

Assumption 1: ω belongs to a compact set Ω such as $\Omega = [\Omega_{\min}, \Omega_{\max}] \subset \Re$.

Assumption 2: $\forall\ y(t)$ and $u(t)$, the function f can be approximated by a piecewise linear function over Ω with reasonable approximation error, which means there exists a constant $d_{\max} > 0$.

2.2.1 Transformation Into Piecewise Linear Function

Start with dividing Ω into N equivalent exclusive small regions such as

$$\Omega \subseteq \bigcup_{i=1}^{N} \Omega_i, \quad \Omega_i = [\underline{\Omega_i}, \overline{\Omega_i}] \quad i = 1 \ldots N, \tag{2.2.2}$$

The approximation error can be defined as

$$|d(t)| = |m_i(y, u, \omega) + r_i(y, u, \omega)(\omega - \bar{\omega}_i) - f(y, u, \omega)| \leq d_{max} \tag{2.2.3}$$

Where $\omega \in \Omega_i$ and $\overline{\omega}_i = \frac{\overline{\Omega_i} + \underline{\Omega_i}}{2}$, and also m_i and r_i are defined as follow

$$m_i(y, u) = f(y, u, \overline{\omega}_i) \tag{2.2.4}$$

$$r_i(y, u) = \frac{\partial f(y, u, \omega)}{\partial \omega}|_{\overline{\omega}_i} \tag{2.2.5}$$

Note, the reason for dividing the compact set into N small regions is because a large class of encountered functions can be piecewise linearly approximated since any smooth function can be linearized locally. For example, if any function f is differentiable and its second order derivative w.r.t ω is bounded

$$\frac{\partial^2(y,u,\omega)}{\partial\omega^2} \leq q \tag{2.2.6}$$

The compact set Ω is divided into N regions, then

$$|m_i(y,u,\omega) + r_i(y,u,\omega)(\omega - \bar{\omega}_i) - f(y,u,\omega)| \leq \frac{q(\Omega_{max} - \Omega_{min})^2}{8N^2} \tag{2.2.7}$$

Comparing Equation (2.2.7) with Equation (2.2.3), the approximation error can be arbitrarily reduced by increasing number of regions N.

2.2.2 Mapping The Unknown Parameter

The unknown parameter ω is mapped into a pair of parameters $[\theta, \zeta]$, θ is a discrete parameter that indicates the small regions in Ω that ω belongs to, and is defined as

$$\theta \in \Theta = \theta_1, \ldots \theta_i, \ldots \theta_N, \quad \theta = \theta_i \quad \text{if} \quad \omega \in \Omega_i, \quad \theta_i = \tfrac{i-1}{(N-1)\Theta_{max}} \tag{2.2.8}$$

Where Θ_{max} is an arbitrary positive number. Furthermore, ζ is a continuous variable which indicates the offset of ω from the center of the small region, and is defined as

$$\zeta \in [-\zeta_{max}, \zeta_{max}], \quad \zeta = \omega - \tfrac{\overline{\Omega_i}+\underline{\Omega_i}}{2} \quad \text{if} \quad \omega \in \Omega_i, \quad \zeta_{max} = max_{i=1\ldots N} \tfrac{\overline{\Omega_i}-\underline{\Omega_i}}{2} \tag{2.2.9}$$

Using Equation (2.2.2), (2.2.3), (2.2.4), (2.2.5), (2.2.8) and (2.2.9), the problem formulation in Equation (2.2.1) is equivalently transformed into

$$\dot{y} = -\alpha y + m(y, u, \theta) + r(y, u, \theta)\zeta + d(t) \tag{2.2.10}$$

Knowing that,

$$\begin{aligned} &m(y, u, \theta) = m_i(y, u) \\ &r(y, u, \theta) = r_i(y, u) \\ &i = (N-1)\Theta_{max}\theta + 1 \\ &|d(t)| = f(y, u, \omega) - m(y, u, \theta) - r(y, u, \theta)\zeta \end{aligned} \tag{2.2.11}$$

In Equation (2.2.10), $m(y, u, \theta)$ and $r(y, u, \theta)$ are not available since θ is unknown, but $m_i(y, u, \theta_i)$ and $r_i(y, u, \theta_i)$ are available for each θ_i.

2.3 Continuous Polynomial Adaptive Estimator

In this section, the CPAE algorithm is introduced which contains two parts, the companion adaptive system and the multiple region law.

2.3.1 Companion Adaptive System

Here is the companion adaptive system which consists of two parts, a companion model and an adaptive law. The companion model is composed as follows:

$$\dot{\hat{y}} = -\alpha\hat{y} + \phi_0 \tag{2.3.1}$$

Here, the companion model estimates every $f(y,u,\omega)$ as $\phi_0(y,u)$ which is deterministic w.r.t y and u. Recall the problem transformation in Equation (2.2.10), note that it has $2N$ freedoms, N offset values and N slope rates. That means $2N-1$ auxiliary estimates are needed which will be governed by the adaptive law which can be seen as follows,

$$\begin{aligned}
\dot{\hat{\theta}} &= \begin{cases} 0 & \text{if} \tilde{y}\phi_i > 0 \quad \text{and} \widehat{\theta}_i \geq \Theta_{max} \\ 0 & \text{if} \tilde{y}\phi_i < 0 \quad \text{and} \widehat{\theta}_i \leq 0 \\ \tilde{y}\phi_i & otherwise \end{cases} \\
&\forall i = 1, \cdots, N-1 \\
\dot{\hat{\zeta}} &= \begin{cases} 0 & \text{if} \tilde{y}\eta_i > 0 \quad \text{and} \widehat{\zeta}_i \geq \zeta_{max} \\ 0 & \text{if} \tilde{y}\eta_i < 0 \quad \text{and} -\widehat{\zeta}_i \leq \zeta_{max} \\ \tilde{y}\eta_i & otherwise \end{cases} \\
&\forall i = 0, \cdots, N-1
\end{aligned} \tag{2.3.2}$$

Where

$$\tilde{y} = \hat{y} - y \tag{2.3.3}$$

$$\Phi = [\phi_0, \ldots, \phi_{N-1}] = A_m^{-1}(C_m - A_r C_\eta) \tag{2.3.4}$$

$$\eta = [\eta_0, \ldots, \eta_{N-1}] = -A_r^{-1} C_r \tag{2.3.5}$$

$$C_r = [r(y,u,\theta_1), \ldots, r(y,u,\theta_i), \ldots, r(y,u,\theta_N)]^T \tag{2.3.6}$$

$$C_m = [m(y,u,\theta_1),\ldots,m(y,u,\theta_i),\ldots,m(y,u,\theta_N)]^T \tag{2.3.7}$$

$$C_\eta = [\widehat{\zeta}_0\eta_0,\ldots,\widehat{\zeta}_0\eta_i,\ldots,\hat{\zeta}_{N-1}\eta_{N-1}]^T \tag{2.3.8}$$

A_m is N by N matrix and defined as follows

$$A_r = \begin{bmatrix} 1 & \cdot\cdot & \cdot\cdot & \cdot\cdot \\ \cdot & \cdot\cdot & \cdot\cdot & \cdot\cdot \\ 1 & \cdot\cdot & a_{ij} & \cdot\cdot \\ \cdot & \cdot\cdot & \cdot\cdot & \cdot\cdot \end{bmatrix} \quad where \quad a_{ij} = \theta_i^{j-1} \tag{2.3.9}$$

In addition, A_m is also an N by N matrix and is defined below

$$\begin{aligned} &a_{i1} = 1 \quad 1 \le \mathrm{i} \le \mathrm{N} \\ &a_{ij} = -g_{j-1}(\hat{\theta}_{j-1} - \theta_i) \quad 1 \le \mathrm{i} \le \mathrm{N}, 2 \le \mathrm{j} \le \mathrm{N} \\ &g_i(x) = \begin{cases} x^{i-1} & \text{if} i \quad \text{Even} \\ kx^{i-1} + x^{i-2} & \text{if} i \quad \text{Odd} \end{cases} \\ &k = \frac{N-1}{N\Theta_{max}} \end{aligned} \tag{2.3.10}$$

Stability Analysis

To consider the stability, the Lyapunov function is introduced as

$$V = \frac{\tilde{y}^2}{2} + \sum_{i=2}^{N} p_i(\tilde{\theta}_{i-1}) + \sum_{i=0}^{N-1} \theta^i \frac{\tilde{\zeta}_i^2}{2} \tag{2.3.11}$$

Where the polynomial function (that's why the name has polynomial) p_i is chosen

to be

$$p_i(x) = \begin{cases} \frac{x^i}{i} & \text{if} i \quad \text{Even} \\ k\frac{x^i}{i} + \frac{x^{i-1}}{i-1} & \text{if} i \quad \text{Odd} \end{cases} \tag{2.3.12}$$

Note, the function g_i in the definition of the A_m matrix in Equation (2.3.10) is nothing but the derivative of the function p_i in Equation (2.3.12). $\tilde{\theta}_i$ is defined as $\tilde{\theta}_i = \hat{\theta}_i - \theta_i$. It can be seen that $\tilde{\theta}_i \in [-\Theta_{max}, \Theta_{max}]$ since $0 \leq \left|\tilde{\theta}_i(t)\right| \leq \Theta_{\max}$ because

$$\begin{aligned} \dot{\hat{\theta}}_i \leq 0 \quad if \quad \hat{\theta}_i \geq \Theta_{\max} \\ \dot{\hat{\theta}}_i \geq 0 \quad if \quad \dot{\hat{\theta}}_i \leq 0 \end{aligned} \tag{2.3.13}$$

That means the function p_i is a well-posed Lyapunov function candidate over $[-\Theta_{max}, \Theta_{max}]$ since

$$\begin{aligned} &p_i(0) = 0 \\ &g_i(x) = \tfrac{dp_i(x)}{dx} < 0, x \in [-\Theta_{\max}, 0) \\ &g_i(x) = \tfrac{dp_i(x)}{dx} > 0, x \in (0, \Theta_{\max}] \end{aligned} \tag{2.3.14}$$

This justifies the choice of p_i.

The following lemma will show that the Lyapunov function is a non-increasing function.

Lemma 1

For the system in Equation (2.2.10) and the companion adaptive system in Equation (2.3.1) and (2.3.2), if $d_{max} = 0$ then

$$\dot{V} \leq -\alpha \tilde{y}^2 \tag{2.3.15}$$

Proof: First, consider the ideal case $d_{max} = 0$, then the adaptive law in Equation (2.3.2) can be rewritten as

$$\begin{aligned} \dot{\hat{\theta}}_i &= \tilde{y}\phi_i + v_i, \quad i = 1, \cdots, N-1 \\ \dot{\hat{\zeta}}_i &= \tilde{y}\eta_i + w_i, \quad i = 0, \cdots, N-1 \end{aligned} \tag{2.3.16}$$

where

$$\begin{cases} v_i = 0 \quad if \quad \dot{\hat{\theta}}_i \in (0, \Theta_{\max}) \\ v_i \leq 0 \quad if \quad \dot{\hat{\theta}}_i \geq \Theta_{\max} \\ v_i \leq 0 \quad if \quad \dot{\hat{\theta}}_i \leq 0 \\ w_i = 0 \quad if \quad \dot{\hat{\zeta}}_i \in (-\zeta_{\max}, \zeta_{\max}) \\ w_i \leq 0 \quad if \quad \dot{\hat{\zeta}}_i \geq \zeta_{\max} \\ w_i \leq 0 \quad if \quad \dot{\hat{\zeta}}_i \leq -\zeta_{\max} \end{cases} \tag{2.3.17}$$

Knowing

$$\begin{aligned} \tilde{\theta}_i &= \hat{\theta}_i - \theta, \quad i = 1, \cdots, N-1 \\ \tilde{\zeta}_i &= \tilde{\zeta}_i - \zeta, \quad i = 0, \cdots, N-1 \end{aligned} \tag{2.3.18}$$

The estimation error is defined by plugging in both Equation (2.2.10) and Equation (2.3.1) into Equation (2.3.7)

$$\dot{\tilde{y}} = \alpha\tilde{y} + \phi_0 - (m(y, u, \omega) + r(y, u, \omega)\zeta) \tag{2.3.19}$$

Combining Equation (2.3.16), (2.3.19) and (2.3.11), then

$$\begin{aligned}\dot{V} = &-\alpha\tilde{y}^2 - \sum_{i=1}^{N-1} g_i(\tilde{\theta}_i)v_i + \sum_{i=0}^{N-1} \theta^i\tilde{\zeta}_i w_i \\ &+\tilde{y}\left[\phi_0 - (m(y,u,\theta) + r(y,u,\theta)\zeta) + \sum_{i=1}^{N-1} g_i(\tilde{\theta}_i)\phi_i + \sum_{i=0}^{N-1}\theta^i\tilde{\zeta}_i\eta_i\right]\end{aligned} \tag{2.3.20}$$

For a well-posed Lyapunov function from Equation (2.3.14), it follows

$$g_i(\tilde{\theta}_i) \geq 0 \tag{2.3.21}$$

when

$$\hat{\theta}_i = \Theta_{max} \tag{2.3.22}$$

Also, from Equation (2.3.17) it follows that,

$$g_i(\tilde{\theta}_i)v_i \leq 0, \quad \hat{\theta}_i = \Theta_{max} \tag{2.3.23}$$

using the same methodology

$$g_i(\tilde{\theta}_i)v_i \leq 0, \quad \hat{\theta}_i = 0 \tag{2.3.24}$$

it can be verified that

$$g_i(\tilde{\theta}_i)v_i = 0, \quad \hat{\theta}_i \in (0, \Theta_{max}) \tag{2.3.25}$$

since $v_i = 0$ when $0 < \hat{\theta}_i < \Theta_{max}$, then from Equation (2.3.13)

$$\hat{\theta}_i \in [0, \Theta_{max}], \quad \forall i = 0, \cdots, N-1, \quad and \quad t \geq 0 \tag{2.3.26}$$

combining Equation (2.3.23), (2.3.24), (2.3.25) and (2.3.26)

$$g_i(\tilde{\theta}_i)v_i \leq 0 \quad \forall i = 1, \cdots, N-1 \tag{2.3.27}$$

hence

$$\sum_{i=1}^{N-1} g_i(\tilde{\theta}_i)v_i \leq 0 \tag{2.3.28}$$

Also, using the same methodology, it can be verified that

$$\sum_{i=0}^{N-1} \theta^i(\hat{\zeta}_i - \zeta)w_i \leq 0 \tag{2.3.29}$$

Combining Equation (2.3.20), (2.3.28) and (2.3.29) it follows

$$\dot{V} \leq -\alpha\tilde{y}^2 + \tilde{y}\left[\phi_0 - (m(y,u,\theta) + r(y,u,\theta)\zeta) + \sum_{i=1}^{N-1} g_i(\tilde{\theta}_i)\phi_i + \sum_{i=0}^{N-1} \theta^i\tilde{\zeta}_i\eta_i\right] \tag{2.3.30}$$

Rearranging Equation (2.3.5) to

$$-A_r\eta = C_r \tag{2.3.31}$$

that implies for any θ_j, where $j = 1, \cdots, N-1$

$$r(y,u,\theta_j) = -\sum_{i=0}^{N-1} \theta_j\eta_i \tag{2.3.32}$$

then

$$r(y,u,\theta)\zeta = -\sum_{i=0}^{N-1}\theta_j\eta_i\zeta \quad \forall\theta\in\Theta,\zeta\in[-\zeta_{max},\zeta_{max}] \tag{2.3.33}$$

Also, it can be verified for the following equation

$$A_m\Phi = C_m - A_rC_\eta \tag{2.3.34}$$

in Equation (2.3.4) and from the definition of the matrix A_m in (2.3.10), and $\forall j = 1.\cdots.N-1$ that implies

$$m\left(y,u,\theta\right) + \sum_{i=0}^{N-1}\theta_j\eta_i\hat{\zeta}_i + \sum_{i=1}^{N-1}g_i(\tilde{\theta}_i)\phi_i - \phi_0 = 0 \tag{2.3.35}$$

Combining Equation (2.3.33) and (2.3.35) it follows

$$\begin{aligned}&\phi_0 - (m(y,u,\theta) + r(y,u,\theta)\zeta) + \sum_{i=1}^{N-1}g_i\left(\tilde{\theta}_i\right)\phi_i + \sum_{i=0}^{N-1}\theta_j\eta_i\hat{\zeta} = 0\\ &\forall\theta\in\Theta,\zeta\in[-\zeta_{max},\zeta_{max}]\end{aligned} \tag{2.3.36}$$

Combining Equation (2.3.33) and (2.3.36)

$$\dot{V} \leq -\alpha\tilde{y}^2 \tag{2.3.37}$$

which completes the proof. □

For the following lemma, $\tilde{y}$ will track y with error bound L^2.

Lemma 2

For the system in Equation (2.2.10) and the companion adaptive system in Equation (2.3.1) and (2.3.2), if $d_{max} = 0$ then

$$\int_0^\infty \tilde{y}^2 dt \leq \frac{V(0)}{\alpha} \tag{2.3.38}$$

Proof:

From *Lemma 1*

$$\int_0^\infty \dot{V} dt \leq \int_0^\infty -\alpha \tilde{y}^2 dt \tag{2.3.39}$$

Then

$$V(\infty) - V(0) \leq \int_0^\infty -\alpha \tilde{y}^2 dt \tag{2.3.40}$$

Which implies

$$\int_0^\infty \alpha \tilde{y}^2 dt \leq V(0) - V(\infty) \tag{2.3.41}$$

Since

$$V(t) \geq 0, \quad \forall t \geq 0 \tag{2.3.42}$$

It follows from Equation (2.3.41) that Equation (2.3.38) holds, which completes the proof. □

2.3.2 Multiple Region Law

Let $\hat{f}_i$ defined as

$$\begin{aligned} \widehat{f}_i &= m_i(y,u,\theta_i) + r_i(y,u,\theta_i)(\hat{\omega}_i - \bar{\omega}_i) \\ i &= 1, \cdots, N \end{aligned} \tag{2.3.43}$$

Where the initial estimates of $\hat{\omega}_i$ are defined as

$$\widehat{\omega}_i(0) = \frac{\underline{\Omega}_i + \overline{\Omega}_i}{2} \tag{2.3.44}$$

where

$$\hat{\omega}_i \in \left[\underline{\Omega}_i, \overline{\Omega}_i\right] \tag{2.3.45}$$

In the companion adaptive system, from both *Lemma 1* and Barbalat's Lemma, $\tilde{y}$ approaches zero as time goes to infinity, which means $\phi_0 \to f(y,u,\theta)$, then the update law for $\hat{\omega}$ is giving as

$$\dot{\hat{\omega}}_i = \begin{cases} 0 & if \quad \hat{\omega}_i > \overline{\Omega}_i \\ 0 & if \quad \hat{\omega}_i < \underline{\Omega}_i \\ \left(\phi_0 - \hat{f}_i\right) r_i & otherwise \end{cases} \tag{2.3.46}$$

As it can be seen in the update law for $\hat{\omega}_i$ is based on the proportional integral (PI) method. However, the projection in the integrator will enable one of the $\hat{f}_i$ to approach the value of ϕ_0 and for other $\hat{f}_i$ there will always be error which means $\hat{f}_i \to \phi_0$

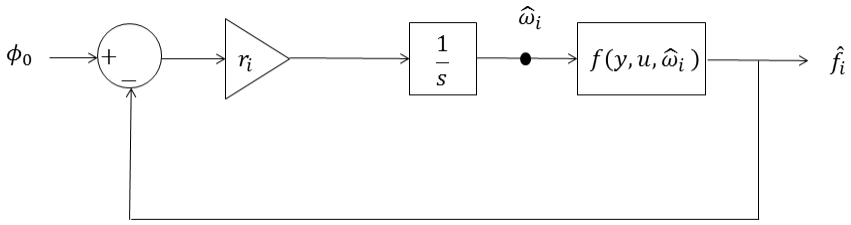

FIGURE 2.3.1: Multiple region law block diagram

2.4 persistence of excitation Condition

In this section a persistent-of-excitation condition is introduce based on [26] to guarantee the parameter convergence for the CPAE.

Definition 1 of PE:

Problem formulation in Equation (2.2.10) under assumptions 1 and 2, and for the ideal case when $d_{max} = 0$. For y, u has a persistent excitation condition if in any time t there exists a time constant T, error ϵ and a time instant t_1 where $t_1 \in [t, t + \mathrm{T}]$ such

$$\left| f\left(y(t_1), u(t_1), \hat{\theta}\right) - f\left(y(t_1), u(t_1), \theta\right) \right| \geq \varepsilon \min_{\theta \in \Theta} \left\| \hat{\theta} - \theta \right\|, \forall \hat{\theta} \in \Omega \tag{2.4.1}$$

Function f here is the piecewise linear function, and θ is the true small region where ω belongs to.

Theorem 1

If u is subject to PE then

$$\lim_{t\to\infty} \phi_0(t) = f\left(u, y, \theta\right) \tag{2.4.2}$$

Proof:

Recall Barbalats Lemma:

Let $f : \Re \to \Re$ *be a uniformly continuous function on* $[0, \infty)$ *and assume* $\lim_{t\to\infty} \int_0^t f\left(\tau\right) d\tau$ *exists, then* $\lim_{t\to\infty} f\left(t\right) = 0$

Let $f(t) = \tilde{y}^2$ which is positive. From *Lemma 2* it can be seen that

$$\lim_{t\to\infty} \int_0^\tau \tilde{y}^2(t) d\tau \tag{2.4.3}$$

exists, since $f(t) = \tilde{y}^2(t)$ is uniformly continuous then

$$\lim_{t\to\infty} \tilde{y}^2(t) = 0 \tag{2.4.4}$$

As a result,

$$\lim_{t\to\infty} \tilde{y}(t) = 0 \tag{2.4.5}$$

From Equation (2.4.5)

$$\lim_{t\to\infty} \tilde{y}(t) = 0 \tag{2.4.6}$$

Knowing $\dot{y}$ is uniformly continuous, it follows from Barbalat's lemma

$$\lim_{t\to\infty} \dot{\tilde{y}}(t) = 0 \tag{2.4.7}$$

From Equation (2.3.19) $\dot{\tilde{y}} = \alpha\tilde{y} + \phi_0 - (m(y,u,\omega) + r(y,u,\omega)\zeta)$

It can be seen that

$$\lim_{t\to\infty} \phi_0(t) = f\,(u,y,\theta) \tag{2.4.8}$$

Which proves Theorem 1. □

Theorem 2

Case 1: If ω belongs to i^{th} region, then

$$\lim_{t\to\infty} \hat{\omega}_i(t) = \omega \tag{2.4.9}$$

Proof: First, the initial value of $\hat{\omega}_i$ in Equation (2.3.44) and the projection in the multiple region law is $\hat{\omega}_i > \overline{\Omega}_i, \quad \hat{\omega}_i < \underline{\Omega}_i$ in Equation (2.3.46). Where $\hat{\omega}_i$ belongs to the i^{th} region. Then from the update law in Equation (2.3.46),

$$\lim_{t\to\infty} \hat{f}_i(u,y,\hat{\omega}_i) = \phi_0 \tag{2.4.10}$$

It follows from Theorem 1 that

$$\lim_{t\to\infty} \hat{f}_i(u,y,\hat{\omega}_i) = f\,(u,y,\theta) \tag{2.4.11}$$

Then this proof will be done by contradiction, assume Equation (2.4.9) is not true, then

$$\lim_{t\to\infty} \hat{\omega}_i(t) = \quad does \quad not \quad exist \tag{2.4.12}$$

From the PE condition

$$|f_i(y(t_1), u(t_1), \omega_i) - f\left(y(t_1), u(t_1), \omega\right)| \geq \varepsilon \min_{\theta \in \Theta} \left\| \hat{\theta} - \theta \right\|, \forall \hat{\theta} \in \Omega \tag{2.4.13}$$

Equation (2.4.13) clearly contradicts the statements in Equation (2.4.11). Therefore, Equation (2.4.12) is not true. As a result,

$$\lim_{t \to \infty} \hat{\omega}_i(t) = \omega \tag{2.4.14}$$

Which proves case 1. □

Case 2: If ω does not belong to i^{th} region, then

$$\lim_{t \to \infty} \hat{\omega}_i(t) = \ \textit{does} \quad \textit{not} \quad \textit{exist} \tag{2.4.15}$$

Proof: From the initial value of $\hat{\omega}_i$ in Equation (2.3.44) and the projection in the multiple region law $\hat{\omega}_i > \overline{\Omega}_i, \ \ \hat{\omega}_i < \underline{\Omega}_i$ in Equation (2.3.46), where $\hat{\omega}_i$ belongs to the i^{th} region. Let ω belong to the j^{th} region where i and j regions are mutually exclusive, then because of the projection there will be always an error between $\hat{f}_i(y, u, \hat{\omega}_i)$ and $f_i(y, u, \omega)$ which means

$$\lim_{t \to \infty} \hat{\omega}_i(t) = \ \textit{does} \quad \textit{not} \quad \textit{exist} \tag{2.4.16}$$

Which proves Theorem 3. □

2.5 Simulation Results

In this section, the simulation will show the performance of the CPAE. Consider the following system

$$f(y,u,\omega) = \begin{cases} u^3\omega^3 - y^3 & \omega \in [0, 2.5) \\ u^3 + (\omega^2 - 2.5)u - y^3 & \omega \in [2.5, 5) \\ -u^3\omega^3 - y^3 & \omega \in [-2.5, 0) \\ -u^3 + (\omega^2 - 2.5)u - y^3 & \omega \in (-2.5, -5] \end{cases} \tag{2.5.1}$$

$$\Omega = [-5, 5]$$

$$u = \sin(0.5t)$$

For this example Ω was divided into 4 regions, $N = 4$. Figure (2.5.1) shows the system response y and the estimated response $\hat{y}$

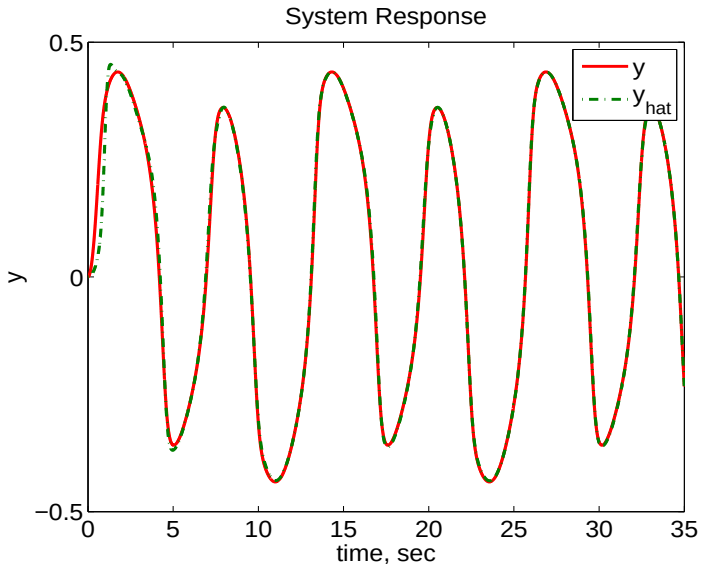

FIGURE 2.5.1: Trajectory of y and $\hat{y}$

Furthermore, from Equation (2.3.19) the phi error e can be defined as

$$e = \phi_0 - (m(y, u, \omega) + r(y, u, \omega)\zeta) \tag{2.5.2}$$

the error e can be seen in Figure (2.5.2)

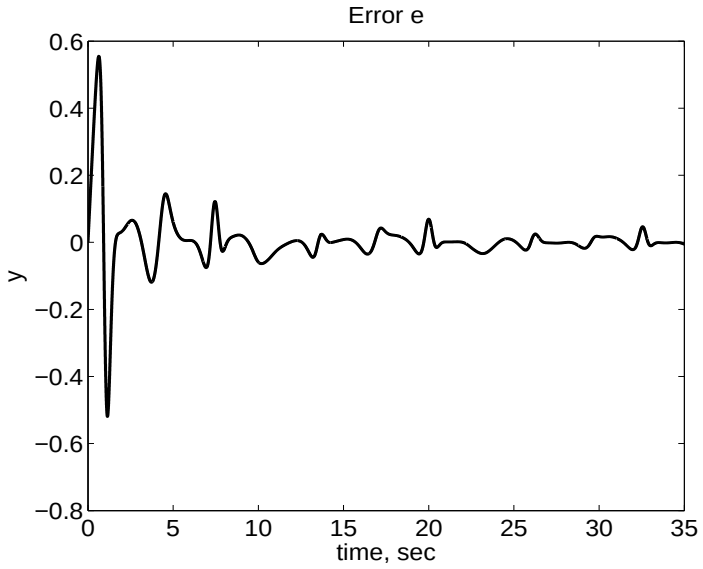

FIGURE 2.5.2: Phi (estimation) error e

The Lyapunov function and its derivative can be seen in Figure (2.5.3) and (2.5.4) respectively. From those figures, note that for a piece-wise continuous parameter, the Lyapunov function is non-increasing and Barbalat's lemma concludes that the derivative of the Lyapunov function approaches zero as time goes to infinity.

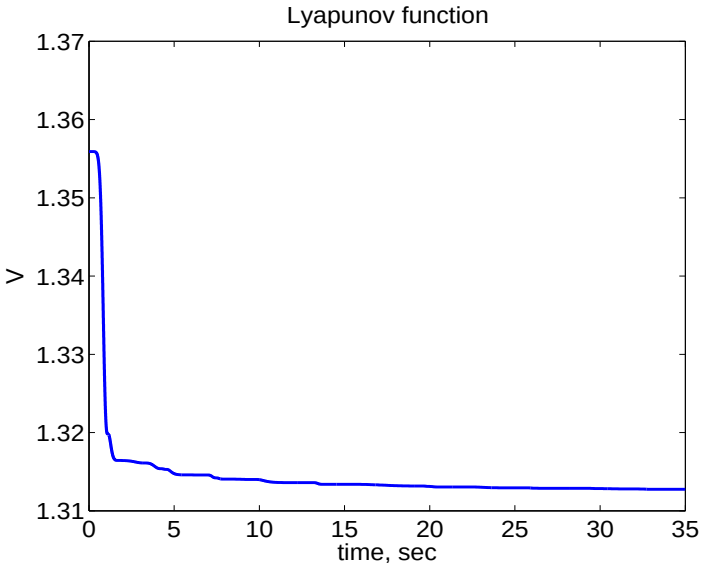

FIGURE 2.5.3: Lyapunov function

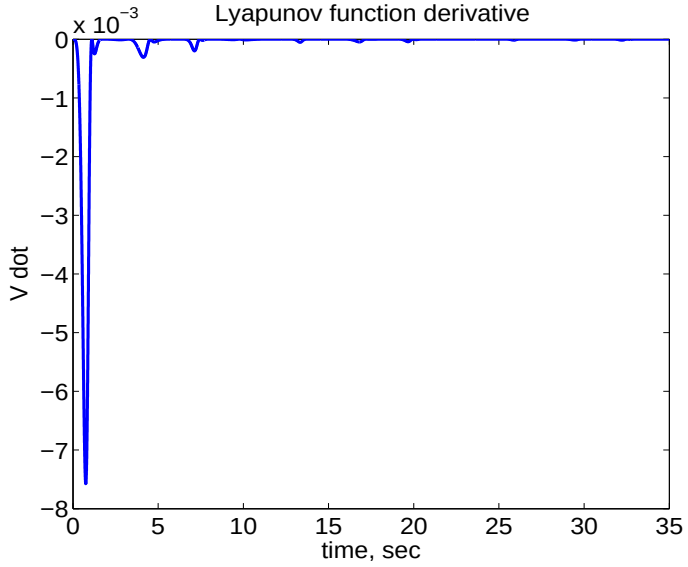

FIGURE 2.5.4: The derivative of the Lyapunov function

Lastly, the multiple region law will have N estimates for ω, in this case 4 estimates are calculated and it can bee seen in Figure (3.3.5). It can be seen that one value converges indicating the real value of ω while other values diverges.

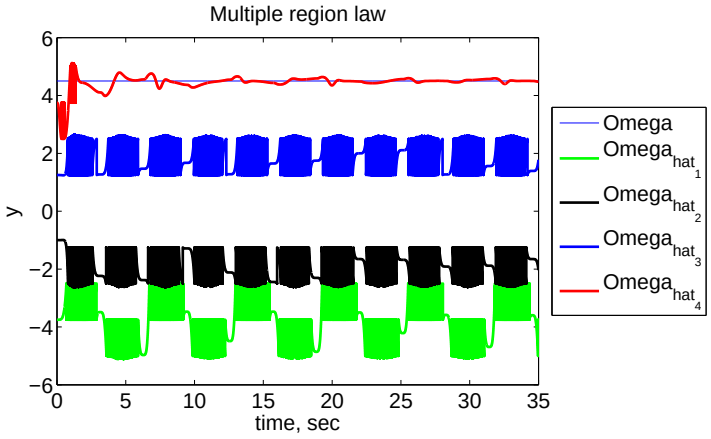

FIGURE 2.5.5: Multiple region law

2.6 Summary

This section presents the CPAE as a class for estimation of a nonlinear parameter in NLP systems. The CPAE transforms the NLP system into a function of piece-wise continuous parameters over a compact set with reasonable approximation error which can be arbitrarily reduced by increasing number of regions N. Form adaptive law which is based on non-increasing Lyapunov function, and from *Barbalats lemma* it can be concluded that the output error between the estimator and real system approaches zero as time goes to infinity, and the multiple region law estimates the unknown nonlinear parameter. Stability analysis was theoretically discussed using the Lyapunov stability. A general definition for persistent-of-excitation condition is introduced to guarantee parameter convergence. An example was introduced and the simulation results showed that the the error approaches zero, and shows the chosen Lyapunov function is a non-increasing function.

Chapter 3

Estimation of Airspeed Using CPAE

3.1 Introduction

Airspeed is an essential gain-scheduling parameter for overall control performance of an aircraft. Being able to estimate airspeed during pressure sensor (Pitot tube) failure is crucial for safety and control performance, since most of aircraft control laws for pitch, yaw and roll angles are dependent on airspeed. Unfortunate accidents occurred during Birgenair Flight 301 in 1996 and Air France Flight 447 in 2009. The primary causes of these accidents were attributed to blockages which formed within the Pitot tube, and which greatly impaired airspeed measurements. This is mainly due to the positioning of the Pitot tube because it is located towards the front of the aircraft and must be partially exposed to the outside air in order to return accurate measurements. Figure (3.1.1) highlights the positioning of the Pitot tube in relation to the aircraft.

Something to keep in mind is that the Pitot tube measures the dynamic pressure which, using Bernoulli's equation, can be used to calculate airspeed.

FIGURE 3.1.1: Example of a Pitot tube position on commercial airplane.

One way to solve this problem is to design an aircraft controller which operates independent of airspeed, or to find a way to estimate it. Since most of control laws in the aircraft depend on airspeed, the first approach seems a lot more complicated, and to find a method to estimate it is more preferable. A number of studies were conducted to estimate airspeed, using GPS velocity measurements and readings from propeller thrust to determine airspeed while statistical information used for detection [27]. Another approach is based on solving the nonlinear equation which relates the dynamics of the aircraft with the angle of attack. This equation can be solved using two approaches: the first is online solving, wherein the equation is derived from accurate knowledge of several parameters and trusted models of aircraft systems. The second approach is offline solving, wherein parameters are estimated from previous flight data [28]. However, the first approach depends on statistical data alone and

the second one considers only the angle of attack and uses the least square (LS) technique, while both of these methods are often implemented in unmanned arial vehicles (UAVs). In this section, continuous and adaptive techniques for airspeed estimation are desirable.

Parameter estimation is rigorously explored area due to its application in system identification (System ID) and modeling. LS [29] and its recursive version, the recursive least squares (RLS) [30, 31], are widely used to estimate linear parameters in both static and dynamic systems. Adaptive estimators [32] based on Lyapunov functions can be applied to estimate linear parameters in dynamic systems. Another main approach in parameter estimation is to transform the parameter estimation into an equivalent state estimation problem. Unknown parameters can be treated as extended states with derivatives equal to zero, and hence the extended state observer (ESO) [33] can be applied to estimate the state. By choosing a high observer gain, ESO can estimate time-varying parameters with the convergence rate defined by specified eigenvalues. The generalized extended state observer (GESO) [34] extends the estimated state to higher order derivatives. Sub-space System ID [35] is a statistical method to build dynamic models, can help in the parameter estimation problem. In addition, the Kalman filter (KF) [36, 37, 38] can also be applied to estimate parameters with its extended state formulation. However, many practical systems including the aircraft model considered here are nonlinear in nature. Estimation of nonlinear parameters is still an open problem in the control community. The extended (EKF) and unscented Kalman filters (UKF) [31, 39, 40, 41], can be used to handle nonlinear parameters. However, convergence is not theoretically proven and fails in some practical scenarios. There have been many efforts to estimate Nonlinearly Parameterized (NLP) systems [10, 9, 24, 12, 42, 23]. However, these results are subject to various

restrictions.

This chapter introduces an application of the continuous polynomial adaptive estimator (CPAE), where the aircraft model was treated as NLP system and the airspeed is the unknown nonlinear parameter. The rest of this chapter is organized as follow, Section 3.2 will analyze the aircraft model and transformed it into piecewise continuous parameter function and then apply the CPAE algorithm on it. Section 3.3, showes the simulation for estimation values. Lastly, Section 3.4 will summarize the main result for this chapter.

3.2 Problem Formulation

Starting from the decoupled system dynamics for longitudinal motion [43] which can be seen in equation (3.2.1)

$$\begin{bmatrix} \dot{\alpha} \\ \dot{q} \end{bmatrix} = \begin{bmatrix} \frac{Z_\alpha}{V_T} & 1 + \frac{Z_q}{V_T} \\ M_\alpha & M_q \end{bmatrix} \begin{bmatrix} \alpha \\ q \end{bmatrix} + \begin{bmatrix} \frac{Z_{\delta_e}}{V_T} \\ M_{\delta_e} \end{bmatrix} \delta_e \tag{3.2.1}$$

Here, α is the angel of attack, q is the pitch rate, δ_e in the input signal, V_T is the airspeed, Z indicates the force dimensional derivative and M indicates the moment dimensional derivative. The derivative was taken with respect to the variable shown in the subscript. Detailed explanation of those variables can be seen in equation (3.2.2)

$$
\begin{aligned}
Z_\alpha &= \frac{-1}{m}(D + \frac{\partial L}{\partial \alpha}) \\
Z_q &= \frac{-1}{m}\frac{\partial L}{\partial q} \\
M_\alpha &= \frac{1}{J_Y}\frac{\partial m_A}{\partial \alpha} \\
M_q &= \frac{1}{J_y}\frac{\partial m_A}{\partial q} \\
Z_{\delta_e} &= \frac{-\bar{q}S}{m}C_{L_{\delta_e}} \\
M_{\delta_e} &= \frac{\bar{q}S}{J_Y}C_{m_{\delta_e}} \\
m_A &= \bar{q}ScC_m
\end{aligned} \tag{3.2.2}
$$

Where J_Y is the moment of inertia, S is the cross sectional area, and c is the length of chord. In addition, the pitch moment coefficient C_m is a function of α and the Mach number $M = \frac{V_T}{a}$. Moreover, the drag, lift, and dynamic pressure can be seen equation (3.2.3)

$$
\begin{aligned}
D &= \bar{q}SC_D \\
L &= \bar{q}SC_L \\
\bar{q} &= \frac{1}{2}\rho V_T^2
\end{aligned} \tag{3.2.3}
$$

Nevertheless, both the coefficient of drag C_D and the coefficient of lift C_L are functions of, α, M and the Reynolds number. Understanding the aircraft model and how each variable is dependent of V_T, we can see from Equation (3.2.2) and Equation (3.2.3), the right hand side of Equation (3.2.1) is a function of, α, q and, V_T and it is a nonlinear function in terms of V_T. Based on that, the aircraft model was treated

as an NLP system and airspeed is the nonlinear parameter.

The aircraft model in Equation (3.2.1) can be formulated into,

$$\dot{y} = -Ay + f(y, \delta_e, V_T) \tag{3.2.4}$$

The state variable y can be defined as $y = \begin{bmatrix} \alpha & q \end{bmatrix}^T$ so $\dot{y} = \begin{bmatrix} \dot{\alpha} & \dot{q} \end{bmatrix}^T$ Where A is a known Hurwitz matrix. Then the function f becomes

$$f(y, \delta_e, V_T) = \begin{bmatrix} \frac{Z_\alpha}{V_T} & 1 + \frac{Z_q}{V_T} \\ M_\alpha & M_q \end{bmatrix} y + \begin{bmatrix} \frac{Z_{\delta_e}}{V_T} \\ M_{\delta_e} \end{bmatrix} \delta_e - Ay \tag{3.2.5}$$

The function f in equation (3.2.5) depends on y and δ_e, while Z and M are dependent on V_T nonlinearly, as a result f depends on V_T as well. Here V_T is the unknown parameter which belongs to a continuous compact set $\Omega = [\Omega_{min}, \Omega_{max}] \subset R$ and f can be approximated by a piecewise linear function over Ω, that means there exists a disturbance $d_{max} > 0$ over N regions.

$$\Omega_i = [\underline{\Omega_i}, \bar{\Omega_i}], i = 1 \dots N \tag{3.2.6}$$

$m_i = (y, \delta_e)$ and $r_i = (y, \delta_e)$ were designed such that,

$$\begin{aligned} \Omega &\subseteq \bigcup_{i=1}^{N} \Omega_i \\ |d(t)| &= |m_i(y, \delta_e, V_T) + r_i(y, \delta_e, V_T)(V_T - \overline{V}_T) - f| \leq d_{max}, V_T \in \Omega_i, \forall i = 1 \dots N \\ \overline{V_T} &= \frac{\overline{\Omega_i} + \underline{\Omega_i}}{2} \end{aligned} \tag{3.2.7}$$

Now, the unknown parameter $V_T \in \Omega$ is being mapped into a new pair of unknown

parameters $[\theta, \zeta]$ shown below,

$$\begin{aligned}
\theta &= \theta_i \text{ if } V_T \in \Omega_i \\
\theta \in \Theta &= \theta_1, \ldots \theta_i, \ldots \theta_N \\
\theta_i &= \frac{i-1}{(N-1)\Theta_{max}} \\
\zeta &= V_{T_i} - \frac{\bar{\Omega}_i + \underline{\Omega_i}}{2} \text{ if } V_T \in \Omega_i \\
\zeta &\in [-\zeta_{max}, \zeta_{max}] \\
\zeta_{max} &= max_{i=1\ldots N} \frac{\bar{\Omega}_i - \underline{\Omega_i}}{2}
\end{aligned} \tag{3.2.8}$$

where Θ_{max} is an arbitrary positive constant and $i = 1, \cdots, N$

With the unknown parameter transformation, the problem formulation for equation (3.2.4) becomes:

$$\begin{aligned}
\dot{y} &= Ay + m(y, \delta_e, \theta) + r(y, \delta_e, \theta)\zeta + d(t) \\
d(t) &\leq d_{max} \\
m(y, \delta_e, \theta) &= m_i(y, \delta_e), i = (N-1)\Theta_{max}\theta + 1 \\
r(y, \delta_e, \theta) &= r_i(y, \delta_e), i = (N-1)\Theta_{max}\theta + 1 \\
|d(t)| &= f(y, \delta_e, V_T) - m(y, \delta_e, V_T) - r(y, \delta_e, V_T)\zeta
\end{aligned} \tag{3.2.9}$$

After the problem formulation is transformed into piecewise linear function, the CPAE algorithm introduced in Section 2.3 was applied to estimate the airspeed. The following section will show simulations and results.

3.3 Simulation and Result

For simulation purposes, an academic model was developed, and the aircraft parameters were taken from *Aircraft Control and Simulation* text book [43] and it can be seen as follows:

TABLE 3.3.1: Nonlinear longitudinal small aircraft parameters

Parameter	Value
Atmospheric density	$2.377 * 10^{-3}\ \frac{slug}{ft^3}$
Weight	2300 lbs
Wing reference area, S	175 ft^2
Mean aerodynamic chord, $\bar{c}$	4.98 ft
Inertia, I_{yy}	2049 $slug - ft^2$
Thrust angel, α_T	0
Drag, C_D	$0.038 + 0.053 * C_L * C_L$
Pitch, C_m	$0.015 - 0.75 * \alpha - 0.9 * \delta_e$
Pitch damping coefficient, C_{m_q}	-12.0 per$\frac{rad}{Sec}$
Lift C_L	$0.25 + 4.58 * \alpha$

Figure 3.3.1 shows both real and estimated angle of attack.

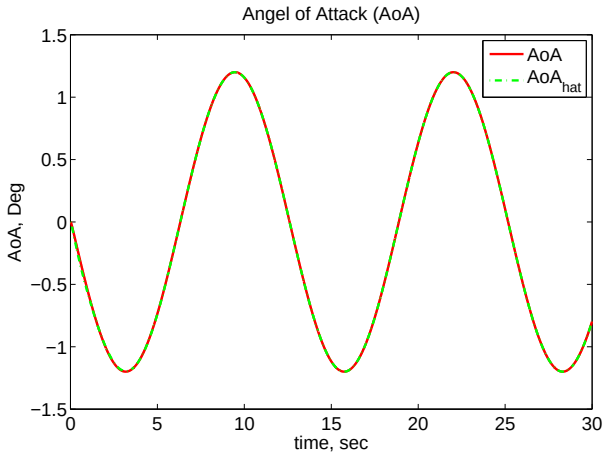

FIGURE 3.3.1: Trajectory for real and estimated angel of attack

Figure 3.3.2 shows the estimation error e which was defined in Equation (3.3.1) and shows that the error goes to zero

$$e = m(y, \delta_e, \theta) + r(y, \delta_e, \theta)\zeta - \phi_0 \tag{3.3.1}$$

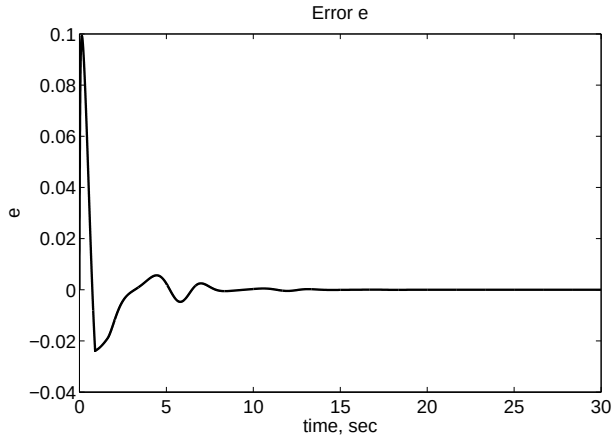

FIGURE 3.3.2: Trajectory for estimation error e

Figure 3.3.3 shows the Lyapunov function is non-increasing function indicating the parameters convergence. In addition, Figure 3.3.4 shows that the derivative of the

Lyapunov function is negative which insures that the Lyapunov function is decreasing. Moreover, it can be seen that the derivative of the Lyapunov function goes to zero as Barbalat's lemma indicates.

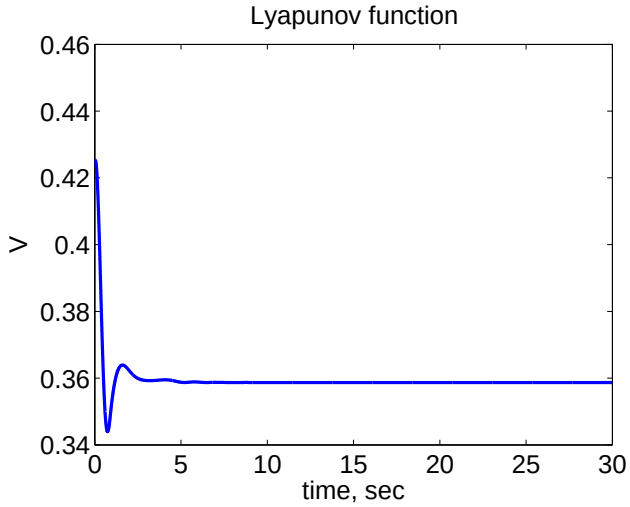

FIGURE 3.3.3: Trajectory for the Lyapunov function

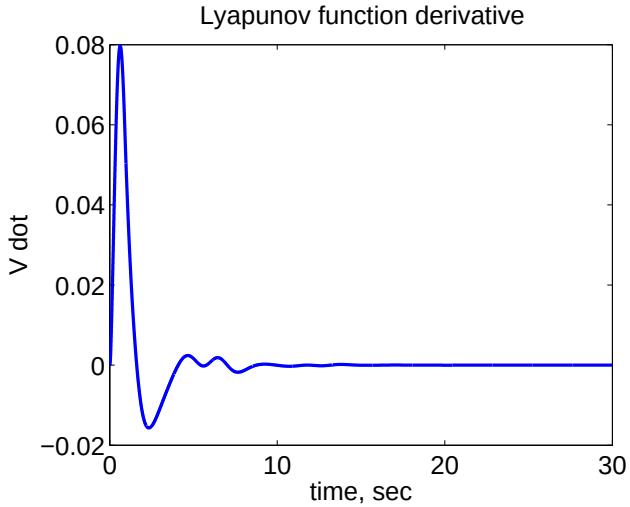

FIGURE 3.3.4: Trajectory for the derivative of the Lyapunov function

Lastly, the multiple region law will have N estimates for the real airspeed V_T, in this case 4 estimates are calculated and it can bee seen in Figure (3.3.5). It can

be seen that one value converges indicating the real value of V_T while other values diverge, here the value of the estimated airspeed is 82.9876 *knot* while the real value of airspeed was set to be 83 *knot*.

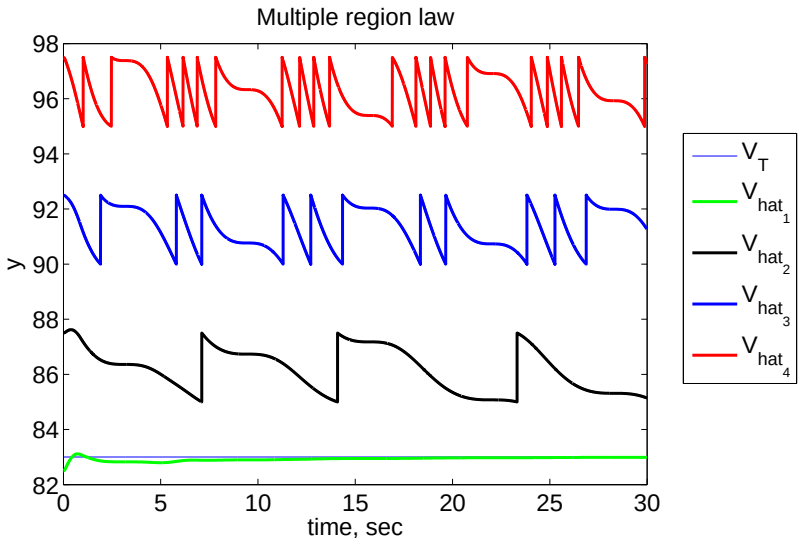

FIGURE 3.3.5: Multiple region law to estimate the real airspeed

3.4 Summary

It can be seen from Equation (3.2.1), Equation (3.2.2) and Equation 3.2.3) how airspeed is an important parameter in aircraft dynamics. These aircraft dynamics are nonlinear functions in terms of airspeed. As a result, airspeed estimation was formulated in this paper as a nonlinear parameter estimation problem. The CPAE was introduced and successfully implemented on the decoupled academic aircraft model for longitudinal motion for purpose of demonstration. This paper could be extended to include literal motion as well, and further investigation can be done on the CPAE for different cases. In the simulation, we can see one of the $\hat{f}$ converged to the real f

value which indicates the value of airspeed V_T, which made the Phi error e in Equation (3.3.1) go to zero. Being able to estimate airspeed in the presence of sensor failures will have a big impact on the aviation industry by making it safer. The analysis and simulation were conducted in hopes to implement it on real systems in order to prevent future disasters.

Chapter 4

Estimation of Airspeed in the Generic Transport Model

4.1 Introduction

Airspeed is an important parameter for aircraft control. Unlike the approach in (Chapter 3), here a practical approach for estimating airspeed based on IMU wind estimation theory and GPS measurements (William Premerlani, 2009) introduced and applied on the GTM model. Starting with

$$\vec{S} = \vec{V} + \vec{W} \tag{4.1.1}$$

See (Figure 4.1.1). Where $\vec{S}$ is a ground speed vector obtained from the GPS. $\vec{V}$ is the airspeed vector and $\vec{W}$ is wind speed vector.

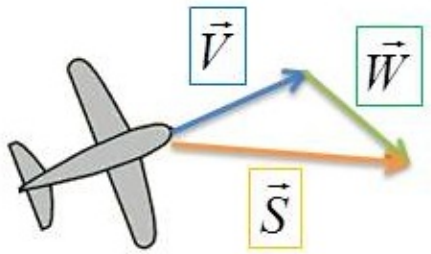

FIGURE 4.1.1: Geometric relation between airspeed, wind speed and ground speed

Several assumptions have to be made:

Assumption 1: The wind speed vector $\vec{W}$ between t_1 and t_2 remains constant, then

$$\vec{S}_2 - \vec{S}_1 = \vec{V}_2 - \vec{V}_1 \tag{4.1.2}$$

Assumption 2: When a maneuver occurred between t_1 and t_2, that results in a change in airspeed *direction* but *not magnitude*. Then

$$\vec{V} \approx V \cdot \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \vec{F} \tag{4.1.3}$$

Where V is airspeed magnitude, θ is residual yaw error in the direction cosine matrix (DCM), and $\vec{F}$ is a column of DCM which represents the fuselage. Then from equation (4.1.2) and (4.1.3), it can be seen that

$$\vec{S_2} - \vec{S_1} \approx V \cdot \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot (\vec{F_2} - \vec{F_1}) \tag{4.1.4}$$

As a result, the magnitude can be calculated as

$$\hat{V} = \frac{\left|\vec{S_2} - \vec{S_1}\right|}{\left|\vec{F_2} - \vec{F_1}\right|} \tag{4.1.5}$$

4.2 Applying IMU on the GTM Model

Before applying the IMU wind estimation theory on the GTM model, it is important to understand how the model works and how the true airspeed (TAS) is calculated. In the GTM model there are two main blocks. The first is the aircraft model, which cannot be modified, and the second is an input generator where all the control work is applied to. Basically, the input generator input wind and control commands to the aircraft model. Then the aircraft's response is fed back to the input generator. See (Figure 4.2.1)

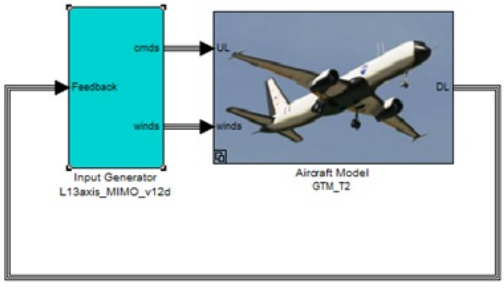

FIGURE 4.2.1: General structure of the GTM model

Inside the input generator there is a function called Winds, the main task for this function is to generate a turbulence body velocity vector and the external wind speed vector in three directions (north, east and vertical winds) see (Figure 4.2.2). The aircraft model receives the turbulence body velocity vector and subtracts it from the body axis velocity obtained from the EOM block to get the TAS vector. Then it calculates its magnitude by doting the vector by itself. Then, it multiplies it by a constant to convert it from fps to $knots$.

FIGURE 4.2.2: Winds block inside the input generator

To calculate the speed vector obtained from the IMU unit, first the Auxiliary Variables block within the aircraft model takes phi (roll), theta (pitch) and psi (yaw) obtained from the EOM block and puts it into Euler DCM form. Then it multiplies it by the body axis velocity vector and adds the wind speed vector in order to get the ground speed vector for the IMU. After that it sends it to the sensor block to adjust the resolution to make the data more realistic.

From (Equation 4.1.1), the challenge is to estimate the airspeed velocity $\vec{V}$ while the wind speed $\vec{W}$ is unknown, thus it can be seen that the IMU theory can fit into this problem. As a result, a Estimate airspeed block was created see (Figure 4.2.3) which takes the IMU velocity vector from the Auxiliary Variables block and the orientation vector from EOM block and estimates the airspeed. Result can be seen in (Figure 4.2.4)

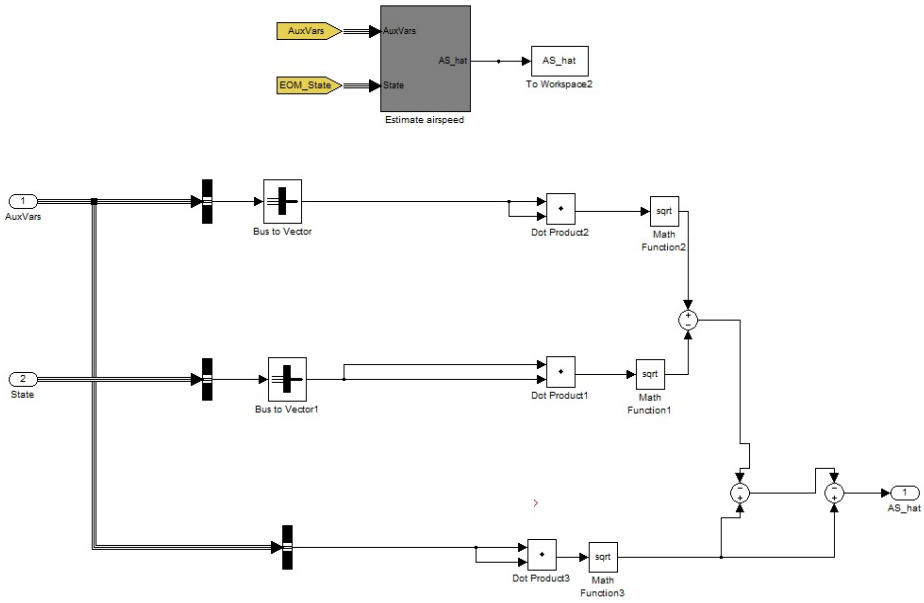

FIGURE 4.2.3: Estimate airspeed block in the input generator

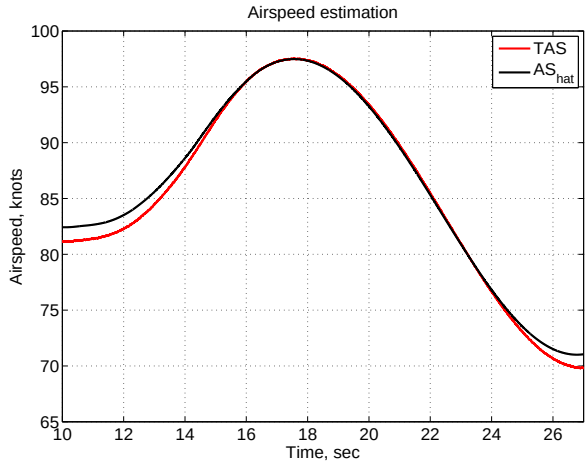

FIGURE 4.2.4: True airspeed and estimated airspeed in the GTM model

Note in (Figure 4.2.4) the estimated airspeed is ± 2 *knots* from the TAS, but it becomes better when the maneuver start around 12 *sec* and ends at 25 *sec*.

Chapter 5

Conclusions

5.1 Summary of The Main Results

This thesis introduced an extension of the polynomial adaptive estimator (PAE) presented in [25] with a new choice of Lyapunov function. Also, it combined the multiple region law with the companion adaptive system presented in [1] to come up with the the continuous polynomial adaptive estimator (CPAE). Moreover, it introduces a general definition of persistence-of-excitation (PE) condition for parameter convergence. Simulation is included to illustrate the parameter convergence using the CPAE. As an application, an academic aircraft model was developed and treated as NLP system where the airspeed is the nonlinear unknown parameter. The CPAE was applied and results showed that the estimated values converge correctly. These analyses and simulations were conducted in hopes to be implemented in real world systems in order to prevent disasters during airspeed sensor failure. Furthermore, as part of *Loss of Control Prevention through Adaptive Reconfiguration* project supported by NASA,

the IMU wind theory was applied on the *generic transport model (GTM)* to estimate the airspeed. Results showed that the estimated value of airspeed is better while the aircraft is maneuvering.

5.2 Future work

For future work, the stability of the CPAE in Section 2.3 can be extended to include approximation error as seen in Equation 2.2.3 which can be treated as disturbance. The CPAE can be extended to cover the vector form where additional investigation might be needed. Section 3.2 presents further opportunity for the CPAE to be more comprehensive and covering both longitudinal and lateral motion. In addition, in Chapter 4, the IMU method can be extended to be recursive, where several maneuvers can be done to estimate the airspeed, see (Figure 5.2.1) and recursive estimation theory such as Kalman filter can be implemented.

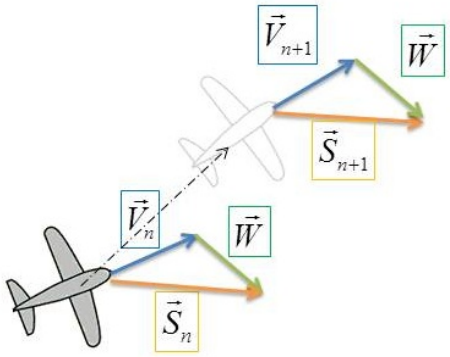

FIGURE 5.2.1: Airspeed estimation based on multiple maneuvers

Bibliography

[1] C. Cao, "Parameter Estimation and Control of Nonlinearly Parameterized System," Ph.D. Dissertation, MIT, Mechanical Engineering Dept., Cambridge, MA, 2004.

[2] H. Felemban, J. Che, C. Cao, and I. M. Gregory, "Estimation of Airspeed Using Continuous Polynomial Adaptive Estimator," 2014.

[3] O. Ghasemi, M. L. Lindsey, T. Yang, N. Nguyen, Y. Huang, and Y.-F. Jin, "Bayesian parameter estimation for nonlinear modelling of biological pathways," *BMC systems biology*, vol. 5, no. Suppl 3, p. S9, 2011.

[4] M. Ashyraliyev, Y. Fomekong-Nanfack, J. A. Kaandorp, and J. G. Blom, "Systems biology: parameter estimation for biochemical models," *FEBS journal*, vol. 276, no. 4, pp. 886–902, 2009.

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

[6] N. Hung, H. Tuan, T. Narikiyo, and P. Apkarian, "Adaptive control for nonlinearly parameterized uncertainties in robot manipulators," *Control Systems Technology, IEEE Transactions on*, vol. 16, no. 3, pp. 458–468, 2008.

[7] C. P. Vyasarayani, T. Uchida, and J. McPhee, "Nonlinear parameter identification in multibody systems using homotopy continuation," *Journal of Computational and Nonlinear Dynamics*, vol. 7, no. 1, p. 011012, 2012.

[8] A. M. Annaswamy, F. P. Skantze, and A.-P. Loh, "Adaptive control of continuous time systems with convex/concave parametrization," *Automatica*, vol. 34, no. 1, pp. 33–49, 1998.

[9] A. Kojić, A. M. Annaswamy, A.-P. Loh, and R. Lozano, "Adaptive control of a class of nonlinear systems with convex/concave parameterization," *Systems & control letters*, vol. 37, no. 5, pp. 267–274, 1999.

[10] J. Boskovic, "Adaptive control of a class of nonlinearly parameterized plants," *Automatic Control, IEEE Transactions on*, vol. 43, no. 7, pp. 930–934, 1998.

[11] A.-P. Loh, A. M. Annaswamy, and F. P. Skantze, "Adaptation in the presence of a general nonlinear parameterization: An error model approach," *Automatic Control, IEEE Transactions on*, vol. 44, no. 9, pp. 1634–1652, 1999.

[12] A. Kojić and A. M. Annaswamy, "Adaptive control of nonlinearly parameterized systems with a triangular structure," *Automatica*, vol. 38, no. 1, pp. 115–123, 2002.

[13] W. Lin and C. Qian, "Adaptive control of nonlinearly parameterized systems: the smooth feedback case," *Automatic Control, IEEE Transactions on*, vol. 47, no. 8, pp. 1249–1266, 2002.

[14] W. Lin and C. Qian, "Adaptive control of nonlinearly parameterized systems: a nonsmooth feedback framework," *Automatic Control, IEEE Transactions on*, vol. 47, no. 5, pp. 757–774, 2002.

[15] H. Han, C.-Y. Su, and Y. Stepanenko, "Adaptive control of a class of nonlinear systems with nonlinearly parameterized fuzzy approximators," *Fuzzy Systems, IEEE Transactions on*, vol. 9, no. 2, pp. 315–323, 2001.

[16] Z. Qu, R. A. Hull, and J. Wang, "Globally stabilizing adaptive control design for nonlinearly-parameterized systems," *Automatic Control, IEEE Transactions on*, vol. 51, no. 6, pp. 1073–1079, 2006.

[17] X.-J. Xie and J. Tian, "Adaptive state-feedback stabilization of high-order stochastic systems with nonlinear parameterization," *Automatica*, vol. 45, no. 1, pp. 126–133, 2009.

[18] W. Lin and R. Pongvuthithum, "Adaptive output tracking of inherently nonlinear systems with nonlinear parameterization," *Automatic Control, IEEE Transactions on*, vol. 48, no. 10, pp. 1737–1749, 2003.

[19] W. Chen, L. Jiao, R. Li, and J. Li, "Adaptive backstepping fuzzy control for nonlinearly parameterized systems with periodic disturbances," *Fuzzy Systems, IEEE Transactions on*, vol. 18, no. 4, pp. 674–685, 2010.

[20] M. Farza, M. MSaad, T. Maatoug, and M. Kamoun, "Adaptive observers for nonlinearly parameterized class of nonlinear systems," *Automatica*, vol. 45, no. 10, pp. 2292–2299, 2009.

[21] V. Adetola, D. Lehrer, and M. Guay, "Adaptive estimation in nonlinearly parameterized nonlinear dynamical systems," in *American Control Conference (ACC), 2011*. IEEE, 2011, pp. 31–36.

[22] H. F. Grip, T. A. Johansen, L. Imsland, and G.-O. Kaasa, "Parameter estimation and compensation in systems with nonlinearly parameterized perturbations," *Automatica*, vol. 46, no. 1, pp. 19–28, 2010.

[23] C. Cao and A. Annaswamy, "A polynomial adaptive estimator for nonlinearly parameterized systems," in *American Control Conference, 2004. Proceedings of the 2004*, vol. 1. IEEE, 2004, pp. 584–589.

[24] C. Cao, A. M. Annaswamy, and A. Kojic, "Parameter convergence in nonlinearly parameterized systems," *Automatic Control, IEEE Transactions on*, vol. 48, no. 3, pp. 397–412, 2003.

[25] C. Cao and A. Annaswamy, "A polynomial adaptive controller for nonlinearly parameterized systems," in *Decision and Control, 2006 45th IEEE Conference on*. IEEE, 2006, pp. 1081–1086.

[26] C. Cao and A. Annaswamy, "A hierarchical discretized-parameter polynomial adaptive estimator for non-linearly parameterized systems," *International Journal of Control*, vol. 79, no. 8, pp. 831–844, 2006.

[27] S. Hansen, M. Blanke, and J. Adrian, "Diagnosis of uav pitot tube defects using statistical change detection," 2010.

[28] M. Fravolini, M. Pastorelli, S. Pagnottelli, P. Valigi, S. Gururajan, H. Chao, and M. Napolitano, "Model-based approaches for the airspeed estimation and fault monitoring of an Unmanned Aerial Vehicle," in *Environmental Energy and Structural Monitoring Systems (EESMS), 2012 IEEE Workshop on*. IEEE, 2012, pp. 18–23.

[29] C. L. Lawson and R. J. Hanson, *Solving least squares problems*. SIAM, 1974, vol. 161.

[30] J. Cioffi and T. Kailath, "Fast, recursive-least-squares transversal filters for adaptive filtering," *Acoustics, Speech and Signal Processing, IEEE Transactions on*, vol. 32, no. 2, pp. 304–337, 1984.

[31] J. H. Lee and N. L. Ricker, "Extended Kalman filter based nonlinear model predictive control," *Industrial & Engineering Chemistry Research*, vol. 33, no. 6, pp. 1530–1541, 1994.

[32] G. Luders and K. Narendra, "An adaptive observer and identifier for a linear system," *Automatic Control, IEEE Transactions on*, vol. 18, no. 5, pp. 496–499, 1973.

[33] J. Han, "The extended state observer for a class of uncertain systems," *Control and Decision,*, vol. 10, no. 1, pp. 85–88, 1995.

[34] R. Miklosovic, A. Radke, and Z. Gao, "Discrete implementation and generalization of the extended state observer," in *American Control Conference, 2006.* IEEE, 2006, pp. 6–pp.

[35] P. Van Overschee and B. De Moor, "A unifying theorem for three subspace system identification algorithms," *Automatica*, vol. 31, no. 12, pp. 1853–1864, 1995.

[36] R. E. Kalman *et al.*, "A new approach to linear filtering and prediction problems," *Journal of basic Engineering*, vol. 82, no. 1, pp. 35–45, 1960.

[37] R. E. Kalman and R. S. Bucy, "New results in linear filtering and prediction theory," *Journal of Basic Engineering*, vol. 83, no. 3, pp. 95–108, 1961.

[38] G. Welch and G. Bishop, "An introduction to the Kalman filter," 1995.

[39] S. J. Julier and J. K. Uhlmann, "New extension of the Kalman filter to nonlinear systems," in *AeroSense'97.* International Society for Optics and Photonics, 1997, pp. 182–193.

[40] E. A. Wan and R. Van Der Merwe, "The unscented Kalman filter for nonlinear estimation," in *Adaptive Systems for Signal Processing, Communications, and Control Symposium 2000. AS-SPCC. The IEEE 2000.* IEEE, 2000, pp. 153–158.

[41] S. Sarkka, "On unscented Kalman filtering for state estimation of continuous-time nonlinear systems," *Automatic Control, IEEE Transactions on*, vol. 52, no. 9, pp. 1631–1641, 2007.

[42] M. Netto, A. Annaswamy, R. Ortega, and P. Moya, "Adaptive control of a class of non-linearly parametrized systems using convexification," *International Journal of Control*, vol. 73, no. 14, pp. 1312–1321, 2000.

[43] B. Stevens and F. Lewis, "Aircraft Control and Simulation." Wiley and Sons, Inc, 2003, pp. 140–263.

Appendices

Appendix A

CPAE Main Program

```
clear all

close all

clc


beg_t=cputime;

W=f(2,0,0,0); %Set the compact set

thes=f(3,0,0,0); %Set the real value

t_step=0.001; %time step
```

```
t_stop=30; %Simulation time

T_hist=0; %Initialize time

t_hist=0; %Initialize time

con_i=1; %Set counter

N=4; %Number of region

%make The_max smaller, estimation faster and more accurate adn the_max bigger,
%less possible that inv(A) singular

The_max=1;

Theta=linspace(0,The_max,N)';

kk=1/(The_max+0.2);

om_interval=(W(2)-W(1))/N;

om_max=om_interval/2;

m=zeros(N,1);
r=zeros(N,1);
Ar=zeros(N,N);
Cr=zeros(N,1);
Am=zeros(N,N);
```

```
Cm=zeros(N,1);
Crr=zeros(N,1);
Phi=zeros(N,1);
Eta=zeros(N,1);
wh=ones(N-1,1)*The_max/2;
omh=zeros(N,1);
alpha=0.2;
t=0;
y=0;
n=noise(1);
yn=y+n;
yh=0;

%estimated value
h_xest=zeros(N,1);
xx_est=[85; 92; 95; 100];

while t<=t_stop

    u=uu(t);

    %get true unknown parameters
    TrueInd=ceil((thes-W(1))/om_interval);
    w_star=Theta(TrueInd);
    om_star=thes-W(1)-(TrueInd-1)*om_interval-om_max;

    %using y_bar to calculate Phi instead of yn
    y_bar=yh;
```

```
%function approximation
%calculate m and r
for i=1:N

    tau_m=W(1)+(i-1+1/2)*om_interval;
    tau_r1=W(1)+(i-1)*om_interval;
    tau_r2=W(1)+(i)*om_interval;
    m(i)=f(1,y,u,tau_m);
    r(i)=(f(1,y,u,tau_r2)-f(1,y,u,tau_r1))/om_interval;

end

%calculate approximation error
a_error=0;
for f_i=W(1):0.1:W(2)
    f_y=f(1,y_bar,u,f_i);
    % approximated value
    test_w=floor((f_i-W(1))/om_interval);
    test_w=test_w+1;
    if test_w>N
        test_w=N;
    end
    test_om=(f_i-W(1)-(test_w-1-1/2)*om_interval);
    f_yh=m(test_w)+r(test_w)*test_om;
    if abs(f_yh-f_y)>a_error
        a_error=abs(f_yh-f_y);
    end
end
```

```
a_error2=f(1,y_bar,u,thes);
a_error2=a_error2-(m(TrueInd)+r(TrueInd)*om_star);

%calculate Eta
for a_i=1:N
    for a_j=1:N
        Ar(a_i,a_j)=Theta(a_i)^(a_j-1);
    end
end
for a_i=1:N
    Cr(a_i)=r(a_i);
end
Eta=-inv(Ar)*Cr;
     A(:,1)=ones(N,1);
     for p=1:N-1
        wh_temp=wh(p)*ones(N,1);
        if mod(p,2)==1
            A(:,p+1)=(wh_temp-Theta).^p;
        else
            A(:,p+1)=kk*(wh_temp-Theta).^p+(wh_temp-Theta).^(p-1);
        end
     end
for a_i=1:N
    Cm(a_i)=m(a_i);
end
for a_i=1:N
    Crr(a_i)=omh(a_i)*Eta(a_i);
end
Phi=inv(A)*(Cm-Ar*Crr);
```

```
%Multiple region approach
alg_y=Phi(1);
for a_i=1:N
f_hat=m(a_i)+(xx_est(a_i)-(W(1)+(a_i-1+1/2)*om_interval))*r(a_i);
tau_r1=W(1)+(a_i-1)*om_interval;
tau_r2=W(1)+(a_i)*om_interval;
if xx_est(a_i)<=tau_r1
xx_est(a_i)=0.5*(tau_r1+tau_r2);

elseif xx_est(a_i)>=tau_r2

xx_est(a_i)=0.5*(tau_r1+tau_r2);
else
xx_est(a_i)=xx_est(a_i)+(alg_y-f_hat)*r(a_i);
end
end

    %calculate derivative
      y_deri=-y+f(1,y,u,thes);
      yh_deri=-yh+Phi(1);
      yt=yh-yn;
      yte=yt;

    wh_deri=100*yte*Phi(2:N);
    omh_deri=100*yte*Eta;

    %record history
    if t_hist>=T_hist
```

```
    %calculate true unknown parameter estimates: alpha*yn+f(1,yn,u,x)=Phi(1)
            for i=1:N
                h_xest(i,con_i)=xx_est(i);
            end
        h_u1(con_i)=u(1);
%         h_ref(con_i)=u(2);
        h_yn(con_i)=yn;
        h_y(con_i)=y;
        h_yh(con_i)=yh;
        h_thes(con_i)=thes;
        h_phierr(con_i)=f(1,y,u,thes)-Phi(1);
        h_aerr(con_i)=a_error2;
        h_yte(con_i)=yte;
        h_t(con_i)=t;



%Calculate Lyapunov function V
V=yte^2/2;
for p=1:N-1
if mod(p,2)==1
 V=V+(wh(p)-Theta(TrueInd))^(p+1)/(p+1);
 else
 V=V+kk*(wh(p)-Theta(TrueInd))^(p+1)/(p+1)+(wh(p)-Theta(TrueInd))^p/p;
 end
 end
 for p=1:N
 V=V+Theta(TrueInd)^(p-1)*(omh(p)-om_star)^2/2;
 end
```

```
hist_V(con_i)=V;

%calculate Lyapunov function derivative Vdot
Vdot=(m(TrueInd)+r(TrueInd)*om_star)+Phi(1);
for p=1:N-1
if mod(p,2)==1
Vdot=Vdot+(wh(p)-Theta(TrueInd))^(p)*Phi(p+1);
else
Vdot=Vdot+Phi(p+1)*(kk*(wh(p)-Theta(TrueInd))^(p)+(wh(p)-Theta(TrueInd))^(p-1));
end
end
for p=1:N
Vdot=Vdot+Theta(TrueInd)^(p-1)*(omh(p)-om_star)*Eta(p);
end
Vdot=yt*Vdot-alpha*yt^2-yt*a_error2;
hist_Vdot(con_i)=Vdot;

t_hist=0;
con_i=con_i+1;
end


   %set time_stepsize
   t_vstep=t_step;
   wh_max=max(wh_deri);
   if wh_max*t_vstep>The_max/(2*N);
        t_vstep=(The_max/(2*N))/wh_max;
   end
   wh_max=max(omh_deri);
```

```
if wh_max*t_vstep>om_max/4;
     t_vstep=(om_max/4)/wh_max;
end


%time advance
t=t+t_vstep;
y=y+y_deri*t_vstep;
yh=yh+yh_deri*t_vstep;
wh=wh+wh_deri*t_vstep;
omh=omh+omh_deri*t_vstep;
t_hist=t_hist+t_vstep;

%bounded wh and omh
for i=1:N
    if omh(i)>om_max
        omh(i)=om_max;
    elseif omh(i)<-om_max
        omh(i)=-om_max;
    end
end
for i=1:N-1
    if wh(i)>The_max
        wh(i)=The_max;
    elseif wh(i)<0
        wh(i)=0;
    end
end
```

```
    n=noise(1);
    yn=y+n;

end

end_t=cputime;

run_time=end_t-beg_t


%Input function
function u=uu(t);
u=sin(.5*t);


%plotting function

figure(1)
axes('FontSize',16);
plot(h_t,h_y,'r','linewidth',2);
hold on;
plot(h_t,h_yh,'k','linewidth',2);
set(gcf,'Color',[1,1,1])
box on
title('Angel of Attack');
xlabel('time, sec');
ylabel('AoA, Deg');
legend('y','y_{hat}');
pic='result_y'
```

```
saveas(gcf,[pic])

figure(2)
axes('FontSize',16);
plot(h_t,h_phierr,'k','linewidth',2);
set(gcf,'Color',[1,1,1])
box on
title('Error e');
xlabel('time, sec');
ylabel('e');
pic='Error'
saveas(gcf,[pic])

figure(3)
axes('FontSize',16);
plot(h_t,hist_V,'b','linewidth',2);
set(gcf,'Color',[1,1,1])
box on
title('Lyapunov function');
xlabel('time, sec');
ylabel('V');
pic='Lyapunov'
saveas(gcf,[pic])

figure(4)
axes('FontSize',16);
plot(h_t,hist_Vdot,'b','linewidth',2);
set(gcf,'Color',[1,1,1])
box on
```

```
title('Lyapunov function derivative');
xlabel('time, sec');
ylabel('V dot');
pic='lyaounov_dot'
saveas(gcf,[pic])

figure(5)
axes('FontSize',16);
plot(h_t,h_thes)
hold on
plot(h_t,h_xest(1,:),'g','linewidth',2);
hold on
plot(h_t,h_xest(2,:),'k','linewidth',2);
hold on
plot(h_t,h_xest(3,:),'b','linewidth',2);
hold on
plot(h_t,h_xest(4,:),'r','linewidth',2);
set(gcf,'Color',[1,1,1])
box on
title('Multiple region law');
xlabel('time, sec');
ylabel('y');
legend('V_{T}','V_{hat_1}','V_{hat_2}','V_{hat_3}','V_{hat_4}');
pic='MRL'
saveas(gcf,[pic])
```