Development of U-model Enhanced Nonlinear Systems

Xin Liu

A thesis submitted to the University of the West of England, Bristol
for the degree of Doctor of Philosophy

Supervisors: Prof. Quan Min Zhu and Dr. Pritesh Narayan

Faculty of Environment and Technologies (FET), University of the West of England (UWE), Bristol, UK

June, 2018

Abstract

Since the first publication, the U-model methodology has progressed and evolved over the course of a decade. By using the U-model technique, researchers have proposed many different linear algorithms for the design of control systems for the nonlinear polynomial model including; adaptive control, internal control, sliding mode control, predictive control and neural network control. However, limited research has been concerned with the design and analysis of robust stability and performance of U-model based control systems.

This project firstly proposes a suitable method to analyse the robust stability of the developed U-model based pole placement control systems against uncertainty. The parameter variation is bounded, thus the robust stability margin of the closed loop system can be determined by using LMI (Linear Matrix Inequality) based robust stability analysis procedures. U-block model is defined as an input output linear closed loop model with pole assignor converted from the U-model based control system. With the bridge of U-model approach, it connects the linear state space design approach with the nonlinear polynomial model. Therefore, LMI based linear robust controller design approaches are able to design enhanced robust control system within the U-block model structure.

With such development, the first stage U-model methodology provides concise and flexible solutions for complex problems, where linear controller design methodologies are directly applied to nonlinear polynomial plant-based control system design. The next milestone work expands the U-model technique into state space control systems to establish the new framework, defined as the U-state space model, providing a generic prototype for the simplification of nonlinear state space design approaches.

The new U-state space platform provides a generalised representation of a broad range of nonlinear state space models and simplifies nonlinear control design procedures. The desired state vector (closed loop specification) is determined by the linear control design method (LQR design). The proposed U-state space control system design approach is applied to develop the controller for a nonlinear quad-rotor rotorcraft model and nonlinear inverted pendulum system. The simulation results are presented to validate the effectiveness and efficiency of the proposed U-state space approach and stabilise with satisfied performance.

Acknowledgements

Firstly, I wish to express my gratitude to my wife for her constant support and love.

The writing of this dissertation has been one of the most significant academic challenges I have ever had to face. This project would not have been completed without the support, patience and guidance of the following people.

I gratefully acknowledge my supervisor team, Professor Quanmin Zhu and Doctor Pritesh Narayan, who have carefully read all my manuscript and provided me with valuable comments and discussions regarding various aspects of this study. I would like to thank them for their supervision throughout the period of this research work. I also would like to thank Professor Yufeng Yao who is monitoring my research progress every year and providing useful suggestions for my research. Special thanks to my colleagues and friends, Doctor Yuan Peng, Doctor Zhiyan Wang, *et al*, for their help and useful discussions on many occasions.

Many thanks to Doctor Catherine Hobbs, Doctor Alistair Clark, Doctor Mokhtar Nibouche, Doctor Sabir Ghauri, Doctor Steve Wright and many other members of EDM Department and FET faculty.

Contents

Chapter 1 Introduction..1
Introduction...1
1.1 Nonlinear control system design…………………………………………………1
1.2 Research motivation……………………………………………………………..5
1.3 Research questions………………………………………………………………8
1.4 The aims and objectives of the project…………………………………………..8
1.5 Contributions……………………………………………………………………10
1.6 Outline of the thesis……………………………………………………………11

Chapter 2 Overview of U-model based Control System Design………………………15
2.1 Introduction……………………………………………………………………15
2.2 Description of U-model structure……………………………………………...17
2.3 U-model based pole placement control system design………………………………20
2.4 Other U-model based control systems…………………………………………23
2.5 Summaries……………………………………………………………………27

Chapter 3 U-Model based Robust Control System Design and Analysis……29
3.1 Introduction…………………………………………………………………..29
3.2 U-block model……………………………………………………………......31
3.3 Robustness analysis…………………………………………………………....33
3.3.1 LMI preliminaries…………………………………………………………....33
3.3.2 LMI based robust analysis…………………………………………………....35
3.4 U-block model based LMI robust controller design……………………………37
3.5 Case studies……………………………………………………………………42
3.6 Conclusions……………………………………………………………………50

Chapter 4 U-State Space Control System Design and Analysis…………………52
4.1 Introduction……………………………………………………………………52

4.1.1 Nonlinear polynomial control……………………………………………....53
4.1.2 Feedback linearisation…………………………………………………....53
4.2 U-state space model………………………………………………………………57
4.3 U-state space control system design…………………………………………59
4.4 Stability analysis…………………………………………………………………63
4.5 Case studies……………………………………………………………………67
4.6 Conclusions……………………………………………………………………75

Chapter 5 U-State Space Control Systems Design for a Quad-rotor……76
5.1 Introduction……………………………………………………………………76
5.1.1 Quadrotor configuration and requirements………………………………....76
5.1.2 Quadrotor control……………………………………………………....78
5.2 The nonlinear quad-rotor model……………………………………………..81
5.2.1 Quadrotor model preliminaries……………………………………....81
5.2.2 Quadrotor kinematic equations……………………………………....83
5.3 U-state space control system design for quad-rotor dynamic model………………..88
5.4 Case studies……………………………………………………………………91
5.4.1 Quadrotor kinematic equations……………………………………....92
5.4.2 Linear model of quadrotor and control…………………………………....94
5.4.3 Nonlinear quadrotor control……………………………………………....98
5.5 Conclusions……………………………………………………………………102

Chapter 6 U-State Space Enhanced Control of Inverted Pendulum………………103
6.1 Introduction……………………………………………………………………103
6.1.1 Overview of inverted pendulum control systems………………………....103
6.1.2 Inverted pendulum control strategies……………………………………...112
6.2 Inverted Pendulum System…………………………………………………………104
6.3 U-State Space Control System Design for Linear System…………………………115
6.3.1 Linear controller design……………………………………………....115
6.3.2 Case study……………………………………………………………....117
6.4 U-State Space Control System Design for Nonlinear System………………………122
6.4.1 Nonlinear control design…………………………………………………...122

6.4.2 Case study..123
6.5 Conclusions..127

Chapter 7 Conclusions and Further Work...128
7.1 Conclusions..128
7.2 List of publications based on this study...130
7.3 Further work...131

Appendix A Procedures of Linearisation for Quad-rotor Dynamic Model..133

Appendix B Program of U-model based Control System Design and Analysis..136

References..149

List of Figures

Figure 1.1 The diagram of project aims and objectives

Figure 1.2 Flowchart of the chapters

Figure 2.1 A general linear pole placement control system

Figure 2.2 Block diagram of U-model based pole placement control system

Figure 3.1 U-Block model

Figure 3.2 Block diagram of enhanced LMI control system

Figure 3.3 Performance of Hammerstein model in case I

Figure 3.4 Performance of Hammerstein model in case II

Figure 3.5 Performance of Hammerstein model in case II

Figure 3.6 System output after internal parameter changed – case I

Figure 3.7 System output after internal parameter changed – case II (No robust controller)

Figure 3.8 System output after internal parameter changed – case II (With robust controller)

Figure 4.1 Feedback linearisation block diagram

Figure 4.2 U-state space control system structure

Figure 4.3 Response of state variable $x_1(t)$

Figure 4.4 Response of state variable $x_2(t)$

Figure 4.5 Controller output $u(t)$

Figure 4.6 Response of state variable $x_1(t)$

Figure 4.7 Response of state variable $x_2(t)$

Figure 4.8 Controller output $u(t)$

Figure 4.9 Response of sate variable $x_1(k)$ velocity

Figure 4.10 Response of sate variable $x_2(k)$ angle of attack

Figure 4.11 Response of sate variable $x_3(k)$ pitch angle

Figure 4.12 Response of sate variable $x_4(k)$ pitch rate

Figure 5.1 Quad-Rotor configuration

Figure 5.2 The standard model of quadrotor

Figure 5.3 Motion of the quadrotor

Figure 5.4 block diagram of U-state space design

Figure 5.5 Control input Δu

Figure 5.6 Resultant in position

Figure 5.7 Block diagram of U-state space nonlinear control

Figure 5.8 Control inputs u

Figure 5.9 Resultant in position

Figure 6.1 Basic rocket motion

Figure 6.2 Sun synchronous orbit

Figure 6.3 Tower Crane

Figure 6.4 Segway model

Figure 6.5 PID controllers for the inverted pendulum

Figure 6.6 Standard LQR control system for the inverted pendulum

Figure 6.7 Sliding mode controller for the inverted pendulum

Figure 6.8 Block diagram of feedback linearisation

Figure 6.9 Schematic diagram of inverted pendulum

Figure 6.10 Force analysis of the cart

Figure 6.11 Force analysis of pendulum

Figure 6.12 Block diagram of U-state space feedback control

Figure 6.13 Response of cart position and speed

Figure 6.14 Response of pendulum angle and angular speed

Figure 6.15 Control input *u*

Figure 6.16 Response of cart position and speed (nonlinear model)

Figure 6.17 Response of pendulum angle and angular speed (nonlinear model)

Figure 6.18 Control input cart position and angle (nonlinear model)

List of Tables

Table 4.1 Trimmed equilibrium for $X_{cg} = 0.3\overline{c}$

Table 5.1 Parameters of the quad-rotor model

Table 6.1 Abbreviation of inverted pendulum

Table 6.2 Parameters of inverted pendulum

Chapter 1

Introduction

The core objective of design in control engineering is invariable in-order to develop a controlled system that best matches required performance or design specifications (Zhu et al., 2016). The most recognised and broadly classification of the usual control system design procedures are grouped into the linear or nonlinear behaviour of the dynamic plant (system or process). Linear control is a mature subject with a variety of powerful methodologies and a long history of successful industrial applications (Slotine and Li, 1991). Comparatively, it is a more difficult subject of nonlinear control, which is defined as the design and analysis of target control systems which contain at least one nonlinear component. Generalisation of the design methodology for nonlinear control systems has been a popular and challenging topic in research and applications including such broad applications as aircraft and spacecraft control, robotics and process control.

1.1 Nonlinear Feedback Control Systems

In a wide range of practical industrial fields, the occurrence of various control problems is characterised by essential nonlinearity. Designing a controller for nonlinear dynamic systems, control engineers were forced to consider on how to best deal with the nonlinear characteristic of the dynamic plants. Thus, the most difficult issue is to establish a general control-oriented model prototype to represent the dynamic behaviour of nonlinear plants, which requires transformation of the original model into a concise, flexible and precise expression.

For nonlinear control system design, many mature linear control system design approaches cannot be directly applied to nonlinear control system design. Linearisation is one of the most frequently used methodologies to describe the operation of physical systems around its operating points and approximate the linear behaviour of the nonlinear system, but inadequate or inaccurate errors will also appear during the analysis of the linearised system behaviour. There are several linearisation techniques including piecewise linearisation, pointwise linearisation, feedback linearisation and back stepping.

1.1.1 Piecewise linearisation

According to the characteristic of the nonlinear dynamic system, piecewise linearisation divides the curve of system's input and output relationship description into some intervals to linearise every interval into straight line approximation (Leenaerts and Bokhoven, 1998). As a result, each interval of the original nonlinear system can be regarded as an equivalent approximate linear system and applied to linear control system design approaches. Based on this point, researchers expect to use mature linear control theory and methodology on analysing and synthesising linearisation problems. However, this linearisation method is only valid in the neighbourhood area of the operating point (Oktem, 2005).

The piecewise linearised model will change immediately according to the corresponding operating point variation in the time varying system. It can be said that piecewise linearisation method can be simply applied to slow time-varying systems, whose parameter variations are guaranteed. Although the characteristic nonlinearity exists in this system, the existing linear control design and analysis methods can be directly used to sort out such linearised problems around the limited margin. On the contrary, the performance of piecewise linearisation control systems cannot be guaranteed on the fast time varying behaviour system, because large time margin of the interval will generate dynamic shift. With regards to piecewise, the linear controller design method can be immediately applied to the several linearised subsystems of the nonlinear system. Piecewise linearisation will obviously process massive numerical calculation procedures on more pieces of linearised subsystems when the target system has stronger nonlinear characteristics.

1.1.2 Pointwise linearisation

The neural network is one of the most widely used numerical modelling methods. It not only has the high fitting ability for mapping any complex nonlinear dynamic behaviour but also easily implements the related program on computers for multifunction. Based on these advantages, linearised neural network approach is proposed to approximate the nonlinear plant dynamic behaviour around the operating point by using the linear model (Hagan, et al. 2002; Zhu et al. 1999).

The parameters of the designed controller can be updated under adaptive rules using the output of neural network, so that these errors caused by linearisation are compensated for such online study performance of neural network. In other words, the neural network output determines the time varying based controller output. This approach has a variety of advantages such as strong robustness, fast online updating ability and accurate nonlinear mapping. For such powerful functions, this method is regarded as a universal approximation approach.

Based on neural network, an approximate linearisation for nonlinear systems is proposed to design a network approximator for involutive equation integration however it has a satisfied or unsatisfied integrability condition (Pei and Zhou, 1998), which has fewer restrictions and can be widely applicable. However, the system uncertainties beyond consideration slow down the efficiency of the training speed of the neural network algorithm. Dynamic output feedback linearisation based neural network modelling and control approach is proposed for ANARX (Additive Nonlinear Auto-Regressive eXogenous) structure (Petlenkov, 2007). The linear model approximation is based on feedback linearisation and the controller design employs simple construction and smarter neural network.

1.1.3 Feedback Linearisation

The central idea of feedback linearisation is the conversion of an equivalent linear model from the original nonlinear dynamic model by coordinate transform (Slotine and Li, 1991; Zhu et al., 2016). The input output feedback linearisation is defined according to the relative degree (generally smaller than dynamic order) of the nonlinear system (Khalil, 2002), which obtains a direct and simple relation between the system output y and the control input u so that successfully cancel the nonlinear dynamics in the closed loop by coordinate transform and output feedback.

This method is straightforwardly achieved in industry as it only requires the measurement of the data of system output and control input. However, the limitations of this method have the following two points. On one hand, the nonlinear dynamic model of the controlled object is required accurate and precise description for it is a model dependency method. On the other hand, the zero dynamics of the primary nonlinear plant must be stable due to the precondition that only the input and output information of the system is used for calculation (Haddad and Chellaboina, 2008).

In terms of discrete-time nonlinear system, Lee and Marcus (1987) propose to implement the related feedback linearisation method, which also can be used to linearise the MIMO process (Kravaris and Soroush, 1990). Commonly, feedback linearisation is not employed in the processing industries since the standard controlled objects (processes) in such industries tend to contain strong nonlinearity. It is hard to obtain accurate and precise model expressions based on state space description. In general, the relation equation between control input and system output can be obtained by various system identification approaches.

1.1.4 Backstepping linearisation

Backstepping is a technology by recursive solutions for stabilising a strict feedback nonlinear system against uncertainty (Wang and Wang, 2009). Combined with feedback linearisation, this approach designs a sequence of 'virtual' systems whose relative degree is one by following backstepping design principles. The last virtual output is used for linear closed loop feedback design for both strict feedback and pure feedback nonlinear systems. The idea of adaptive backstepping can be shown through the development of a Lyapunov-based controller which recursively considers some of the state variables as "virtual controls" by stepping back toward the scalar equation based control input (Zhou and Wen, 2008). The designed control system has a strong robust performance because of compensation of the uncertainty in every virtual control step. A pH process control research has been proposed to compare between adaptive backstepping and input-output linearisation techniques (Nejati et al., 2012). Simulated and experimental results show that the globally linearising controller based on pH reduced order model has a better performance than adaptive backstepping designs based on pH full order model. However, both of the design approaches need to measure particular state variables and precise state space model description for implementation.

Overall, the key challenge of nonlinear control system design is to establish a general framework that describes smooth nonlinear plants/processes (without linearisation) which allow the synthesis of the simple linear control laws (such as pole placement and state feedback). This framework plays an essential role in the control system design of a nonlinear dynamic system, because the controller input can be carried out effectively, only if the equivalent structure represents accurate nonlinear behaviours. Therefore, the modelling of nonlinear plants becomes particularly significant. It indicates that the desired framework must be available to describe a variety of nonlinear plants/processes and be simply applied to the nonlinear controller design.

1.2 Research motivation

There has already been many powerful methods and sophisticated implementation approaches for linear control successfully applied to industry. Most of the practical applications can be operated well by the designed control systems within the linearised model. It is nature question to ask 'why many researchers show an active interest in the development and applications of nonlinear control methodologies?' (Slotine and Li, 1991). There are many reasons to explain the important and necessary of the study on nonlinear control. It can be conducted as four essential aspects which are the improvement of the existing control systems, analysis of hard nonlinearities, the simplicity of the design procedures and strong robustness against uncertainties.

For nonlinear design, linear control methods rely on the fundamental assumption of small range operation for a similar linear model behaviour. When the nonlinearities compensation cannot adequately be guaranteed in the control system (such as the large operation range), the performance of a linear controller will be abysmal or to be even unstable (Slotine and

Li, 1991). Alternatively, the direct way is to design suitable nonlinear controllers to handle the nonlinearities in large range operation.

Nonlinearity can be almost found in every practical application. To simplify the procedures of analysis and design for nonlinear systems, a classical approach called linearisation, briefly introduced in the previous section, obtaineds the linearised model from the original nonlinear model; is approximated on selected operation point. However, the controlled plant dynamics are described more and more complicated for its complex dynamic behaviour. Meanwhile, the best match of desired performance and accuracy for the designed control system is required under the modern technology development. With the higher demands of the control system, the designed linearisation control system cannot achieve the desired performance for the nonlinear dynamic plant. For the computer technology development, the development of general and efficient nonlinear control system design methods are not only necessary and essential but also are currently considerable enthusiasm for the research and application.

In designing linear control systems, for example pole placement approach, it is usually to determine the desired closed loop system and resolve the pole assignor with the proper feedback loop. Substituting the particular parameters during calculations is necessary. However, uncertainties involved in the model parameters may exist and introduce many control problems. Examples of uncertainties could include: the ambient air pressure of aircraft (slow time parameter variation) or the internal parameters of a robot grasping arm (abrupt parameters change). A linear controller based on inaccurate or obsolete values of the model parameters may exhibit significant performance degradation or even instability (Slotine and Li, 1991). Two classes of nonlinear control system design approach respectively robust control and adaptive control are introduced to tolerated with nonlinearities against model uncertainties.

In the last several decades, robust control approach has been developed into a mature subject. It is concerned with the designed controller performance of original systems (with uncertainty) to deal with unknown disturbances. On the other words, the key issue of robust controller design is considering the bounded system uncertainty and how the robust controller can perform under this problem without any revision. Indeed, many unavoidable reasons that can lead to uncertainties appearing include: model mismatch, temperature change, component operation and measurement error. For nonlinear control systems, the existing general method of dealing with the nonlinearity is linearisation which always leads to unpredicted errors. Some issues cannot be designed appropriate controller by the linear control methodologies or found a satisfactory solution to analyse the system performances of a designed local stable linearised control system. The Linear Matrix Inequality (LMI) framework based robust stability conditions (Stipanovie and Siljakd, 2001) is proposed to design a stabilising feedback law with a bounded uncertain nonlinear disturbance terms. The enhanced LMI based control system design (Shen and Zhu, 2004) is proposed to solve a class of robust stability analysis and the closed loop system has a larger stable bound. The designed feedback controller can be directly obtained by computational simulation, which can be easily verified highly improved effectiveness and efficiency of the controller design. However, LMI based control system is difficult to directly apply to the design for nonlinear state space model and polynomial model. So that a generalised control-oriented framework for the nonlinear dynamic model is necessary.

State space model is a convenient structure for restoring in computer memory for a modern control system which is also essential to reduce the complexity of the mathematical expressions (Ogata, 2009). Based on state space model, linear controller design approaches have been well theoretically studied in research publications and validated in a wide range of industrial applications (Zhu et al., 2016). For a linear polynomial model, many realisation approaches (such as controllable or observable realisations) are available to convert into state space representation to satisfy the primary state space fundamental requirement. Compared with the linear model, a nonlinear polynomial model is complicated and more difficult to transfer into a proper state space expression, nevertheless almost impossible to convert into an equivalent linear state space model (Zhu, 2016). Is there any proper way to propose a powerful algorithm to directly use linear control approaches for the design of nonlinear control?

Overall, researchers are continuously discovering possible ways to directly use linear controller design methodologies to develop nonlinear control systems for both polynomial model expression and state space expression. Therefore, a geometric synthesis framework can be established to represent nonlinear dynamic plants and simplify and generalise nonlinear control system design procedures.

1.3 Research Questions

From the above information, research questions of this project can be listed as follows:

- How can the robust stability for the designed U-model based pole placement control system be analysed? How can the enhanced robust controller for the U-model based control systems be developed to deal effectively against uncertainty such as parameter variation?
- Based on U-model methodology, is there any potential solution to define the new U-model realisation within state space description so that linear state space design approaches can be applied for nonlinear control system design?
- What is an effective way to control both linear and non-linear quad-rotor models using only a linear U-state space control system?
- What is an effective way to control both linear and non-linear inverted pendulum models using only a linear U-state space control system?

1.4 The Aims and Objectives of the Project

With such insight of the U-model based design approach for nonlinear polynomial control systems (Zhu and Guo, 2002), the aim of this PhD research is to develop and analyse enhanced robust control systems, and to expand this powerful approach into systems described in state space expressions. This required establishing a comprehensive U-state space framework converted from nonlinear dynamic state space models, which realises the direct use of mature linear state space design approaches for the nonlinear control systems design and widen the related U-model technique approach for practical applications.

Therefore, this research not only brings forward new concepts (such as U-state space model and U-state feedback) as well as algorithms in academic research development, but also provides useful generic solutions for industrial applications in modelling and control of complex modern systems. The figure 1.1 graphically presents the aim and objectives of this project.

To achieve this aim the following major objectives have been outlined:

- To provide a general framework for using linear state space control strategies to develop control systems for nonlinear polynomial plants. The U-block model is defined as a linear input output model, which is converted from the closed loop system of U-model based pole placement control system.
- To design for effectiveness and efficiency within control systems, an accurate and precise dynamic model is generally necessary to represent in mathematical realisation. The uncertainty always exists due to model mismatch or other unexpected reasons (e.g. temperature change or component variation). Therefore, robust performance is one of the most important indexes for evaluating the behaviour of control system against uncertainties. Based on U-block model, linear robust analysis approaches are applied to discuss the robust performance of U-model based pole placement control system.
- To design an enhanced LMI based robust control system to improve stability against the uncertainty of the linearised model within the U-block model structure.
- To establish a new prototype U-state space model to represent a class of nonlinear state space model and the general linear/nonlinear state space model which can be easily converted into U-state space model expression.
- To design a U-state space model based control system and analyse the system stability.
- Case studies: Bench test on selected dynamics model to implement the corresponding U-model design approach. For example, a Hammerstein model is selected to demonstrate LMI robust analysis and robust control system design. The quad-rotor rotorcraft dynamic model and inverted pendulum are selected to demonstrate the performance and applicability of the proposed U-state space based feedback control strategy.

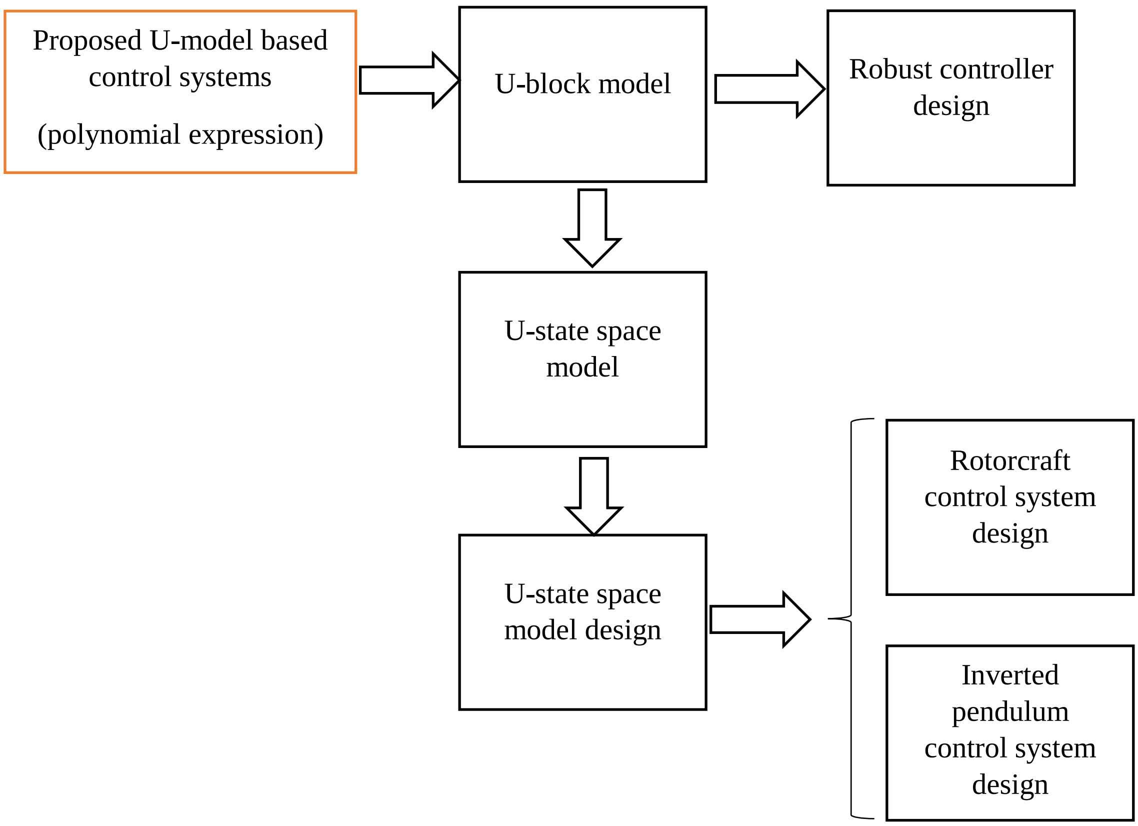

Figure 1.1 The diagram of project aims and objectives

1.5 Contributions

The contributions of this thesis are mainly

- The first study of U-model based design approach has been proposed in pole placement controller design for nonlinear dynamic plants (Zhu and Guo, 2002). This new prototype can be represented as a wide range of smooth nonlinear polynomial models. In the following decades study, researchers are developed various control algorithms (such as adaptive control and general predictive control) and bench tested with computational simulations and applications which are mainly focused on the nonlinear polynomial control systems. Based on those fundamental studies, the new input output model, named U-block model, is proposed to represent the closed loop of U-model based pole placement control system. This equivalent block model behaves as a linear model. The LMI approach is applied to this transformation for robust analysis and robust controller design.

- The initial U-state space platform has been established. The new description of U-state space model is defined and is converted from the original nonlinear state space model in a straightforward manner. The U-state space model based state feedback control for nonlinear dynamic model has been developed and bench tested through computational simulations. This design approach can be easily extended to some MIMO/SIMO systems.
- Within U-model methodology, the nonlinear control system design can directly use those mature linear design approaches. This significantly simplifies the design procedures and provides straightforward step by step iterative calculation results on computational algorithms. It should be mentioned that this approach does not require any linearised approximation before applying linear design approaches. This method is applied to design linearised rotorcraft (and inverted pendulum) control systems.
- The U-model methodology is also available for nonlinear control system design. Compared with classic linearisation design, it gives simple control structures (U-mapping structure) and detailed numerical solutions within an effective framework. The bench test platform nonlinear control of rotorcraft (and inverted pendulum) systems have been established.

1.6 Outline of the thesis

Chapter 1 introduces the research background, motivation, project aim and objectives, as well as highlighting the contributions for research development and puts forward the main research outcomes.

Chapter 2 briefly introduces the description of U-model, which is followed by the literature review of U-model based pole placement control system design; introduced to represent the fundamental methodologies. Also, other U-model based control systems are also proposed to introduce the efficiency and effectiveness of the U-model approach during last decade.

In chapter 3, a procedure for LMI based robust stability analysis of U-model pole placement control system is presented to determine the stability range. Then, an enhanced U-model LMI based robust control system is designed to enlarge this robust stability range. Finally, the computational simulation results are presented to verify the effectiveness of the enlarged robust stability bound.

Chapter 4 establishes a U-state space realisation which is converted from the nonlinear state space dynamic model. Within the U-state space platform, the controller for the nonlinear control system is developed by using linear state feedback approach. Through the numerical simulation, it can be inspected that the system performance of designed U-state space control system achieves the desired requirements.

In chapter 5, the proposed U-state space control system design approach is applied to develop the controller for a nonlinear quad-rotor rotorcraft model. Firstly, a brief introduction to quad-rotor modelling is presented. In order to test the availability of U-state space design approach, a nonlinear quad-rotor model is selected as the dynamic plant for implementation. Then the simulation results of navigation and control architecture for the quad-rotor are presented to highlight the application and performance of the proposed control laws.

In chapter 6, an inverted pendulum system is selected to demonstrate the U-state space control algorithm. The standard inverted pendulum system is presented. The U-state space feedback control is applied to this SIMO system. The numerical case study is selected to simulate the developed control system performance. Then the simulation results are presented to analyse the performance of the U-state space control.

Finally, in chapter 7 conclusions are drawn to summarise the study, the key findings can be summarised as below:

- The U-block model, defined as an input output model, is converted from the closed loop of the U-model based pole placement control system.
- Based on the current development of the U-model methodology for polynomial models design, an applicable robust analysis method is proposed to validate the robust performance for U-model based control systems. The robust stability margin

is analysed by the determined LMIs. The LMI based robust control system is designed for U-model based pole placement control system to improve the robust stability margin.

- The initial U-state space platform is established. The new description of U-state space model is defined and then converted from the original nonlinear state space model.
- The U-state space model based state feedback control for nonlinear dynamic model and bench test on with computational simulations. This design approach can be easily extended to some MIMO/SIMO systems.
- The design for nonlinear control system can be directly applied to mature linear design approaches, which significantly simplifies the design procedures and provides straightforward step by step iterative calculation results on computational algorithms. It should be mentioned that this approach does not require any linearized approximation before applying linear design approaches. The U-model based approaches bridge the linear design approaches with nonlinear dynamic plants including both polynomial and state space descriptions. The U-model based approach is also suitable for linear control system design, which gives simple control structures and solutions within a general and effective framework.

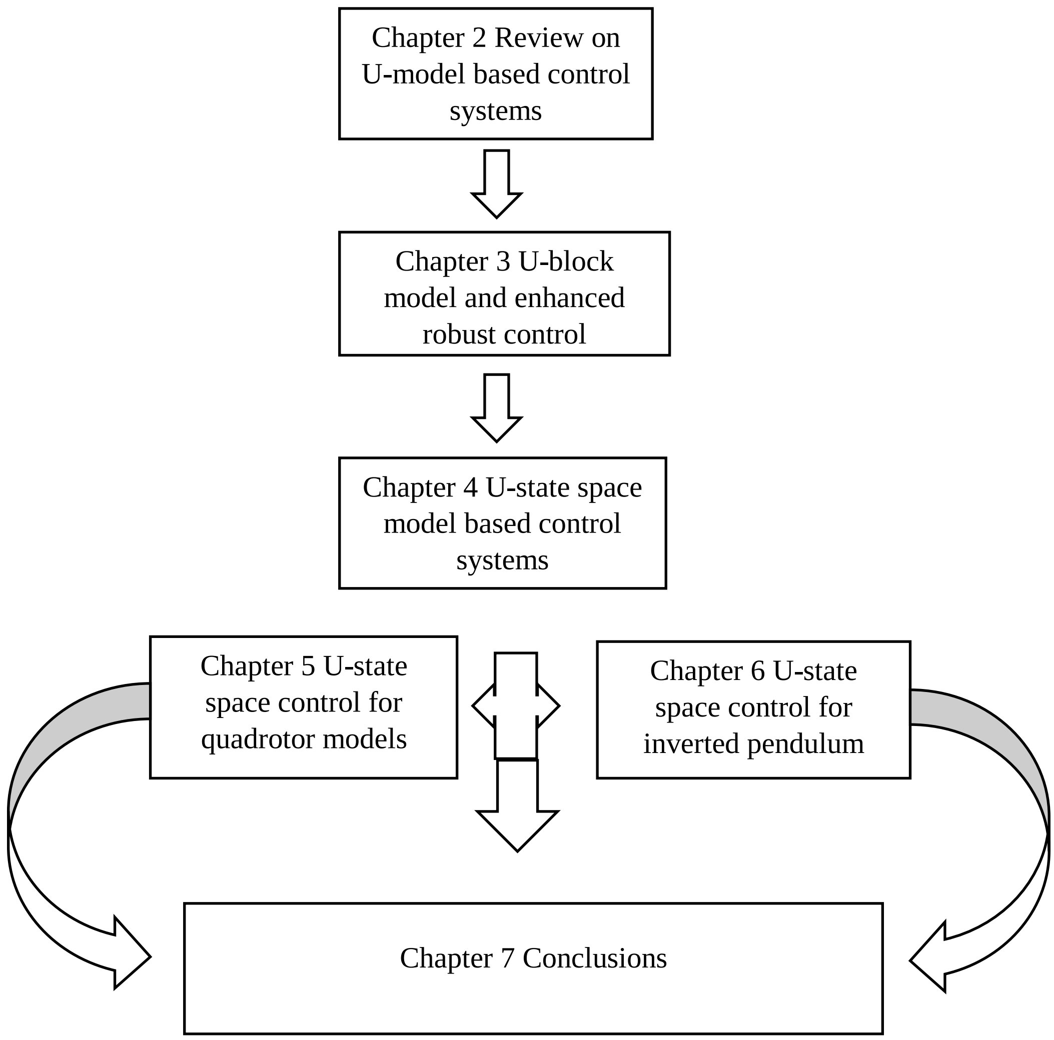

Figure 1.2 Flowchart of thesis chapters

Chapter 2

Overview of U-Model based Control System Design

2.1 Introduction

As introduced in the last chapter, linearisation is one of the most popular approaches for nonlinear control system design. There are two well-known methods for linearisation; Feedback linearisation and State Dependent Parameter (SDP) Transformation. Feedback linearisation (including input-state linearisation and input-output linearisation) transforms the equivalent linear expressions from original nonlinear state space description models (Isidori, 1995; Slotine&Li, 1991). For cancelling the nonlinear dynamics in the closed-loop, the central idea of feedback linearization is to convert the nonlinear model into a linear form by the appropriate coordinate transform. Then, linear state-space approaches to designing the corresponding control systems can be implemented for the obtained linear model. However, this method requires the application of a case by case approach with a certain degree of skill in manipulating differential equations and selecting coordinates (Zhu et al., 2016). It should be noted that this state space linearisation approach cannot to deal with the nonlinear polynomial model based control system design. Further details about this method will be discussed in Chapter 4.

The other typical approach based on polynomial models is called the state dependent parameter (SDP) transformation. The fundamental idea is to treat nonlinear polynomial models as time varying linear models, which reduces the closed loop system to a linear transfer function with the determined poles. The stability performance of the nonlinear system is considered at the design stage by assuming pole assignability at each sample (Cimen, 2010; Taylor et al., 2009). It can be found that there is a common strategy for the nonlinear control system design in many studies, that attempts to convert/build up an equivalent linear form to represent the original nonlinear dynamic plants. Then, the linear control algorithm can be demonstrated for the transferred linear models. From the model structure side, SDP transformation provides a link between the nonlinear polynomial model and the linear time varying state space expression. However, this transform does not provide the formative framework/prototype or clear process to follow.

More recently, a new methodology, referred to as the U-model approach, has been proposed with clear advantages to Feedback Linearisation and SDP transformation design methods. Consider a polynomial function $y = f(.)$ and an open set $A = \{x_1, x_2, x_3....x_n\}$. The value of $f(.)$ at $x = (x_1, x_2...x_n)$ can be denoted as $f(x) = f(x_1, x_2...x_n)$. The function can be said to be a smooth function if its partial derivatives of any order with respect to x_i exist and are continuous (Zhu et. Al, 2016). Based on the time varying parameters polynomial, it can present a wide range of smooth nonlinear systems without any deficiency for the nonlinear characteristics and dynamic performance of the plant. Thus, U-model provides the traditional nonlinear system constructing a universal structure that can be used by nonlinear controller design. As a result, it bridges the gap between the linear control system design method and nonlinear dynamic system, which makes the design of nonlinear control system with the simple process to design linear control system come true.

In this chapter, the U-model methodology is introduced as being fundamental to representing a wide range of the nonlinear polynomial models. The structure of this chapter is organised as follows. In section 2.2, the description of the U-model framework is defined. A brief introduction to the concepts of mathematical transformation from the nonlinear polynomial model into U-model expression is given. In section 2.3, the earliest implementation of the U-model based design approach (U-model based pole placement control system design) for the nonlinear dynamic polynomial model is presented. An outline of different linear control design approaches developed by numerous researchers in development of U-model based control systems is presented in section 2.4. Finally, section 2.5 gives general summaries of this chapter.

2.2 Description of U-model

A key challenge of nonlinear control system design is to develop a standard model prototype with conciseness, flexibility and manipulability. Therefore, U-model based control system design is proposed to firstly represent a broad range of linear and nonlinear dynamic plants. Regarding time varying parameter polynomial, the U-model approach skillfully converts the original nonlinear model into a linear control designable framework (U-model). The merit of the U-model framework for the nonlinear system controller design is the signifiant reduction in the difficulties encountered in nonlinear control system synthesis as well as computational complexities.

Compose Single Input Single Output (SISO) nonlinear dynamic plants with the Nonlinear Auto-Regressive Moving Average with exogenous inputs (NARMAX) representation of the form as follows (Zhu and Guo, 2002):

$$y(t) = f[y(t-1),\ ...,y(t-n),u(t-1),\ ...,u(t-n),e(t),\ ...e(t-n)] \qquad (2.1)$$

where $y(t)$ and $u(t)$ are the output and input signals of the plant respectively at the discrete-time instant t, n is the plant order, $f(.)$ is a nonlinear function and the modelling error term $e(t)$ could be induced from measurement noise, disturbance, plant variation, uncertain dynamics, modelling inaccuracy and imperfect or partial knowledge of plants. Note that here the plant delay has been assumed to be one for the sake of brevity. Without losing generality, the proposed control procedure is applicable for arbitrary known plant delay as well. Leontarities and Billings (1985) have shown that such the NARMAX model can represent a broad class of nonlinear systems. Furthermore, the Hammerstein, Wiener, bilinear and several other well-known linear and nonlinear model sets can be shown to be special classes of the NARMAX model. With its generality, the difficulty occurs when controlling a plant based on the NARMAX model is considered because of the lack of a manoeuvrable structure. Therefore various possibilities for parameterising $f(.)$ exist including the extended model set NARMAX models. The control oriented model for the nonlinear dynamic plants can be expressed as the polynomial of $u(t-1)$, as follows:

$$y(t)=\sum_{j=0}^{M}\lambda_j(t)u^j(t-1)+e(t) \qquad (2.2)$$

where M is the power of model input $u(t-1)$, the parameter $\lambda_j(t)$ is a function of the past inputs $u(t-1),\ ...,\ u(t-n)$, the past outputs $y(t-1),\ ...,\ y(t-n)$, and errors $e(t),\ ...,\ e(t-n)$.

Rearranging the polynomial in equation (2.2), the control-oriented model can be derived as a classic power series of input $u(t-1)$ with related time varying parameters $\lambda_j(t)$. Here errors $e(t),\ ...,\ e(t-n)$ are unknown quantities, equation (2.2) is a more realistic representation for the nonlinear dynamic plants, which can be described a general nonlinear plant with this kind of sample mathematic expression. Equation (2.2) is the U-model expression, where $y(t)$ if directly used in the linear controller design method, then one of the roots solved can be obtained the output of the controller (more details in section 2.3). Note that during the transformation of the U-model from the nonlinear dynamic plants, it does not lose any nonlinear dynamic characteristics, so this control oriented model highly improves the accuracy and efficiency for the nonlinear control systems design.

Such an instance illustrates the conciseness and generality of U-model transformation. Suppose an expression of the nonlinear plant as follows (Quan and Guo, 2002):

$$\begin{aligned} y(t) &= 0.1 + 0.9y^2(t-1) + 0.4u(t-1)e(t-1) - 0.4y(t-1)u^2(t-1) \\ &\quad + 0.6y(t-1)u^3(t-1)e(t-2) + e(t-1) + e(t) \end{aligned} \tag{2.3}$$

which can be rewritten in the notation of equation (2.2) as

$$y(t) = \lambda_0(t) + \lambda_1(t)u(t-1) + \lambda_2(t)u^2(t-1) + \lambda_3(t)u^3(t-1) + e(t) \tag{2.4}$$

where $\lambda_0(t) = 0.1 + 0.9y^2(t-1) + e(t-1)$, $\lambda_1(t) = 0.4e(t-1)$, $\lambda_2(t) = -0.4y(t-1)$ and $\lambda_3(t) = 0.6y(t-1)e(t-2)$ are time varying parameters.

Note that the parameter $\lambda_j(t)$ is a function of past inputs and outputs $u(t-1), \ldots, u(t-n)$, $y(t-1), \ldots, y(t-n)$ and errors $e(t), \ldots, e(t-n)$ and, in particular, $e(t)$ is an unknown quantity, which hence is unpredictable. Therefore, equation (2.4) is a more realistic representation for real nonlinear plants, and the above representation in equation (2.2) is mathematically simple and can be used to represent a wide class of nonlinear plants in practice as well.

The U-model framework has some advantages compared with other nonlinear model transformation approaches:

- The proposed U-model framework is more practical than the other models such as NARMAX model and Hammerstein model.

- The U-model framework is applicable to use in almost all of the smooth nonlinear discrete time input-output dynamic models, and the mapping is reversible.

- Nearly all expressions of the sampling data of the input-output nonlinear dynamic model can be presented as equation (2.2). Thus the discrete time nonlinear dynamic models are obtained the new expressions from equation (2.2).

- When a polynomial structure exists in the current control model, the linear controller design approaches can be directly used to design control system for the polynomial structure expression nonlinear models. This methodology is not only simplifies the processes of root solver but also proposes a suitable controller design approach for a class of nonlinear dynamic control systems. The remarkable advantage is that the output of the controller is obtained from the only root solved by the nonlinear constant equation. There is a remarkable difference between the U-model and state-dependent model. The U-model formula is the power series of the current control law, and the coefficients of U-model are the time varying function of the input and output at the past time. But the state dependent model consists of the linear combination of the system output at the past time, whose coefficients are the time varying function of the state variables.

2.3 U-model based Pole Placement Control System Design

Pole placement is one of the most popular and powerful methods to resolve control system design problems in a wide variety of engineering fields. This is a simple method for controller design that the idea is to determine a controller as offset operator to place the desired poles of the closed loop systems (in Figure 2.1). Since in linear system, the use of it requires the desired state variable is controllable, pole placement usually cannot be used immediately to nonlinear dynamic model because the behaviour of the nonlinear model is very difficult to map the position of the system zeros and poles. In fact, the nonlinearity of nonlinear models is always left out in the existing design of nonlinear control system so as to treat it as a linear closed loop system.

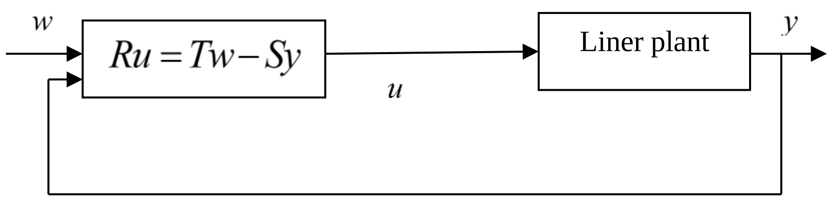

Figure 2.1 A general linear pole placement control system (Astrom and Wittenmark, 1995)

This type of pole placement method can only be used to design local operating points of the controlled object in the nonlinear model. With a limitation of local performance, such kind of design is likely to generate unacceptable performance in the strict nonlinear conditions. While the nonlinear objects under U-model framework can be used for the design of pole placement in the nonlinear system due to the fact that its control variable is expressed through polynomial and that the variable is controllable. This means that the pole placement of nonlinear system can be realized by acquiring the output of controller through solving of the polynomial equation.

Zhu and Guo (2002) expound how to do the pole placement controller design for the nonlinear dynamic model on basis of U-model framework. By adopting the negative feedback principle of control theories, the pole placement based on U-model framework compares the expectant output with the actual one and obtains the input signals of the controller so as to finish the design of controller and to settle down the desired poles. As a result, the output of the controller is found by solving the mathematical equation (2.2) and the design of control system is completed by analysing and demonstrating that the polynomial with the current control input $u(t-1)$ effectively, whilst decreasing the difficulty and computational complexity of nonlinear control system.

Figure 2.2 shows the block diagram of the U-model based pole placement control system. In the U-pole placement design, the U-model is firstly transferred from the nonlinear model. With the polynomial equation of U-model as a root solver, the Newton-Raphson (Langtangen, 2012) algorithm can be used to find the controller output $u(t-1)$, which is also the control input at the next time-step.

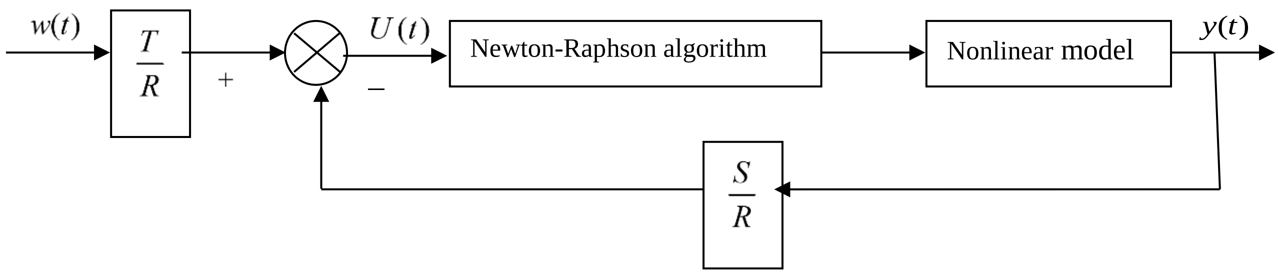

Figure 2.2 Block diagram of U-model based pole placement control system

The general pole placement control law can be obtained from equation (2.2):

$$RU(t) = Tw(t) - Sy(t) \qquad (2.5)$$

Where $w(t)$ is the reference for the output, the polynomial R, S and T are the forward operators. From the control law (2.5) represent a negative feedback and feed-forward with the transfer function $\frac{S}{R}$ and $\frac{T}{R}$. The output $y(t)$ can be described as the equation with $w(t)$, as following:

$$y(t) = \frac{T}{R+S} w(t) = \frac{T}{A_c} w(t) \qquad (2.6)$$

where polynomial A_c is the closed loop characteristic equation. The polynomial R, S and T can be resolved by a Diophantine equation.

To make the control output equal the desired output $U(t) = y_d(t)$, polynomial T is specified with $T = A_c(1)$ from equation (2.6). The signal $U(t)$ can be obtained by equation (2.5) with the determined polynomial operator *R*, *T* and *S*. The remaining design task to resolve one of the roots of (2.2) to obtain the controller output $u(t-1)$, which is

$$u(t-1) = \psi\left[y_d(t) - \sum_{j=0}^{M} \lambda_j(t) u^j(t-1) = 0 \right] \qquad (2.7)$$

With $U(t)$ as a root solver, the Newton-Raphson algorithm can be used to find the root $u(t-1)$ as the controller output.

$$\begin{aligned} u_{k+1}(t-1) &= u_k(t-1) - \frac{\phi[U_k(t-1)] - U(t)}{d\phi[u(t-1)]/du(t-1)} \\ &= u_k(t-1) - \frac{\sum_{j=0}^{M} \alpha_j(t) u_k^j(t-1) - U(t)}{d[\sum_{j=0}^{M} \alpha_j(t) u^j(t-1)]/du(t-1)} \end{aligned} \qquad (2.8)$$

Zhu and Guo (2002) also analyse the pole placement controller performance including identification error and stability of the controller. There will be an unpredictable quantity $e(t)$ contained in $U(t)$. The other unknown items $e(t-1),...,e(t-n)$ can be identified with the estimated value at each sampling time. In each sampling interval, the root solver always converges to a real root. Then the close-loop system output should be converged closely to the desire value, but with some discrepancy due to the modelling errors. The disturbance cannot change the poles of the close-loop system.

The pole placement controller output can be solved and the stability of the controller can be guaranteed with the root solver of the Newton-Raphson algorithm. However, there may be some problems that lead to the instability of the controller. One is a critical point where the derivative of the function equates to zero caused by plant variation, calculation error or the unsuitable initial value. As the infinite value of root solver, the controller will be unstable. This problem can be solved with the subsidiary condition of iteration calculation. Instability could also occur if no real root of the polynomial equation exists, so the algorithm will break down under this situation. An enhanced Newton-Raphson algorithm is proposed to guarantee the stability of the controller in a minimum phase system (Zhu, et al. 1999).

With this procedure, it does not request the plant model (no matter line or nonlinear) in the conversion stage. It only uses the plant model, an equivalent U-model expression, to obtain the controller output $u(t-1)$ in the second stage. For a nonlinear plant model, the calculation is merely to resolve one of the roots from the U-model (2.2). For a linear plant model, it has a simple expression,

$$u(t-1)=\frac{y_d(t)-\lambda_0(t)}{\lambda_1(t)} \qquad (2.9)$$

where $\lambda_1(t)$ is the coefficient associated with $u(t-1)$ (for linear time invariant models, $\lambda_1(t)$ is a constant). $\lambda_0(t)$ (non-zero value) is the summation of the rest of the terms in the linear model (Zhu & Guo, 2002).

2.4 Other U-model based Control Systems

Many researchers have proposed novel algorithms through the use of U-model based control system design for nonlinear dynamic models in the last decade. For the nonlinear polynomial model design, U-model structure has been applied to many different kinds of design approaches including adaptive control, internal control, sliding mode control, predictive control and neural network control.

In a stochastic control system, a varying learning rate of radial basis neural network control law proposes to identify the U-model time varying parameters for the nonlinear dynamic plants by the least squares method (Chang et al., 2011). The control law is constructed as a part of the radial basis neural network. The amount of stochastic U-model parameters are already known, supposing a minimum performance index number structure which is applied to a varying learning operator to update the time varying parameters $\lambda_j(t)$ and the weight value w_i of the radial basis neural network.

The U-model framework based adaptive control algorithm is proposed to control a class of stochastic discrete nonlinear dynamic plants with unknown parameters (Wu et al., 2011). Assuming that the time varying parameters of U-model is unknown, and the new recursive least squares method is developed to identify the unknown plants. The unknown time varying parameters in the U-model expression is identified by least squares method and demonstrates the convergence for the online estimation of the time varying parameters in the case studies.

U-model based adaptive tracking scheme for unknown MIMO bilinear system is proposed to develop a simple expansion for the nonlinear which is similar linear behaviour dynamic plants (Ali et al., 2006). A general dynamic bilinear control system is transformed to the U-model framework and used Radial Basis Function (RBF) to estimate the time varying parameters of U-model in the radial basis neural network online.

Research has also focused on the U-model based adaptive IMC control systems design (Muhammad and Butt, 2005). The dynamic nonlinear plants are modified into the U-model expression which time varying parameters are identified by using an adaptive filter. The adaptive tracking nonlinear models are based on the root of the solver and compose a simple control law. The advantages of the U-model based adaptive tracking algorithm for nonlinear dynamic plants are applied to a class of the nonlinear dynamic plants, which are stable at the past time. Moreover, the conventional and simple control law design can be used in this adaptive tracking control system.

The internal model control method is applicable to both linear and nonlinear dynamic systems, which is extremely useful for stabilisation open loop control systems. In an internal model control system, the closed loop system is input output stability determined by a stable controller and an accurate stable dynamic model to represent the dynamic plant. If the inverse of the dynamic plants exists, the close-loop system with this designed control law (determined by the inverse of the dynamic model) is input output stability. For a standard control law, the U-model expression in (2.2) is easily transformed from original model (such as neural network). The control input $u(t-1)$ can be solved easily with the Newton-Raphson algorithm from a general and standard control law, and this method is very suitable for the nonlinear control area.

U-model based inverse control is proposed to control the dynamic systems by a nonlinear filter which includes an adaptive linear scheme and a nonlinear polynomial pre-processor (Tahir, K., Muhammad, S., 2006). The pre-processer generates a signal of power series of input $u(t-1)$, the nonlinear adaptive filter is used to modify the unknown dynamic plants by producing the signal $u(t-1)$.The sum of the signal weight value is regarded as the output of the nonlinear adaptive filter. With the standard of least mean squares error criterion, the weight value of nonlinear adaptive filter is adjusted to minimise the least mean squares error.

Facing the great challenge of the design of generalised predictive controller for a class of nonlinear dynamic plants, a U-model based control system is the suitable solution for the generalised predictive control system design (Du et al., 2012). The key point of the generalised predictive algorithm is to transform the nonlinear dynamic model into the U-model framework and then to obtain controller output by the Newton-Raphson algorithm root solver. The model obtained by any of the linearisation methods has more or less the error from approximation. However, U-model framework is an accurate model matching without any approximation. U-model based generalised predictive control has the minimum performance index, and this achievement breaks the limitation of the generalised predictive control method directly used in the nonlinear dynamic control system. The dynamic model of the generalised predictive control system is the linear dynamic system that is expressed as the controlled integral regression moving the average model. A class of unstable disturbances can be described with this model and the steady state error of the generalised predictive control system is guaranteed for the designed control law including the integral effect.

It was found that there has been limited research on the analysis robust performance. Some studies have discussed the use of feedback analysis in determining the optimal learning rate to guarantee the stability of algorithm in the U-model based adaptive control systems for nonlinear dynamic plants (Ali et al., 2008). The importance of algorithm robustness is the estimation error under the influence of the disturbances i.e. the small disturbance may lead to the small estimation error, besides it also may lead to big identification error. The robustness of the adaptive algorithm can be regarded as a bounded positive constant that is derived from the proportion of estimation error energy and disturbance energy. The U-model based control systems robustness, stability and convergence speed of the adaptive algorithm is guaranteed by the compressed mapping and the boundary of the learning rate; the learning rate is selected by using small gain theorem.

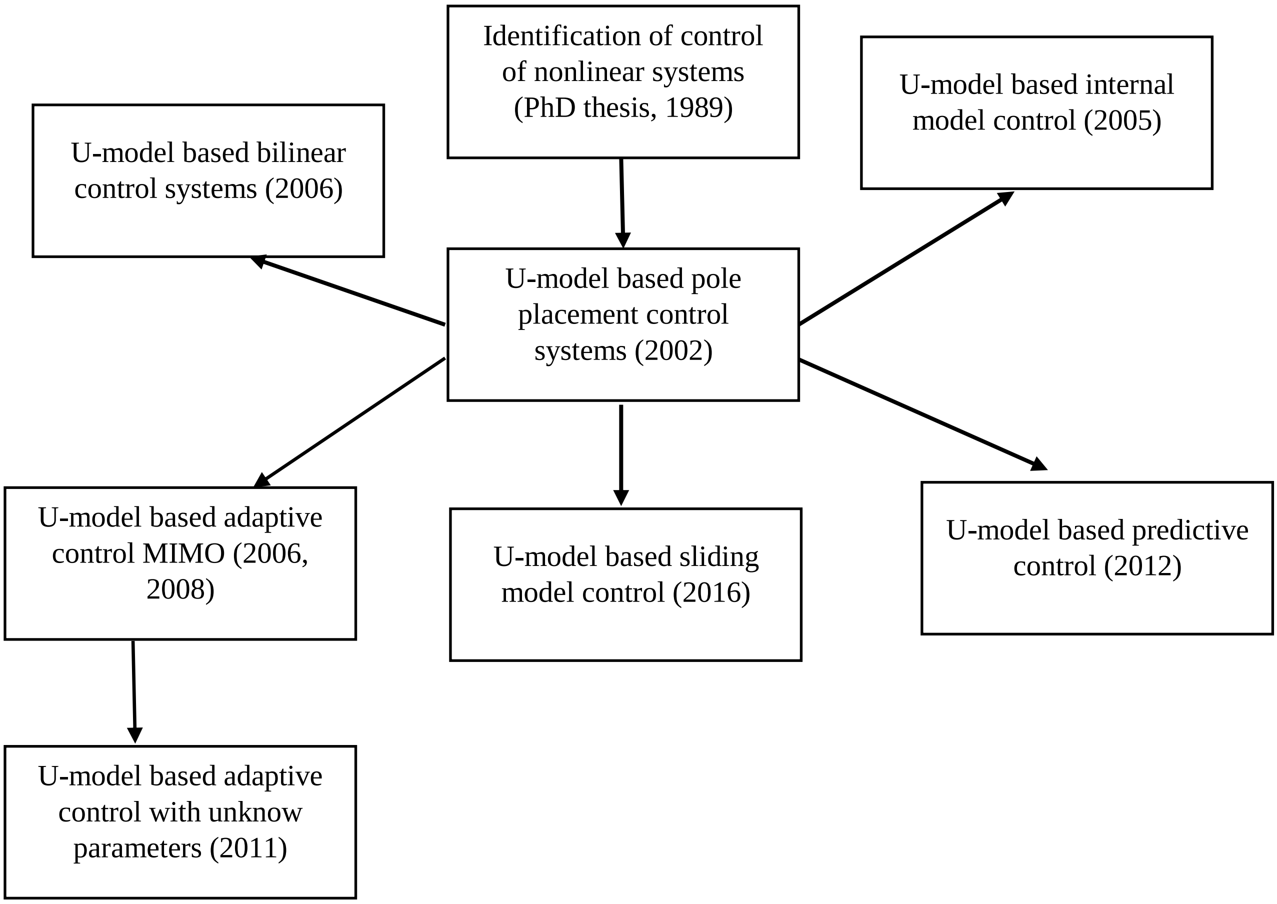

Figure 2.3 U-model technology development

2.5 Summary

In this chapter, the description of U-model framework is introduced to define the new converted equivalent realisation for nonlinear discrete time polynomial dynamic model which is called U-model representation. Pole placement is one of the most common and useful control system design approach in engineering areas. Based on U-model structure, a standard pole placement control system for the nonlinear dynamic plant has been proposed to use linear controller design method for nonlinear dynamic plant.

Many other linear design methods have been successfully demonstrated for the nonlinear control system. Although not widely attended, U-model methodology, is a generic systematic approach to convert the nonlinear polynomial model into a controller output $u(t-1)$ based time-varying polynomial model. This has been studied (Du et al., 2012; Muhammad and Haseebiddon, 2005; Zhu, 1989; Zhu and Guo, 2002) for facilitating nonlinear control system design over the last decade. Consequently, linear polynomial model-based design approaches (such as pole placement and general predictive control can be directly used to design such nonlinear control systems (Du et al., 2012; Zhu and Guo, 2002). The major contribution of the U-model-based design procedure can be listed in order (Zhu et al., 2016).

(1) In methodology, those well-known approaches developed from linear systems can be directly applied to nonlinear control system design, which significantly reduces the design complexity and effectively provides straight forward computationally efficient algorithms. It should be noted that this new approach does not request to linearize nonlinear plant models in design. It is just to use linear design approaches to directly design nonlinear plant model based control systems.

(2) In design, it obtains desired plant output first (compared to designing desired controller output in the classical framework) and then works out the controller output from the U-model in a relaxed root-resolving routine (compared to resolving complex solutions from the inverse of the whole designed systems).

(3) For linear control system design, it provides new insight and solutions within a more general and effective framework.

It should be noted that the U-model-associated publications are still in the very early stages. Although various algorithms have been developed and bench tested with simulations and applications, they have not had a rigorous analytical description and highly regarded journal publications until recently (Zhu et al., 2016). It is the author's belief that the next research progression should be the development to make the U-model approach available for using linear state space based design approaches to design the control of nonlinear polynomial models described plants. The key issue in the on-going study is obtain an equivalent linear input output model from its U-model based nonlinear control system. The linear robust analysis and design approach will apply to the input output framework for improvement robust performance of U-control systems against uncertainties.

Chapter 3

Robust Analysis of U-model based Control System

3.1 Introduction

Robustness is a method to evaluate the performance changes of a designed control system with system parameter variation (Skogestad and Postlethwaite, 2007). In recent years, there have been two significant breakthroughs in modern robust control; H_2 and H_∞ optimal control have both attracted a lot of research interest.

H_2 robust control systems have advantages on performance especially for handling stochastic aspects such as measurement noise as well as control cost capture (Peng, 2014). The H_2 robust target considers to minimise the output energy by optimising an upper bound in the worst case of H_2 norm (Yu, 2002). Based on this H_2 controller, the closed loop of the control system has excellent dynamic performance. However, H_2 controller design lacks robustness to external disturbances and the mathematical operation of H_2 norm is very limited due to non-induction characteristics.

H_2 control research provides very useful fundamental theorems and references for H_∞ problems. The H_∞ robust design minimises the H_∞ norm of the transfer function which is a maximum singular value of the transfer function matrix. The performance of H_∞ is convenient to enforce robustness to model uncertainty, but it is based on compromising system performance (Peng, 2014).

Consider a standard state space linear system with matrices $A, B, C \text{ and } D$. The transfer fucntion is defined as $T(s) = C(sI - A)^{-1} B + D$. The H norm can be defined as $\|T(s)\|_\infty = \sup_\omega \sigma_{\max}(T(j\omega))$. The solution for H_∞ performance optimization of the system can be solved as the following eqations (Yu, 2002):

$$\min \gamma$$

$$\text{s.t.} \begin{bmatrix} A^T P + PA & PB & C^T \\ B^T P & -\gamma I & D^T \\ C & D & -\gamma I \end{bmatrix} < 0$$

$$P > 0$$

Although H_∞ norm robust techniques have a wide ranging popularity among researchers and have provided the platform for many successfully theoretical solutions to control problems, application demonstrations with those approaches is quite limited. One of the reasons for this is due difficult in the implementation of the high order of the synthesised controllers. The other reason is the preference to use certain controller structures with simple tuning and reliable stability such as PID control or microprocessor instead of H_∞.

Combined with H_2/H_∞, LMI techniques are often considered for multi-objective synthesis thus providing more flexibility for combining various design objectives in a numerically tractable manner, and can even cope with those problems to which an analytical solution is out of the question (Soliman et al., 2011; Chesi, 2011).

To analyse robustness of control systems, the uncertainty of plant is very significant. The first priority which concerns a control engineer is the modelling an accurate description of the controlled subject. The uncertainties caused by environmental changes, components, internal drift and other unknown reasons are always occuring and will influence the existing control systems (Calafiore and Dabbene, 2002). Such uncertainty is entirely different from the external disturbance which is also unpredictable and not measurable before controller design.

For U-model methodolgy, the nonlinear model can be converted into U-model expression without losing any nonlinear dynamics. Based on U-model, it is also archiving to directly use linear H-norm robust design approach to deisgn for nonlinea systems.

In this chapter, the new U-model concept, U-block model, is defined as an equivalent input-output block from the closed loop transfer function of U-model based pole placement control system. In section 3.3, the robustness analysis is discussed against uncertainties. U-block model based robust control system is developed in section 3.4. The computational experiment used to demonstrate the proposed design approach is presented in section 3.5. Finally, a conclusion of this chapter is summarised in end.

3.2 U-block model

In chapter 2, U-model based pole place control system already has been introduced through the use of linear design approach for nonlinear polynomial plants. In this section, a new U-model concept is defined based on U-control systems.

Definition (Zhu et al, 2016): The closed loop of U-control system is defined as a new input output structure called U-block model. The U-block model can be express as

$$\frac{y(t)}{w(t)} = \frac{A_c(n)}{A_c}(desired\ closed\ loop)$$

Without any linearised approximation, the U-block model will behave as a general linear model. For example, the equivalent U-model expression is converted from the original nonlinear polynomial model. The required poles are assigned following the rules of a linear feedback control algorithm. As shown in Figure 3.1, the pole assignor is formulated by a revised U-model-based pole placement controller design algorithm (Zhu, 2016).

A standard pole placement controller is used as reference to design the internal part of U-block model. Consider the U-model of expression (2.2), a general controller can be structured as: (Astrom and Wittenmark, 1995)

$$Ru(t) = Tw(t) - Sy(t) \qquad (3.1)$$

where $w(t)$ is the reference input and *R*, *S* and *T* are the polynomials of the forward shift operator *q* with the following description

$$\begin{aligned} R &= q^n + r_1 q^{n-1} + \cdots r_n \\ T &= t_0 q^m + t_1 q^{m-1} + \cdots t_n \\ S &= s_0 q^l + s_1 q^{l-1} + \cdots s_n \end{aligned} \qquad (3.2)$$

where *q* is the forward operator, *n*, *m* and *l* are the orders of the polynomials *R*, *T* and *S*, respectively. A designed controller must satisfy the causality conditions order(*S*)<order(*R*) order(*T*) ≤ order(*R*), that is $l<n$ and $m \le n$.

The standard pole placement control law (3.1) represents a negative feedback loop with the transfer function *S/R* and a feedforward loop with *T/R*; it thus has two degrees of freedom. For the proposed U-control design procedure, control law (3.1) can be rewritten as:

$$Ry_d(t) = Tw(t) - Sy(t) \qquad (3.3)$$

where y_d is the desired system output.

By letting $y(t) = y_d(t)$, the designed U-block model can be linked to the reference $w(t)$ as:

$$y_d(t) = \frac{T}{R+S} w(t) = \frac{T}{A_c} w(t) \qquad (3.4)$$

where the polynomial A_c is the closed loop characteristic equation and specified in advance.

To cancel the possible output offset in steady state, i.e., to make steady state error equal to zero at the controlled output, polynomial *T* is specified with

$$T = A_c(1) \qquad (3.5)$$

The key idea of the design is to determine the desired closed loop characteristic polynomial A_c, then resolve the polynomials R and S through a Diophantine equation (Zhu and Guo 2002). After the desired plant output $y_d(t)$ is designed, the controller output $u(t-1)$ can be determined by resolving one of the roots of the U-model.

Theorem 3.1 (Zhu et al.,2016): The U-model based pole placement design procedure does not depend on the plant model. Only the solution of the designed controller output is involved in the plant model.

Equation (3.4) provides a new design framework, using the U-control design procedure once, then it can be applied to many different plant models, which only calculates the corresponding controller output each time from a given plant model. For example, a nonlinear plant polynomial model is selected and the design of its control system is undertaken with a linear state-space-based approach; the procedure can be ordered into the following step-by-step implementation:

(1) Convert the nonlinear polynomial model into the U-model expression (Note, not the linear approximation).

(2) By pole assignor (a pole placement algorithm), convert the U-model into an input–output linear closed-loop model with assigned poles. This is defined as the U-block model, which is linear.

(3) Convert the linear U-block model into its state-space realisation by commonly used realisation techniques. This is defined as the U-state-space model.

(4) With reference to the U-state-space model, using linear state-space based design approaches, design the control system.

3.3 Robustness analysis

3.3.1 LMI preliminaries

A standard description of linear matrix inequality can be expressed as:

$$F(x) = F_0 + x_1F_1 + \cdots + x_mF_m < 0 \qquad (3.6)$$

where $F_i = F_1, F_2 ... F_m \in \mathbb{R}^m$ are symmetric matrices. $x = (x_1, x_2 \cdots, x_m) \in \mathbb{R}^m$ are real vector and known as decision variables of LMI. The inequality symbol means the $F(x)$ in (3.6) negative.

In many control problems, the variables are described as matrices for the convenience of calculations such as Lyapunov stability theory (Isidori, 1995):

$$F(x) = A_a^T X + XA_a + Q_b < 0 \qquad (3.7)$$

where $A_a, Q_b \in R^{n\times n}$ are the given constant matrices, particularly Q_b is a symmetric matrix. $X \in R^{n\times n}$ is the symmetric unknown matrix which is required to be calculated.

If there exists a basic series $E_1, E_2, \cdots, E_M \in S^n$ that makes symmetric $X = \sum_{i=1}^{M} x_iE_i$, then equation (3.7) can be rewritten as:

$$F(X) = F\left(\sum_{i=1}^{M} x_iE_i\right) = A_a^T\left(\sum_{i=1}^{M} x_iE_i\right) + \left(\sum_{i=1}^{M} x_iE_i\right)A_a + Q_b$$

$$= Q_b + x_1\left(A_a^T E_1 + E_1A_a\right) + \cdots + x_M\left(A_a^T E_M + E_M A_a\right) < 0 \qquad (3.8)$$

LMI problems are divided into two main classes which are LMI feasibility problems and LMI Optimization problems.

- **LMI feasibility problems**

Based on a given LMI (3.7), the only concern of corresponding LMI problems is to find a feasible solution X_{fes} ($F(X_{fes}) < 0$) or determine that the LMI is not feasible.

- **LMI Optimization Problem**

Optimization problems are involved in the minimization or maximization; subject to LMI constraints:

$$\min \alpha\left(x_1, \cdots x_n\right)$$

$$\text{s.t.}\, F\left(X_1, \cdots, X_n\right) < 0$$

The numerical solution of LMIs can be implemented by Matlab LMI toolbox which provides a set of useful functions to solve LMIs.

3.3.2 LMI based robust analysis

Consider an uncertain parameter system:

$$\dot{x}(t) = A(\delta)x(t) \tag{3.6}$$

where $x \in R^n$ is the state vector, and $A(\delta)$ is the function of the real parameter vector $\delta = [\delta_1, ..., \delta_n]^T \in R^k$. Assume that the uncertain parameter δ takes values from the given set Δ, thus the uncertain system (3.6) is always asymptotical stable under this system robust stability condition. According to the uncertain parameter δ which is a time-varying parameter, the uncertain system (3.6) is also a time-varying control system. An effective method to discuss the stability of the time-varying system is the Lyapunov stability theory.

To all the uncertain parameters $\delta \in \Delta$, if and only if the positive definite matrix $P > 0$ satisfies

$$A^T(\delta)P + PA(\delta) < 0 \tag{3.7}$$

Thus the uncertain system (3.6) is quadratic stability.

In such a quadratic stable system (3.6), the quadratic form of Lyapunov function is obtained from equation (3.7):

$$V(x) = x^T Px, \ \delta \in \Delta \tag{3.8}$$

where equation (3.8) satisfies

$$\dot{V} = \frac{dV(x(t))}{dt} < 0 \tag{3.9}$$

According to Lyapunov stability theory, the uncertain system (3.6) is asymptotical stable. Note that the asymptotical stable uncertain system which is also called robustness stability can be obtained from the system quadratic stability.

Generally, Δ is an infinite set. Therefore, the definition of quadratic stability requires testing infinitely many matrix inequalities. It is obviously impossible to test in a specific control system.

$$A(\delta(t)) = A_0 + \delta_1(t)A_1 + ... + \delta_k(t)A_k \tag{3.10}$$

where $A_0,...,A_k$ is known $n \times n$ real constant matrixes and the uncertain parameter $\delta_i(t)$ is the bounded time varying function, where $\delta_i(t) \in [\delta_i^-(t), \delta_i^+], \ i = 1, 2, ..., k$.

Define a vertex set described as:

$$\Delta_0 = \{\delta = [\delta_1, ..., \delta_k]; \delta_i = \delta_i^- or \ \delta_i^+, i = 1, ..., k\} \tag{3.11}$$

The allowed value range of the uncertain parameter Δ is a convex cell of vertex set, which means that a set is constituted by all convex combinations of the middle point of vertex set Δ_0. The necessary theorem is proven in **Theorem 3.1** to demonstrate the quadratic stability of uncertain parameter systems.

Theorem 3.1 The uncertain parameter system (3.6) with the matrix $A(\delta)$ has quadratic stability, if and only if a symmetrical positive definite matrix exists, such that the LMI (3.7) is tenable (Yu, 2002).

It should be noted that this theorem establishes a basis for developing the algorithm for the

LMI based robust stability margin analysis used in this study.

The set Δ is infinite but Δ_0 is a finite set. Following the theorem, the only need is to test if the LMI is true or false so that the system quadratic stability can be estimated. The condition of the system quadratic stability is to judge the feasibility of linear matrixes based on a group of LMI.

The question of the feasibility of LMI has been solved through the use of the MATLAB LMI toolbox. In this toolbox, it provides a set of convenient functions to solve problems involving LMIs (Erkus and Lee, 2004). A standard LMI problem is normally solved in two stages in MATLAB. In the first step, the LMIs are defined to specify the decision variables. In the following step, the remaining task is to solve numerically using the available solvers in order to find any flexible solutions.

The MATLAB LMI toolbox supplies the function to test the quadratic stability of the uncertain parameter system (3.6). This function determines the maximum range of the uncertain parameter to keep system quadratic stability, which is the maximum quadratic stability area. Note that $\mu_i = \frac{1}{2}(\delta_i^-, \delta_i^+), \lambda_i = \frac{1}{2}(\delta_i^-, \delta_i^+)$, where $\delta_i(t) \in [\delta_i^-, \delta_i^+]$. The maximum quadratic stability range of closed loop system is found through estimation of θ, which satisfies the quadratic stability with all $\delta_i \in [\mu_i - \theta\lambda_i, \mu_i + \theta\lambda_i]$.

3.4 U-block model based LMI robust controller design

Due to the U-model framework, the nonlinearity of the nonlinear model is cancelled. The closed loop of U-model based pole placement control system behaves similarly to that of a linear system. Such linear systems can be easily transformed to state space representation under standard controllable principle. Therefore, the LMI based robust control system is designed in this section to demonstrate improved robustness of the developed U-model based pole placement control systems.

Consider the discrete-time system

$$x(k+1) = Ax(k) + Bu(k)$$
$$y(k) = Cx(k) + Du(k) \qquad (3.12)$$

where $x \in R^n$ is the state, $y \in R^n$ is system output.

The uncertain system can be expressed

$$x(k+1) = A_1x(k) + B_1w(k)$$
$$z(k) = C_1x(k) + Dw(k) \qquad (3.13)$$

where $w \in R^q$ is the disturbance input, $z \in R^p$ is the controlled output, and $A_1, B_1, C_{1,}$ and D_1 are uncertain matrices.

Denote the transfer function from *w* to *z* by

$$G(z) = C(zI - A^{-1})B + D \qquad (3.14)$$

Lemma 1 (de Souza et al., 1993): Supposed that the system (3.12) is stable. Then, $\|G(z)\|_\infty < 1$ if and only if there exists a matrix $P = P^T > 0$ such that

$$\begin{bmatrix} A & B \\ C & D \end{bmatrix}^T \begin{bmatrix} P & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} A & B \\ C & D \end{bmatrix} - \begin{bmatrix} P & 0 \\ 0 & I \end{bmatrix} < 0 \qquad (3.15)$$

or equivalently

$$\begin{bmatrix} -P^{-1} & 0 & A & B \\ 0 & -I & C & D \\ A^T & C^T & -P & 0 \\ B^T & D^T & 0 & -I \end{bmatrix} < 0 \qquad (3.16)$$

Accordingly to lemma 1, the uncertain system (3.8) is said to be quadratically stable with unitary H_∞ disturbance attenuation if there exists a matrix $P = P^T > 0$ such that

$$\begin{bmatrix} -P^{-1} & 0 & A_1 & B_1 \\ 0 & -I & C_1 & D_1 \\ A_1^T & C_1^T & -P & 0 \\ B_1^T & D_1^T & 0 & -I \end{bmatrix} < 0 \tag{3.17}$$

The H∞ state space model adopted for U-pole placement control system is shown in Figure 3.2 and can be expressed as (Yu, 2002):

$$\begin{aligned} x(k+1) &= Ax(k) + B_1 w(k) + B_2 u(k) \\ z_{\inf}(k) &= C_{\inf} x(t) + D_{\inf 1} w(k) + D_{\inf 2} u(k) \\ y(k) &= C_o x(k) + D_{o1} w(k) + D_{o2} u(k) \end{aligned} \tag{3.18}$$

where $u(k), w(k), y(k), z_{\inf}(k)$ is the discrete-time state variable, $u(k), w(k), y(k), z_{\inf}(k)$ are the discrete-time robust controller output, system disturbance input, system output, and disturbance output respectively. By the way, A is the state matrix, B_1, B_2 are the disturbance input matrix and regulated input matrix respectively, $C_{\inf}, C_e$ are the H∞ output matrix and system output matrix. Matrices D ($D_{\inf 1}, D_{\inf 2}, D_{o1}, D_{o2}$) with different subscribers are real matrices with proper dimension for the system.

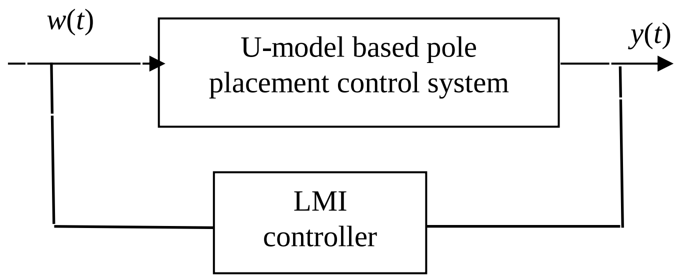

Figure 3.2 Block diagram of enhanced LMI control system

As mentioned by (3.18), a H∞ output feedback controller $K(z)$ should be designed to force the closed loop system to have the performance of asymptotic stability. The state space expression for $K(z)$ can be presented as:

$$x_k(k+1) = A_k x_k(k) + B_k u(k)$$
$$u(k) = C_k x_k(k) + D_k y(k) \tag{3.19}$$

where x_k is the state variable and A_k, B_k, C_k, D_k are unknown H∞ output feedback controller matrices. Combining (3.18) with (3.19), the closed loop system can be expressed as:

$$x_{ct}(k+1) = A_{ct} x_{ct}(k) + B_{ct} w(k)$$
$$z_{\inf ct}(k) = C_{\inf ct} x_{ct}(k) + D_{\inf ct} w(k) \tag{3.20}$$

where the state vectors and closed loop metrics are expressed as:

$$x_{ct}(k) = \begin{bmatrix} x(k) \\ x_k(k) \end{bmatrix},$$

$$A_{ct} = \begin{bmatrix} A + B_2 D_k C_o & B_2 C_k \\ B_k C_o & B_k D_{o1} \end{bmatrix}, \quad B_{ct} = \begin{bmatrix} B_1 + B_2 D_k D_{o1} \\ B_k D_{o1} \end{bmatrix}$$

$$C_{\inf ct} = \begin{bmatrix} C_{\inf} + D_{\inf 2} D_k C_o & D_{\inf 2} C_k \end{bmatrix}, \quad D_{\inf ct} = D_{\inf 1} + D_{\inf 2} D_k D_{o1}$$

The H∞ output feedback controller should be designed to take H∞ performance ($\|T_{cl\infty}(z)\|_\infty < \gamma_1$) into consideration, where $\|T_{cl\infty}(z)\|_\infty$ is the H∞ norm of the transfer function from w to $z_{\inf ct}$, and γ_1 is the upper bound of $\|T_{cl\infty}(z)\|_\infty$. Such output feedback controller ought to be designed to make the system have an acceptable H∞ norm form w to $z_{\inf ct}$ keeping the system robustness.

In order to enhance the performance of the U-pole placement control system, LMI is applied to address the solvability of discrete-time H∞ robust control system design problems. With the theorem of γ-suboptimal controller for discrete-time, Gahinet and Apkarian consider a proper discrete-time plant realisation (3.19), and assume that plants (A, B_2, C_0) is stabilizable and detectable and $D_{02} = 0$ (Gahinet and Apkarian, 1994).

Let W_{12} and W_{21} denotes basis of null spaces of $(I - D^{+}_{\inf 2} D_{\inf 2}) B_2^T$ and $(I - D_{o1} D^{+}_{01}) C_0$, where $D^{+}_{\inf 2}$ and D^{+}_{01} are respectively for the Moore-Penrose pseudo inverse of matrix $D_{\inf 2}$ and D_{o1}.

The discrete-time γ-suboptimal H∞ problem is solvable if and only if there exist symmetric matrices *R*, *S* satisfying the following LMI system (Gahinet and Apkarian, 1994):

$$\begin{bmatrix} N_R & 0 \\ 0 & I \end{bmatrix}^T \begin{bmatrix} ARA^T - R & ARC_{\inf}^T & B_1 \\ C_{\inf} RA^T & -\gamma_1 I + C_{\inf} RC_{\inf}^T & D_{\inf 1} \\ B_1^T & D_{\inf 1}^T & -\gamma_1 I \end{bmatrix} \begin{bmatrix} N_R & 0 \\ 0 & I \end{bmatrix} < 0 \tag{3.21}$$

$$\begin{bmatrix} N_S & 0 \\ 0 & I \end{bmatrix}^T \begin{bmatrix} A^T SA - S & A^T SB_1 & C_{\inf}^T \\ B_1^T SA & -\gamma_1 I + B_1^T SB_1 & D_{\inf 1} \\ C_{\inf} & D_{\inf 1}^T & -\gamma_1 I \end{bmatrix} \begin{bmatrix} N_S & 0 \\ 0 & I \end{bmatrix} < 0 \tag{3.22}$$

$$\begin{bmatrix} R & I \\ I & S \end{bmatrix} \geq 0 \tag{3.23}$$

where N_R and N_s respectively denotes basis of the null spaces of $(B_2^T, D_{\inf 2}^T)$ and (C_2, D_{01}).

Lemma 2 (Zhai et al, 2001) the following statements are equivalent:

(*i*) A is Schur stable and $\left\| C(zI - A)^{-1} B + D \right\|_\infty < \gamma_1$

(*ii*) The desired H∞ controller exists if and only if there are matrices P and K positive definite solution P to the LMI:

$$\begin{bmatrix} -P & PA_{cl} & PB_{cl} & 0 \\ A_{cl}^T P & -P & 0 & C_{cl}^T \\ B_{cl}^T P & 0 & -\gamma I & D_{cl}^T \\ 0 & C_{cl} & D_{cl} & -\gamma I \end{bmatrix} < 0 \tag{3.23}$$

where $K = \begin{bmatrix} A_k & B_k \\ C_k & D_k \end{bmatrix}$. The LMI (3.23) is a BMI with respect to *P* and *K*. The controller matrices A_k, B_k, C_k, D_k can be obtained by solving bilinear matrix inequalities (3.23) mentioned in the lemma.

3.5 Case studies

In the case studies, a Hammerstein model is selected for the robust stability test. The closed loop characteristic equation is specified with

$$A_c = q^2 - 1.3205q + 0.4966 \qquad (3.24)$$

Therefore the closed loop poles are a complex conjugate pair of $-0.6603 \pm j0.2463$. This design specification corresponds to a natural frequency of 1 rad/sec and a damping ratio of 0.7. To achieve zero steady-state error, specify

$$T = A_c(1) = 1 - 1.3205 + 0.4966 = 0.1761 \qquad (3.25)$$

For the polynomials *R* and *S*, specify

$$\begin{aligned} R &= q^2 + r_1 q + r_2 \\ S &= s_0 q + s_1 \end{aligned} \qquad (3.26)$$

Substitute the specifications of (3.24) and (3.26) into Diophantine equation of (2.6), the coefficients in polynomials *R* and *S* can be expressed with

$$\begin{aligned} r_2 + s_1 &= 0.4966 \\ r_1 + s_0 &= -1.3205 \end{aligned} \qquad (3.27)$$

To guarantee the computation convergence of the sequence *U*(*t*), that is to keep the difference equation with stable dynamic response, let $r_1 = -0.9$ $r_2 = 0.009$. This assignment corresponds the characteristic equation of *U*(*t*) as $(q - 0.89)(q - 0.01) = 0$. Then the coefficients in polynomial *S* can be determined from the Diophantine equation of (3.27)

$s_0 = -0.4205 \quad s_1 = 0.4876$

Substituting the coefficients of the polynomials *R* and *S* into controller of (2.5), gives rise to:

$$U(t+1) = 0.9U(t) - 0.009U(t-1)$$

$$+0.1761w(t-1) + 0.4205y(t) - 0.4876y(t-1) \qquad (3.28)$$

Therefore the controller output *u*(*t*) can be determined in the following way:

Consider the following Hammerstein model (Zhu and Guo, 2002)

$$y(t) = 0.5y(t-1) + x(t-1) + 0.1x(t-2)$$
$$x(t) = 1 + u(t) - u^2(t) + 0.2u^3(t) \qquad (3.29)$$

The corresponding control oriented model is obtained from formulation (2.2)

$$y(t) = \lambda_0(t) + \lambda_1(t)u(t-1) + \lambda_2(t)u^2(t-1) + \lambda_3(t)u^3(t-1) \qquad (3.30)$$

where

$$\lambda_0(t) = 0.5y(t-1) + 1 + 0.1x(t-2) \qquad \lambda_1(t) = 1$$
$$\lambda_2(t) = -1 \qquad \lambda_3(t) = 0.2$$

The system response under the proposed pole placement control has been discussed in (Zhu and Guo, 2002). It can be seen from the simulation results that the resultant closed loop system behaves similarly to that of a linear system, which is due to the cancellation of the nonlinearity by the proposed control-oriented model and controller design approach.

However, if the internal parameter of the nonlinear model is changed, the controller performance will not be same standard and that is the purpose of using a robust controller which is going to be studied in the simulations.

In the following simulation, the LMI based H_∞ output feedback controller is tested to improve the system performance of the designed U-pole placement control system. To the selected Hammerstein model, the variation of the internal parameter is the change of $\lambda_j(t)$. The characteristic equation of the LMI based H_∞ robust controller (step 2) can be expressed as:

$$A_{rc}(z) = z^2 + 0.4084z + 0.1452 \qquad (3.31)$$

This equation is obtained based on closed loop equation (3.24) by using Matlab LMI toolbox solver.

The controller is going to be applied for all cases in the U-model system simulations.

The case I: For the selected Hammerstein model, the time varying parameter $\alpha_0(t)$ was determined as the unknown parameter. The variation of the parameter $\alpha_0(t)$ was selected as

$$\alpha_0(t) \in \left[0.1y(t-1)+1+0.1x(t-2),\ \ 0.8y(t-1)+1+0.1x(t-2)\right] \qquad (3.34)$$

After the least squares data fitting, the closed loop characteristic equation is obtained as

$$A_c' = q^2 + \alpha_1 q + \alpha_2 \qquad (3.35)$$

where the parameters can be expressed as:

$$\begin{aligned} A_c'(1) &= q^2 + 0.6883q + 0.4336 \\ A_c'(2) &= q^2 + 1.7580q + 1.4700 \end{aligned} \qquad (3.36)$$

The variation range of the parameters are $\alpha_1 \in [0.6883,\ \ 1.7580]$ and $\alpha_2 \in [0.4436,\ \ 1.4700]$ respectively. The reference input and the plant output of the system ($\alpha_0(t) = 0.8y(t-1)+1+0.1x(t-2)$) are shown in figure 3.2

The result of the robust stability margin is 1.1239 which indicates that the U-model controller can guarantee the system stability within the selected range of $\alpha_0(t)$ and even if the range is increased to 12.39%.

The case II: For the same Hammerstein model, the time varying parameter $\alpha_0(t)$ was still selected as the unknown parameter. The different variation of the parameter $\alpha_0(t)$ was

$$\alpha_0(t) \in \left[0.5y(t-1)+1+0.1x(t-2),\ \ 1.3y(t-1)+1+0.1x(t-2)\right] \quad (3.37)$$

The closed loop characteristic equation can be obtained by the least squares data fitting method, the characteristic equation expression is

$$A_c' = q^2 + \alpha_1 q + \alpha_2 \quad (3.38)$$

The estimation results are respectively

$$\begin{aligned} A_c'(1) &= q^2 + 0.0752q + 1.9010 \\ A_c'(2) &= q^2 + 0.4229q + 0.8926 \end{aligned} \quad (3.39)$$

The variation range of the parameters are $\alpha_1 \in [0.0752,\ \ 0.4229]$ and $\alpha_2 \in [0.8926,\ \ 1.9010]$ respectively. The reference input and the plant output of the system are shown in figures 3.3 and 3.4. Figure 3.3 shows the U-pole placement control system is in the bound of the robust stability area. And figure 3.4 shows the U-pole placement control system is unstable in the range of the internal parameter variation.

The result of the robust stability margin is 0.4139 which indicates that the U-model controller can only guarantee the system stability within 41.39% of the selected parameter range and in the other 58.61% the closed loop system is not stable with the designed U-model controller.

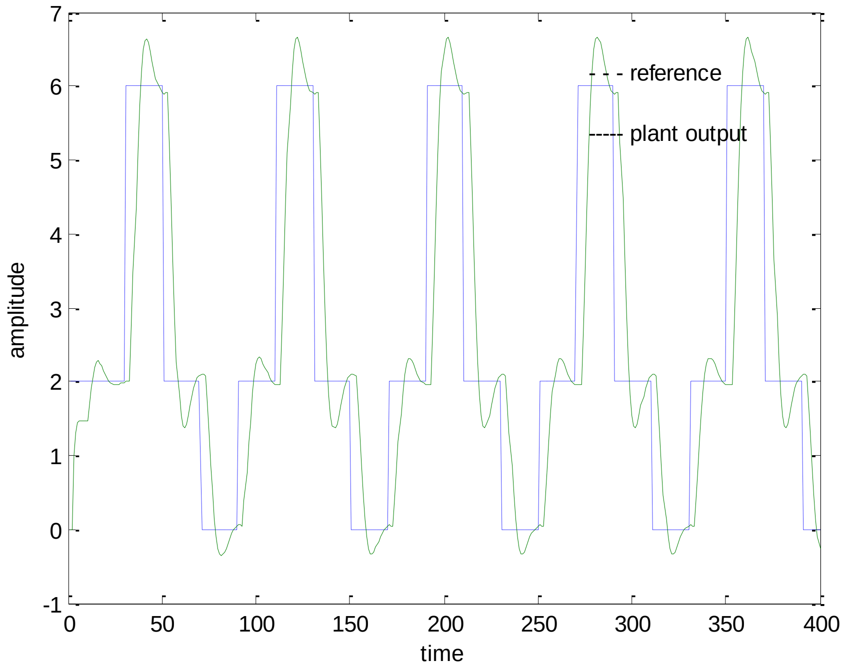

Fig. 3.3 Performance of Hammerstein model in case I

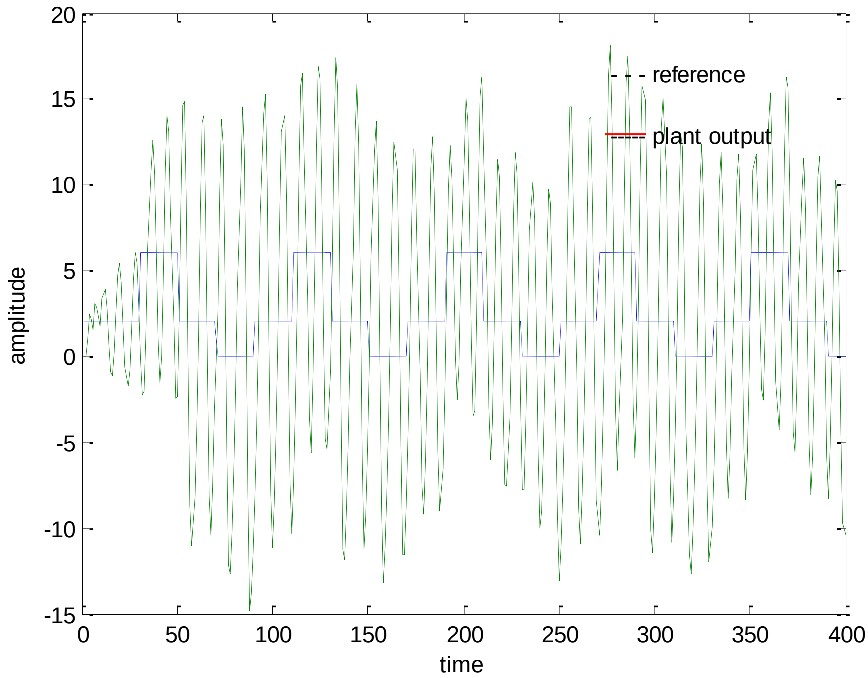

Fig. 3.4 Performance of Hammerstein model in case II

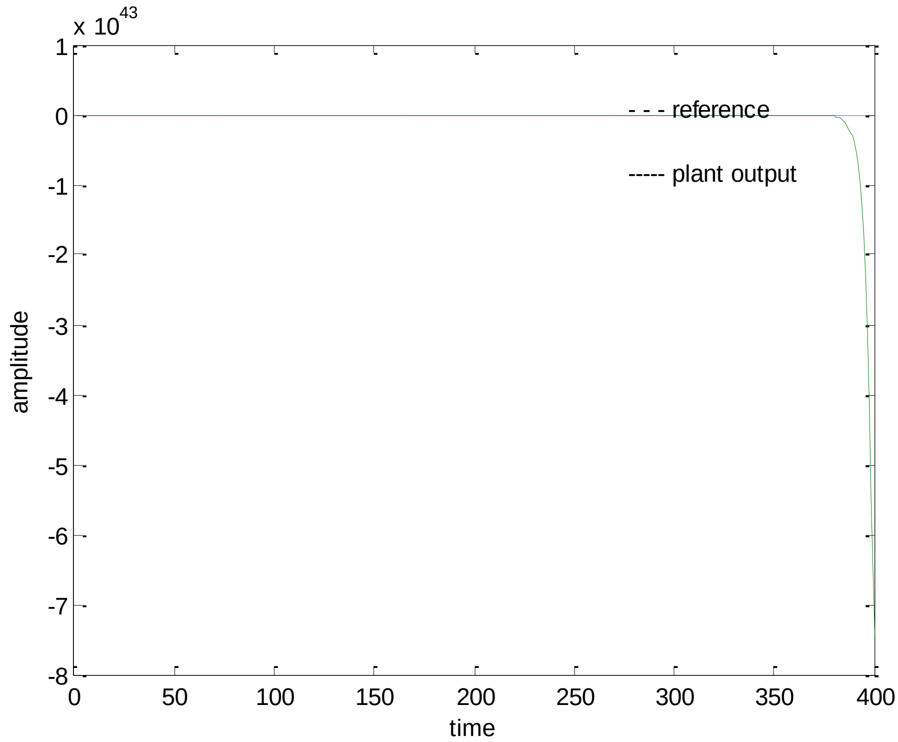

Fig. 3.5 Performance of Hammerstein model in case II

The simulation results show that the robustness of the U-model based pole placement control system depends on the uncertainty of the non-linear model. The U-model controller can keep the system to be stable within a certain range of the parameter uncertainty. However, if the parameter of the non-linear model is changing far away from the original one, the performance of the controller cannot be guaranteed.

The case III: For the selected model with uncertainty, that is the internal parameter $\lambda_j(t)$ is changed to

$$\lambda_0(t) = 1.1y(t-1) + 3 - x(t-2) \quad (3.40)$$

The disturbance system output of the U-model based pole placement control system before and after robust controller applied are shown in Figure 3.6. The output of the U-model pole placement control system without plant uncertainty is also shown in Figure 3.6.

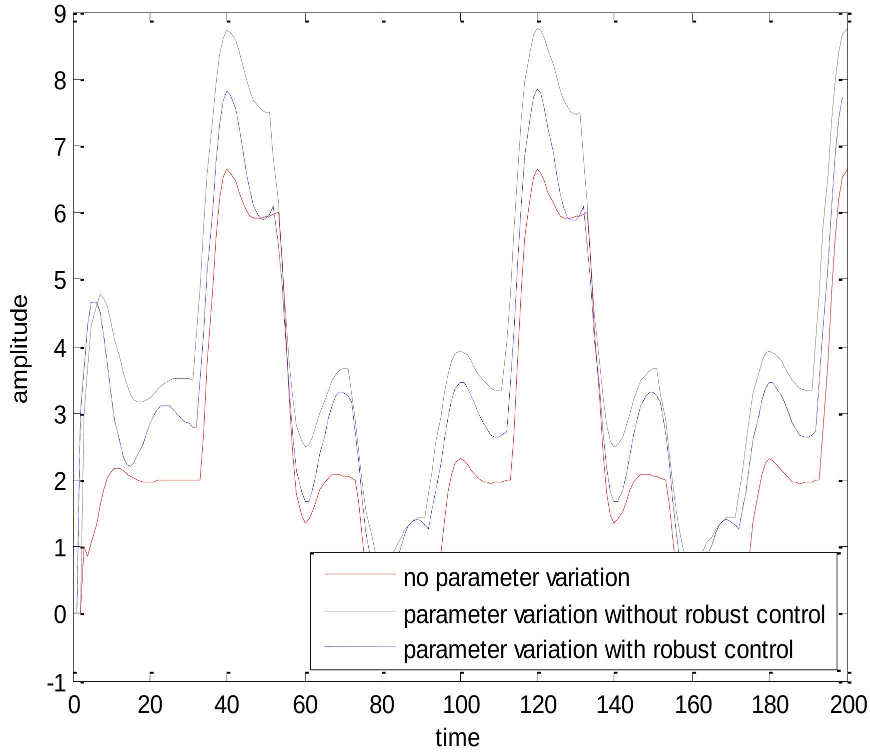

Figure 3.6 System output after internal parameter changed – case I

Deriving from the simulation result, the system performance after internal parameter variations have been improved by the designed robust controller. The amplitude of the output decreased from 8.8 to 7.6 compared with the case without a robust controller. The overshoot is reduced in this simulation which is closer to reference input.

The case IV: In the other different case, that is the internal parameter $\lambda_j(t)$ is changed to

$$\lambda_0(t) = -0.2y(t-1) + 1 - 0.1x(t-2) \qquad (3.41)$$

The closed loop system becomes unstable and the output of the system without robust control is shown in Figure 3.7.

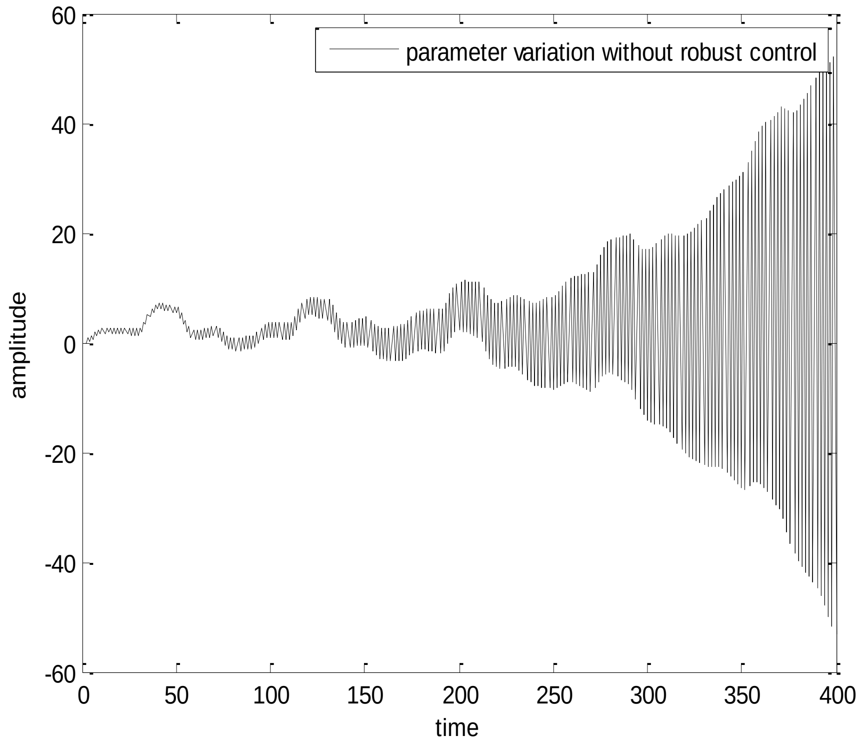

Figure 3.7 System output after internal parameter changed – case II (No robust controller)

While the LMI based robust controller is applied to the closed loop system, acceptable simulation results can be achieved. Figure 3.8 shows the system output response.

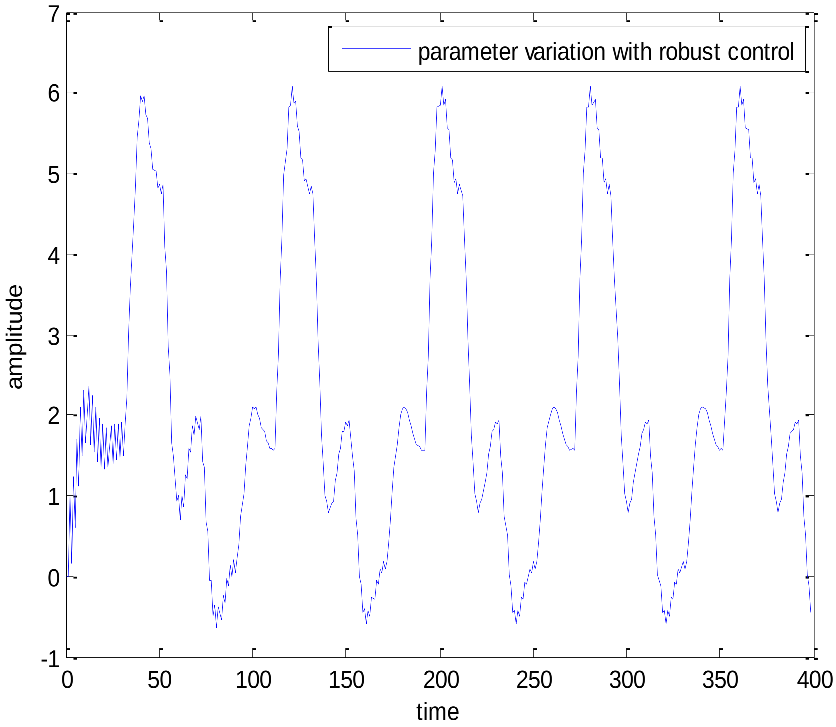

Figure 3.8 System output after internal parameter changed – case II (With robust controller)

It can be inspected from the figure that the system can be stabilised with the help of the robust controller. In another aspect, the variation of the internal parameter exceeds the stability margin (Peng et al, 2013) of the U-model pole placement control system itself. However, with the help of the LMI based robust controller, the stability margin of the system is enlarged and such internal parameter variation can be guaranteed with a stable performance.

3.6 Conclusions

Firstly, this chapter introduces a so-called U-block model, which is defined as a linear input-output model from the closed loop of U-model based pole placement control system. Then the robust stability margin of this control system is analysed by determined LMIs against uncertainty. The LMI based robust control system is designed for U-model based pole placement control system to improve the robust stability margin. Finally, the numerical simulation results are given to show the proposed approaches effective.

The simulation results for both cases show that the robustness of the robust controller design for U-model based pole placement control system is effective. The U-model controller can keep the system to be stable within a certain range of the parameter uncertainty. However, if the parameter of the nonlinear model is changed far away from the original one, the performance of the controller cannot be guaranteed. At this time, the LMI based robust controller can help to main its stability in a relatively large range of uncertainty.

In the next chapter, U-state space platform is established to expand the U-model methodology from polynomial model into state space model domain. Based on U-state space model, the feedback control system is designed on the enhanced development of using linear design approach for nonlinear state space model.

Chapter 4

U-State Space Control System Design and Analysis

4.1 Introduction

Linear control system design approaches for state space model have been well studied in research and successfully adopted in the wide ranges of industrial applications (Maciejowski 1994; Ogata 2009). In the linear case, most methodologies require the controlled object to be expressed as the linear state space model realisation. For linear polynomial models, there are also many approaches to convert the model into the equivalent state-space expression, such as two standard state space realisations which are called controllable and observable. The intrinsic properties and stability are clearly defined and analysed in linear control system. Whilst nonlinear polynomial models are very practical and predominantly used model structure in practice (Billings 2013; Zhu, Wang, Zhao, Li, and Billings 2015), they can be very difficult to convert it into a state space expression, nevertheless almost impossible to convert into an equivalent linear state space model (Zhu, Zhao and Zhang, 2016).

There are various classical tools of nonlinear state space control system design, including linearisation, integral control, gain scheduling as well as feedback linearisation. The most popular method of nonlinear state space design is linearisation (approximation around an equilibrium point). For a general nonlinear system, the most practical way to approach the stabilisation problem for nonlinear systems is to obtain neat results in the linear case via linearisation. The stabilising linear state feedback control law is then designed for the linearized system about the desired equilibrium point.

4.1.1 Nonlinear polynomial control

There is one approach to treat nonlinear polynomial models as time varying linear models. One of the typical approaches is the state-dependent parameter (SDP) transformation (Taylor, Chotai, and Young 2009, Çimen 2010), which reduces the closed-loop system to a linear transfer function with the specified (design) poles. Hence, assuming pole assignability at each sample, global stability of the non-linear system is guaranteed at the design stage.

It is clear from these studies that a common strategy for the control of non-linear systems involves attempting to bring the original system into a quasilinear domain, before subsequently designing an appropriate linear control algorithm. In model structure, the parameters of this quasi-linear State-Dependent Parameter (SDP) are functionally dependent on other variables in the system (Young 2000). Although this provides a bridge from nonlinear polynomial model to linear time-varying state space expression, the transform is not unique and selection of SDP is subjective, personal with no clear rule to follow, which has been found to be the main barrier for a wide range of users.

4.1.2 Feedback linearisation

The other popular method of nonlinear state space control system design is feedback linearisation (input state linearisation and input output linearisation), which has attracted a great deal of research interest in recent years. The central idea of the approach is to algebraically transform a nonlinear system dynamics into a (fully or partly) linear one, so that linear control techniques can be applied. This differs entirely from conventional linearisation (i.e., Jacobian linearisation) in that feedback linearisation is achieved by exact state transformations and feedback, rather than by linear approximations of the dynamics. This approach is based on the transforming of nonlinear dynamics into a linear form by state feedback, so as to cancel the nonlinear dynamics in the designed control input or establish a linear input output by coordinate transform (Isidori, 1995); linear state space approaches can then be applied to design the corresponding control systems. However, this is a case by case approach with certain degree of skills required in manipulating different equations and selecting coordinates. Compared with the previous linearisation method, this method is exact because of perfect knowledge of the state equation(s); the nonlinearities of the systems can then be cancelled by this knowledge. However, in reality, perfect knowledge of the state equation and exact mathematical cancellation in terms are impossible to implement in practical applications.

The idea of simplifying the form of a system's dynamics by choosing a different state representation is not entirely unfamiliar. In mechanics, for instance, it is well known that the form and complexity of a system model depend considerably on the choice of reference frames or coordinate systems. Feedback linearisation techniques can be viewed as a way of transforming original system models into equivalent models of a simpler form. Thus, they can also be used in the development of nonlinear control systems in combination with robust or adaptive controllers.

Feedback linearisation has been successfully applied to address some practical control problems. These include the control of helicopters, high performance aircraft, industrial robots, and biomedical devices (Slotine and Li, 1991). More applications of this methodology are being developed in industrial applications. However, there are also a number of important shortcomings and limitations associated with the feedback linearisation approach. Such problems are still very much topics of current research. In its simplest form, feedback linearisation amounts to cancelling the nonlinearities in a nonlinear system so that the closed-loop dynamics is in a linear form. For an example (shown in

Figure 4.1), the closed-loop system with the feedback linearisation control law is represented in the block diagram. It can be inspected by two loops in this control system, with the inner loop achieving the linearisation of the input-state relation, and the outer loop achieving the stabilisation of the closed-loop dynamics.

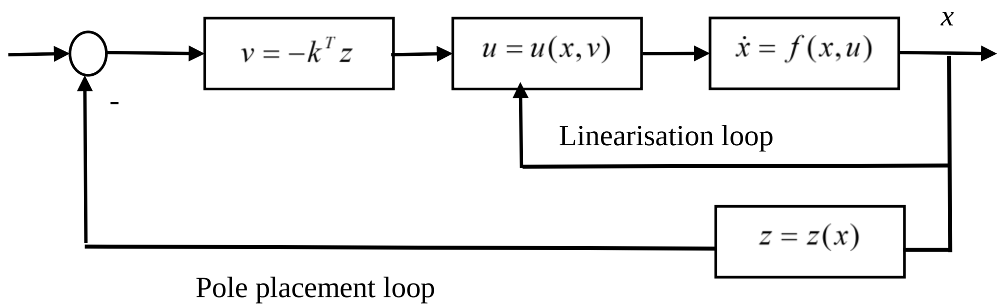

Figure 4.1 Feedback linearisation block diagram (Slotine and Li, 1991)

A number of remarks can be made about the feedback linearisation design approach:

a) With feedback linearisation control, it can be valid in a large region of the state space systems. However, in some particular cases, the controller does not work properly such as when the initial states equal zero. This is due to the state variables dependent relationship for obtaining controller output.

b) In order to implement the control law, the new state components must be available. If they are not physically meaningful or cannot be measured directly, the original state variable must be measured or estimated. Thus, the new state variables can be computed by the original state vectors.

c) In general, the system model is relied on both for the controller design and for the computation of z . If there is uncertainty in the model such as uncertainty on the parameter and parameter variation, this inaccuracy of system model will result in an error in the computation of both the new state variable z and of the control input u .

The feedback linearisation method has a number of significant limitations (Slotine and Li, 1991):

- Feedback linearisation cannot be used for the design of every kind of nonlinear system. The full state variables have to know or to be appropriately measured. The robustness is difficult to guarantee in the presence of parameter uncertainty or unmodelled dynamics.
- The second problem is due to the difficulty of finding convergent observers for nonlinear systems and, when an observer can be found, the lack of a general separation principle (analogous to that in linear systems) which guarantees that the straightforward combination of a stable state feedback controller and a stable observer will guarantee the stability of the closed-loop system.
- The third problem is due to the fact that the exact model of the nonlinear system is not available in performing feedback linearisation. The sensitivity to modelling errors may be particularly severe when the linearising transformation is poorly conditioned.

Active research is being performed to overcome the above drawbacks listed. One potential method is through the extension of the U-model methodology from polynomial model design into state space field. This will establish a generic systematic approach to convert the nonlinear state space model into a controller output based time varying model. This has been studied by Zhu et. al. (Zhu et al., 2014; Zhu, 2016; Zhu and Guo, 2002) for facilitating nonlinear control system designs over the last decade. Consequently, linear polynomial model based design approaches (such as pole placement and general predictive control can be directly used to design such nonlinear control systems (Du et al., 2012; Zhu and Guo, 2002). The state space based U-control system research progression aims to progress the development of the U-model approach to enable the use of linear state space based design approaches to allow for the feedback control of nonlinear state space models described plants (Zhu, 2016). The key challenge identified in the on-going study is how to obtain an initial state space expression within the reference U-model framework from its original nonlinear state space model.

In this chapter, the new U-state space framework is established in section 4.2. The linear state space design approach is then applied to design for nonlinear models within proposed U-state space prototype in section 4.3. In the following section, the stability analysis of development on U-state space control system is discussed. The case studies of the proposed approach are demonstrated and validated in section 4.5 and finally, conclusions are presented in section 4.6.

4.2 U-State Space Frameworks

4.2.1 Linear U-State Model

In chapter 3, U-block model was introduced as a linear input output model from the conversion from U-control system for nonlinear polynomial plants. Thus, the linear U-state model can now be defined as the equivalent state space realisation of U-block model.

Let the assigned poles as $\alpha_1 \cdots \alpha_k$ then the closed loop characteristic equation is:

$$A_c = (q-\alpha_1)\cdots(q-\alpha_k) = q^k + a_{c1}q^{k-1} + \cdots a_{ck} = 0 \qquad (4.1)$$

Correspondingly, the U-block model can be expressed as a transfer function realisation:

$$\frac{Y(q)}{W(q)} = \frac{A_c(1)}{A_c} = \frac{A_c(1)}{q^k + a_{c1}q^{k-1} + \cdots a_{ck}} \qquad (4.2)$$

Consequently, the standard linear state space form is:

$$\begin{aligned} x(t+1) &= Ax(t) + Bw(t) \\ y(t) &= Cx(t) \end{aligned} \qquad (4.3)$$

From the controllable realisation (Ogata, 2009), the state equation is:

$$\begin{bmatrix} x_1(t+1) \\ x_2(t+1) \\ \vdots \\ x_{k-1}(t+1) \\ x_k(t+1) \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & & & & \\ & 0 & 0 & \cdots & 1 \\ -\alpha_{ck} & -\alpha_{c(k-1)} & -\alpha_{c(k-2)} & \cdots & -\alpha_{c1} \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \\ \vdots \\ x_{k-1}(t) \\ x_k(t) \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix} u(t) \qquad (4.4)$$

and the output equation:

$$y(t) = \left[A_c(1), \quad 0, \quad \cdots, \quad 0, \quad 0\right] \begin{bmatrix} x_1(t) \\ x_2(t) \\ \vdots \\ x_{k-1}(t) \\ x_k(t) \end{bmatrix} \quad (4.5)$$

The linear U-state is a linearised state space description with the transformation from the U-block model (U-control system). It can then be used as a plant model for many different design algorithms.

4.2.2 U-State Space Model

Consider a general discrete time state space model description below, which includes all the currently studied combinational models of affine, non-affine, strict feedback, and pure feedback descriptions as its subsets (Zhu, et al., 2014):

$$\begin{aligned} & x(t+1) = f(x(t), \Upsilon(t), u(t)) \\ & = f_x(x(t), \Upsilon(t)) + f_u(x(t), \Upsilon(t), u(t)) \qquad (4.6) \\ & y(t) = h(x(t)) \end{aligned}$$

where $x(t) \in R^n, \quad u(t) \in R, \quad y(t) \in R$ are the state variable, system input and output respectively. $\boldsymbol{f_x(x(t), \Upsilon(t))}$ and $\boldsymbol{f_u(x(t), \Upsilon(t), u(t))}$ represent the summations of the products formed with $\boldsymbol{(x(t), \Upsilon(t))}$ and $\boldsymbol{(x(t), \Upsilon(t), u(t))}$ respectively. Further $\Upsilon(t) = \Upsilon(x(t), u(t-1), ..., u(t-n)) \in R^m$, which excludes the current model input $u(t)$. Consequently, the control oriented U state space model is defined as below:

$$\begin{aligned} & x(t+1) = f(x(t), \Upsilon(t), u(t)) \\ & = f_x(x(t), \Upsilon(t)) + U(t) \qquad (4.7) \\ & y(t) = h(x(t)) \end{aligned}$$

Rearrange state equation in (4.7) into its regression form as:

$$x(t) = \sum_{j=0}^{M} \lambda_{ij}(t) u^{j}(t-1) \qquad (4.8)$$

This is expanded from each nonlinear state equation of $x(t+1)$ as a polynomial form with respect to $u(t-1)$, where M is the degree of model input, the time varying parameters vector $\lambda_{ij} \in R^{M+1}$ is a function of past states and inputs, i is the series number of state space variable and j is the sequences of time varying parameters.

The following is an example to illustrate how to transfer the nonlinear state space model into U-state space expression. Consider control oriented total nonlinear state space model:

$$\begin{aligned} x_1(t+1) &= 0.6x_1(t)x_2(t) + 0.2x_2(t) + u(t)u(t-1) + u^2(t) \\ x_2(t+1) &= 0.3x_1(t)u(t) + 0.5x_2(t) - 0.2x_2^3(t) + u(t)u(t-2) \\ y(t) &= x_1(t) + x_2(t) \end{aligned} \qquad (4.9)$$

and the U-state space model can be determined in the notation of (4.8), thus it gives:

$$\begin{aligned} x_{d1}(\mathrm{t+1}) &= \lambda_{10}(\mathrm{t}) + \lambda_{11}(\mathrm{t})u(\mathrm{t}) + \lambda_{12}(\mathrm{t})u^2(\mathrm{t}) \\ x_{d2}(\mathrm{t+1}) &= \lambda_{20}(\mathrm{t}) + \lambda_{21}(\mathrm{t})u(\mathrm{t}) \end{aligned} \qquad (4.10)$$

where $\lambda_{10}(t) = 0.6x_1(t)x_2(t) + 0.2x_2(t)$, $\lambda_{11}(t) = u(t-1)$, $\lambda_{12}(t) = 1$, $\lambda_{20}(t) = 0.5x_2(t) - 0.2x_2^3(t)$ and $\lambda_{21}(t) = 0.3x_1(t) + u(t-2)$.

4.3 U-state Space Control System Design

A standard state feedback controller reference (Dorf and Bishop, 2011) is used to develop the following formulations for designing a full state variable feedback to achieve the desired pole locations of the closed-loop system for nonlinear state space models. The first step of design in this section assumes that all states are available for feedback which means the state vector $x(t)$ for all sampling time point t is measurable or observable. A general feedback control law is shown as:

$u = -Kx$ (4.11)

where K is determined by the feedback gain matrix and u is control input. With the system defined by the state space model, the closed loop system can be defined as:

$$x(t+1) = (A - BK)x(t) = A_d x(t) \qquad (4.12)$$

The following is a standard nonlinear discrete time system based on state space description as:

$$\begin{aligned} x(t+1) &= f(x(t), u(t)) \\ y(t) &= h(x(t)) \end{aligned} \qquad (4.13)$$

Where $x(t) \in R^n$, $u(t) \in R$, $y(t) \in R$ are the state variable, system input and output respectively. $f(\cdot,\cdot) \in R^n \times R^l$ is a nth dimensional smooth vector field.

In order to use linear state feedback design approaches, the desired state variable is defined as $x_d(t)$, which is determined by designers according to customer's requirements. The relationship between desired output $x_d(t)$ and the required controller input $u(t-1)$ is written as:

$$x_d(t+1) = A_d x(t) + B_d w(t) \qquad (4.14)$$

where A_d is the closed loop dynamic matrix and B_d is the closed loop input matrix. Accordingly, the task of the design is to determine the desired state variable $x_d(t)$ according to specified performance index A_d, B_d.

With U-state space model (4.7), in simple mathematical expression, the state equation of system (4.13) is rearranged as an equivalent expression:

$$x_{dn}(t) = \sum_{j=0}^{M} \lambda_j(t) u^j(t-1) \qquad (4.15)$$

where $\lambda_j(t)$ is time varying parameters which contains the state variable $x(t)$.

The desired closed loop state space can be written as:

$$x_d(t+1) = A_d x(t) + B_d w(t)$$
$$y(t) = C_d x(t) + D_d w(t) \qquad (4.16)$$

where A_d, B_d, C_d, D_d are the state description matrices of the desired closed loop system. A_d is determined by the closed loop characteristic equation. C_d is determined by the closed loop zeroes. The state equation is expressed as:

$$\begin{bmatrix} x_1(t+1) \\ x_2(t+1) \\ \vdots \\ x_{k-1}(t+1) \\ x_k(t+1) \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & & & & \\ & 0 & 0 & \cdots & 1 \\ -a_{dk} & -a_{d(k-1)} & -a_{d(k-2)} & \cdots & -a_{d1} \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \\ \vdots \\ x_{k-1}(t) \\ x_k(t) \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix} w(t) \qquad (4.17)$$

The output equation is expressed as:

$$y(t) = [\beta_{dk}, \quad \beta_{dk-1}, \quad \cdots, \quad \beta_{d2}, \quad \beta_{d1}] \begin{bmatrix} x_1(t) \\ x_2(t) \\ \vdots \\ x_{n-1}(t) \\ x_n(t) \end{bmatrix} + dw(t) \qquad (4.18)$$

The corresponding transfer function of (4.6) is:

$$g(z) = \frac{\beta_{d1} z^{n-1} + \beta_{d2} z^{n-2} + ... + \beta_{dn}}{z^n + a_{d1} z^{n-1} + ... + a_{dn}} + d \qquad (4.19)$$

where the closed loop characteristic equation is determined by the denominator of (4.16). Assume that the state variable $x(t)$ is measurable or obtained by a proper observer, the desired state space equation can be updated from (4.14).

As mentioned in chapter 2, the remaining design task is to resolve one of the roots of (4.15) to obtain the controller output. That is:

$$u(t-1) = \Psi^{-1}\left[x_d(t) - \sum_{j=0}^{M} \lambda_j(t) u^j(t-1) = 0 \right] \qquad (4.20)$$

where $\Psi^{-1}[*]$ is a root-solving algorithm, such as the Newton-Raphson algorithm. A detailed analysis on the root solving issues has been presented in (Zhu et al., 1999).

For the U-state space design approach, the desired plant output $x_d(t)$ is obtained by closed loop state equation (4.14), and then controller output $u(t-1)$ equation can be converted from the state equation $x(t)$ (4.15) to resolve the criterion function (4.14) to determine the designed/desired state variable $x_d(t+1)$. The controller output $u(t-1)$ can then be found through equation (4.20), that is, by resolving one of the roots of the equation (4.20). With this procedure, it only uses the state space equation, in U-state space expression, to obtain controller output $u(t-1)$ in the first stage, where the system output can be obtained by the state equation. For a nonlinear state space model, the calculation is merely to resolve one of the roots from the U-state space (4.15).

Especially, it has the following root solver for a linear desired state space equation (4.10)

$$u(t-1) = \frac{x_{dn}(t) - \lambda_0(t)}{\lambda_1(t)} \qquad (4.21)$$

In this study, it is assumed that all the state variables are known or measurable. For the future study, an appropriate observer for U-state space control system is need to develop. A potential block diagram of U-state space control system with observer is shown in Figure 4.2.

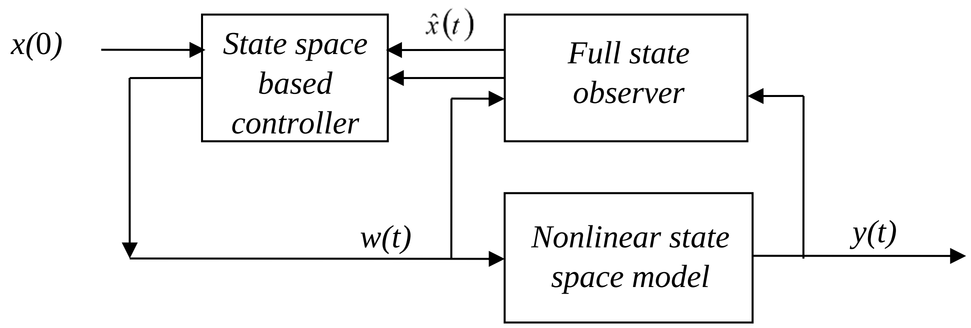

Figure 4.2 Structure of U-state space control system with observer

4.4 Stability Analysis

U-state space control systems require the means of generation of a linear differential relation between the desired states $x_d(t)$ and a control input. Specifically, we shall discuss the following issues:

- How to generate a linear input output relationship for a nonlinear system?
- What are the internal dynamics and zero-dynamics associated with the U-state space systems?
- How to design stable controllers based on U-state space control systems?

For linear systems, the stability of the internal dynamics is determined by the locations of the zeroes which can be easily found from transfer functions; where the poles of the zero dynamics are exactly of the zeroes of the system. The stability of the zero dynamics implies the global stability of the internal dynamics. However, for nonlinear systems the transfer function cannot always be precisely defined. The stability of the internal dynamics will depend on the specific control input; thus, the relation of zeroes and zero dynamics is hard to describe.

Zero dynamics is defined to be the internal dynamics of the system when the system output is kept at zero by the input (Isidori, 1995). The reason for defining and study of the zero dynamics is to find a proper and simple way of determining the stability of the internal dynamics for the design of U-state space control systems. Two remarks can be made about the zero dynamics of nonlinear systems (Slotine and Li, 1991).

a) The zero dynamics is an intrinsic feature of a nonlinear system.

b) Examining the stability of zero dynamics is much easier than examining the stability of internal dynamics. Studying zero dynamics is an effective alternative way to check the stability of the internal dynamics.

Consider a linear state space as:

$$\begin{aligned} \dot{x} &= Ax + Bu \\ y &= Cx \end{aligned} \qquad (4.22)$$

with state $x(t) \in R^n$, control input $u(t)$ and output $y(t)$. Define the tracking error as:

$$e(t) = r(t) - y(t) \qquad (4.23)$$

In U-state space control, the relationship of output $y(t)$ and control input $u(t)$ can be expressed as:

$$\dot{y} = C\dot{x} = CAx + CBu \qquad (4.24)$$

Define an auxiliary input $v(t)$ where is:

$$x_d = CBu + CAx - \dot{r} \qquad (4.25)$$

So that

$$u = (CB)^{-1}(\dot{r} - CAx + x_d) \qquad (4.26)$$

where $(CB)^{-1}$ is full rank.

The full closed-loop system is obtained by substituting the controller (4.26) into state equation (4.22):

$$\dot{x} = Ax + B(CB)^{-1}(\dot{r} - CAx + x_d) \qquad (4.27)$$

The zero dynamic is defined as $y(t) = 0$ shows that:

$$x_d = -\dot{e} = \dot{y} - \dot{r} = -\dot{r} \qquad (4.28)$$

Then yields

$$\dot{x} = \left[I - B(CB)^{-1}C \right] Ax = A_z x \qquad (4.29)$$

The error dynamics can guarantee stability through the choice of $x_d(t)$. However, there remain poles that may or may not be stable. For example, if some of these poles are non-minimum phase, then the designed closed loop system will be unstable.

Theorem 4.1 (Slotine and Li, 1991) The nonlinear system (4.13), with $f(x)$ and $g(x)$ being smooth vector fields, is input-state linearisable if, and only if, there exists a region Ω such that the following conditions hold the vector fields $\left\{ g, ad_f g, \cdots, ad_f^{n-1} g \right\}$ are linearly independent in Ω.

This condition is proposed to discuss the controllability condition for the nonlinear system. For linear systems, the vector fields $\left\{ g, ad_f g, \cdots, ad_f^{n-1} g \right\}$ become $\left\{ b, Ab, \cdots, A^{n-1} b \right\}$, and therefore their independence is equivalent to the invertibility of the familiar linear controllability matrix. From U-state space realisation (4.8), it is clearly to find that the rearrangement of state equations into time varying parameters with respect to controller input $u(t-1)$. The next time state variables are determined by previous state variables and controller input so that the U-state space model is valid for further design only if the nonlinear system is controllable.

Lemma (Slotine and Li, 1991) An n^{th} order nonlinear system is converted into U-state space if, and only if, there exists a scalar function $z_1(x)$ such that the system's with $z_1(x)$ as output function has relative degree *n*.

The relative degree of the system is to differentiate the output of a system *r* times to generate an explicit relationship between the output y and input u, then r is the relative degree. It can also be shown formally that for any controllable system of order n, it will take at most n^{th} differentiation of any output for the control input to appear, it requires $r < n$. It can be understood in an easy way. If it took more than n^{th} differentiation, it represents that the original system order higher than n. The control input never appeared or infuled with this system so that this is the uncontrollable system (Slotine and Li, 1991). Another particular case is the relative degree of a system is the same as its order ($r = n$), i.e., when the output y has to be differentiated n times (with n being the system order) to obtain a linear input-output relation. In this case, the variables $y, \dot{y}, \dots, y^{n-1}$ may be used as a new set of state variables for the system, and there is no internal dynamics associated with this input output relationship. The U-state space is still simply converted from original state equations, but it is difficult to guarantee the stability of the control systems.

In the above, the concepts of nonlinear state space system is described, which provides an interesting interpretation of the previous tracking control design based on feedback linearisation, and also yields insights about the tracking control of non-minimum phase systems.

For a SISO nonlinear state space system described as:

$$\begin{aligned} x(t+1) &= f(x(t)) + g(x(t))u(t) \\ y &= h(x(t)) \end{aligned} \qquad (4.30)$$

From above system, the initial conditions $x(0)$ and control input $u(t)$ should be in order for the plant output to track a reference output $y_r(t)$ perfectly. It can be assumed that the system output $y(t)$ is identical to the reference output $y_r(t)$. This implies that the time derivatives of all orders should be the same as those of the desired output:

$$y^k(t) = y_r^k(t) \qquad (4.31)$$

In terms of normal coordinates, the above equation can be written as:

$$\mu(t) = \mu_r(t) = \begin{bmatrix} y_r(t) & \dot{y}_r(t) & \cdots & y_r^{r-1}(t) \end{bmatrix}^T \qquad (4.32)$$

Thus, the control input $u(t)$ must satisfy:

$$u(t) = \frac{y_r - a(\mu_r, \psi)}{b(\mu_r, \psi)} \quad (4.33)$$

where $\psi(t)$ is the solution of the differential equation:

$$\dot{\psi}(t) = w[\mu_r(t), \psi(t)] \quad (4.34)$$

Given a reference trajectory $y_r(t)$, it can be used to obtain the required control input for output $y(t)$ to be identically equal to $y_r(t)$. Note that this output depends on the internal states $\psi(t)$ and thus, in particular, on the initial $\psi(t)$.

4.5 Case Studies

In this section, two nonlinear mathematical models described by state space representation are selected to test the proposed design method. An F-16 aircraft dynamic model is also applied to the proposed U-state space control system design approach. The simulation results is given to show the proposed method effective. These numerical simulations are achieved on computational test by using Matlab programming.

The desired closed loop characteristic equation is specified with

$$A_c = q^2 - 0.2q + 0.3 \quad (4.35)$$

Therefore the closed loop poles are a complex conjugate pair of $0.1 \pm 0.5385i$, which gives equivalently in continuous time domain of damping ratio 0.3980 and undamped natural frequency 1.5100 rad/s. To achieve zero steady state error, specify

$$h_0 = A_c(1) = 1 - 0.2 + 0.3 = 1.1 \quad (4.36)$$

Case I: Consider a nonlinear discrete time state space system:

$$x_1(t+1) = x_1^2(t) + x_2(t)$$
$$x_2(t+1) = x_2(t)\cos x_1(t) + u(t) \qquad (4.37)$$
$$y(t) = x_1(t)$$

Where $x_i, i = 1, 2$ are state variables, and $u(t)$, $y(t)$ are control input and system output.

According to (4.17) and (4.18), the specified closed loop standard controllable realisation of (4.1) is:

$$A_d = \begin{bmatrix} 0 & 1 \\ -0.3 & 0.2 \end{bmatrix} \quad B_d = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \quad C_d = \begin{bmatrix} 1.1 & 0 \end{bmatrix} \quad D_d = 0 \qquad (4.38)$$

It can be found that:

$$x_{d2}(t+1) = f(x_1(t), x_2(t), u(t)) \qquad (4.39)$$

From system (4.37), the desired state equation $x_{d2}(t)$ can be expressed as:

$$x_{d2}(t+1) = \lambda_0(t) + \lambda_1(t)u(t-1) \qquad (4.40)$$

where $\lambda_0(t) = x_2(t-1)\cos x_1(t-1)$ and $\lambda_1(t) = 1$.

From (4.16), the controller output is determined by

$$u(t-1) = \frac{x_{d2}(t+1) - \lambda_0(t)}{\lambda_1(t)} \qquad (4.41)$$

where the state $x_1(t), x_2(t)$ is measurable or obtained by a proper observer.

Set up the initial state variables $x_1(0) = -0.5, x_2(0) = -0.5$, and the desired state variables $x_{d1}(0) = 0, x_{d2}(0) = 0$. The simulation results are shown in the following figures.

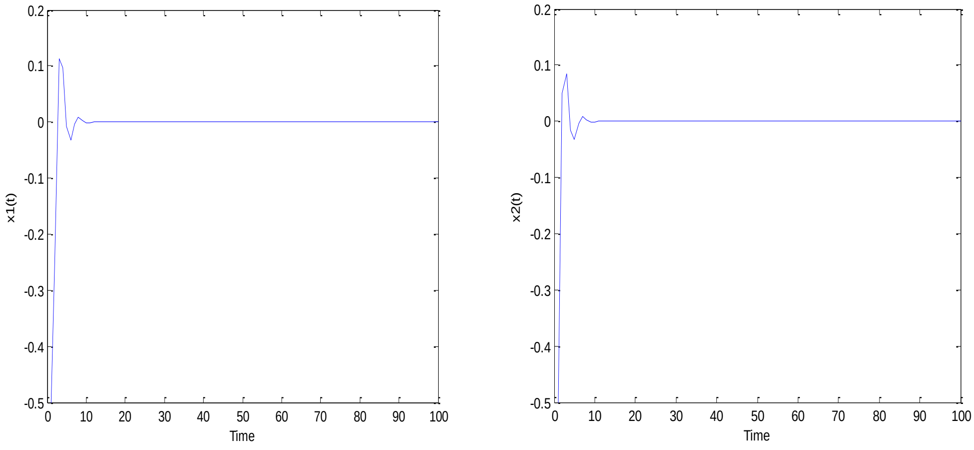

Figure 4.3 Response of state variables $x_1(t)$ and $x_2(t)$

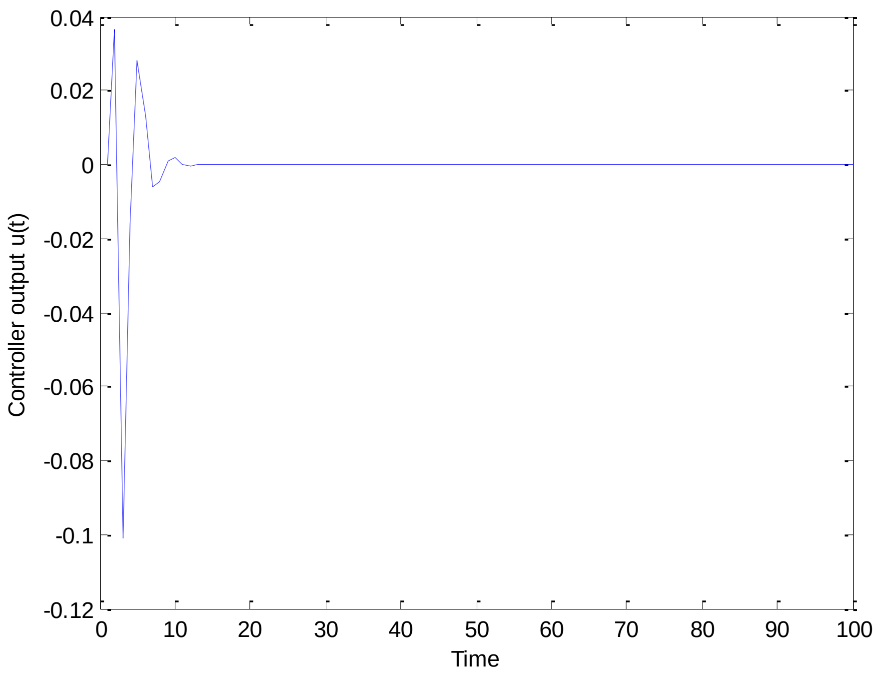

Figure 4.4 Controller output $u(t)$

It can be inspected from Figures 4.3 and 4.4 that the state variable $x_1(t)$ reaches the peak value at 5s with overshoot 0.12. After 13s, the response of state variable $x_1(t)$ settles down. Compared with $x_1(t)$, the state variable $x_2(t)$ has a lower overshoot but similar oscillation. Within the first 8s, the controller output fluctuates between 0.04 to -0.1, and it settles down at about 13s. The results of the controller output show an appropriate amplitude level and tuning profile. The simulation results give a strong indication that the proposed U-state space approach could be applied to design most practical industrial systems (subject to certain level of nonlinearity) initially, even though a lot of bench tests will be conducted in the following thorough validation work.

Case II: Consider the following nonlinear discrete time model as:

$$\begin{aligned} x_1(t+1) &= -x_1(t) + x_2(t) + \sin x_1(t) \\ x_2(t+1) &= x_2(\mathrm{t}) - x_2(t)\cos x_1(t) + \cos 2\,\mathrm{x}_1(t)u(t) \\ y(t) &= x_1(t) \end{aligned} \quad (4.42)$$

where $x(t)$ is state variable, and $u(t)$, $y(t)$ are control input and system output.

From (4.8), the desired state equation $x_{d2}(t)$ can be expressed as:

$$x_{d2}(t+1) = \lambda_0(t) + \lambda_1(t)u(t-1) \qquad (4.43)$$

where $\lambda_0(t) = x_2(t) - x_2(t)\cos x_1(t)$, $\lambda_1(t) = \cos 2x_1(t)$ $\lambda_1(t) = \cos 2x_1(t)$.

The controller output is determined by solving equation (4.20). The same initial specification in previous case is used in the simulation. The simulation results are shown in the following figures 4.5 to 4.7.

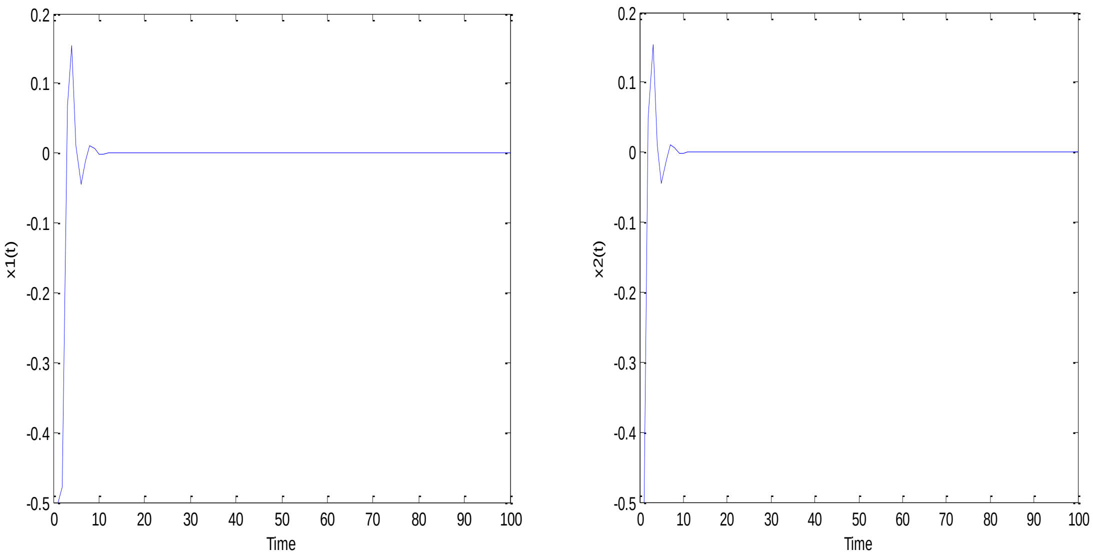

Figure 4.5 Response of state variable $x_1(t)$ and $x_2(t)$

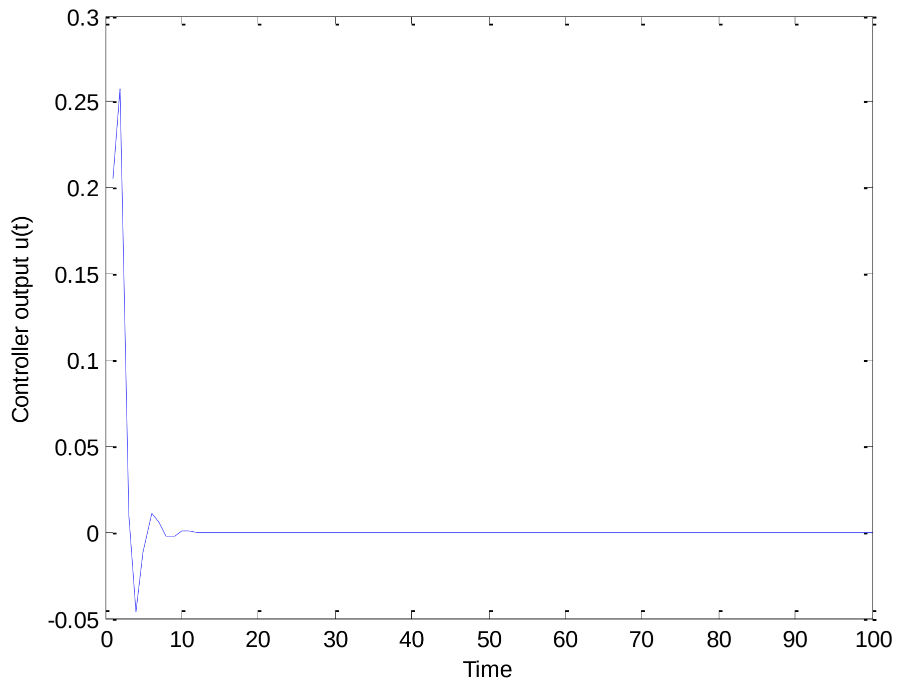

Figure 4.6 Controller output $u(t)$

It can be inspected from Figures 4.5 and 4.6 that the state variable $x_1(t)$ reaches the peak value at 4s with overshoot 0.16. After 15s, the response of state variable $x_1(t)$ settles down.

Compared with $x_1(t)$, the state variable $x_2(t)$ has a similar oscillation.

Case III: In this part, a linearised F16 dynamic model in selected to model a trim flight condition to demonstrate the U-state space control system design approach.

Now consider a F16 flight condition in straight and level flight at 502 ft/s with a cg position of $0.3\bar{c}$ (Stevens and Lewis, 2003), the determined trimmed equilibrium is presented in Table II.

TABLE 4.1 Trimmed equilibrium for $X_{cg} = 0.3\bar{c}$

V_T	502 ft/s
α	0.003936 rad
θ	0.03544 deg
δ_e	-0.0559 deg

Convert this F16 longitudinal model into a discrete time expression (4.16) with a sampling time $T = 0.01s$

$$x(k+1) = Gx(k) + Hu(k) \qquad (4.44)$$

where

$$G = \begin{bmatrix} 0.998 & 0.7654 & -3.2137 & -0.1809 \\ 0 & 0.893 & 0 & 0.0799 \\ 0 & -0.0115 & 1 & 0.093 \\ 0 & -0.2207 & 0 & 0.8606 \end{bmatrix}$$

$$H = \begin{bmatrix} -0.0422 \\ -0.045 \\ 0.0155 \\ 0.0055 \end{bmatrix}$$

The four states give rise to one complex conjugate pair of eigenvalues. The eigenvalues are

$-0.8768 \pm j0.1318$.

By using LQR design, setup the matrices Q and R. The feedback gain can be determined as $K = [-1.0745 \quad -4.373 \quad 16.8521 \quad 3.1182]$. The desired closed loop state space equation is

$$x_d(k+1) = A_d x(k) + B_d w(k) \qquad (4.45)$$

where

$$A_d = \begin{bmatrix} 0.9527 & 0.5808 & -2.5027 & -0.0492 \\ -0.0484 & 0.6962 & 0.7586 & 0.2204 \\ 0.0166 & 0.0563 & 0.7389 & 0.0447 \\ 0.0059 & -0.1968 & -0.0922 & 0.8436 \end{bmatrix}$$

It can be found that

$$x_d(k+1) = A_d x(k) = \lambda_0(k) + \lambda_1(k)u(k) \qquad (4.46)$$

Assume that the initial state variables are $x(k) = [0 \quad 12 \quad 0 \quad 0]^T$. From equation (6.1), the U-state space equation of angle of attack is

$$x_2(k+1) = \lambda_0(k) + \lambda_1(k)u(k) \qquad (4.47)$$

where $\lambda_0(k) = 0.893x_2(k) + 0.0799x_4(k)$ and $\lambda_1(k) = 1$.

From (4.34), the controller output is determined by

$$u(k) = \frac{x_{d2}(k+1) - \lambda_0(k)}{\lambda_1(k)} \qquad (4.48)$$

The simulation results are shown in figures 4.7 and 4.8. It can be inspected from following figures that the state variable delta velocity $x_1(k)$ fluctuates between 9.5 and -4 and it settles down at 60 time units (0.6s). After 50 time unit (0.5s), the state variable delta alpha $x_2(k)$ settles down from initial state 12 to steady state. The state variable delta theta $x_3(k)$

fluctuates between 1.2 and -0.35 and it settles down at 60 time units (0.6s). The state variable delta pitch rate $x_4(k)$ reaches the nadir at 5time units (0.05s) and settles down at 60 time units (0.6s).

The simulation results show the evidence that the proposed U-state space approach could be applied to design most practical linearised aircraft systems initially, the nonlinear control system based U-state space will be conducted in the following thorough justification work. The related design approach for nonlinear aircraft model is also expected to proposed and demonstrated.

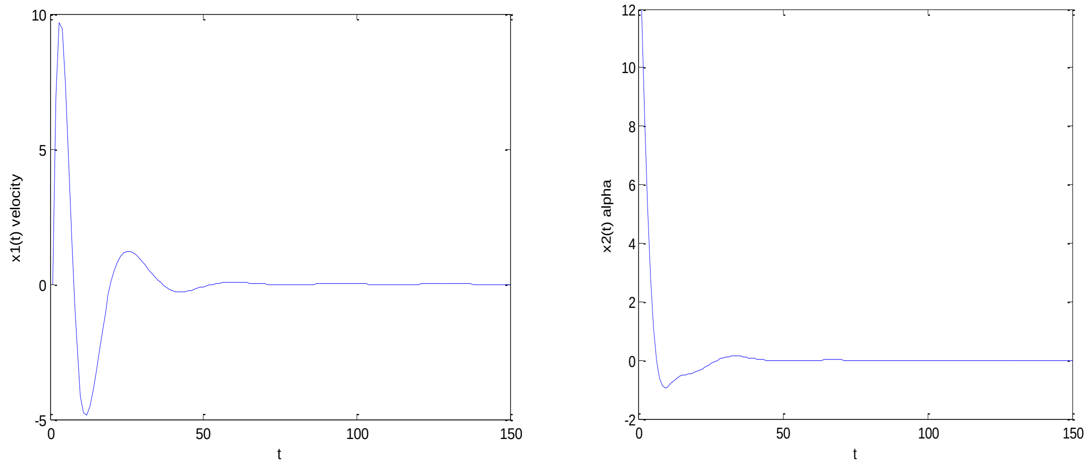

Figure 4.7 Response of state variable $x_1(k)$ velocity and $x_2(k)$ angle of attack

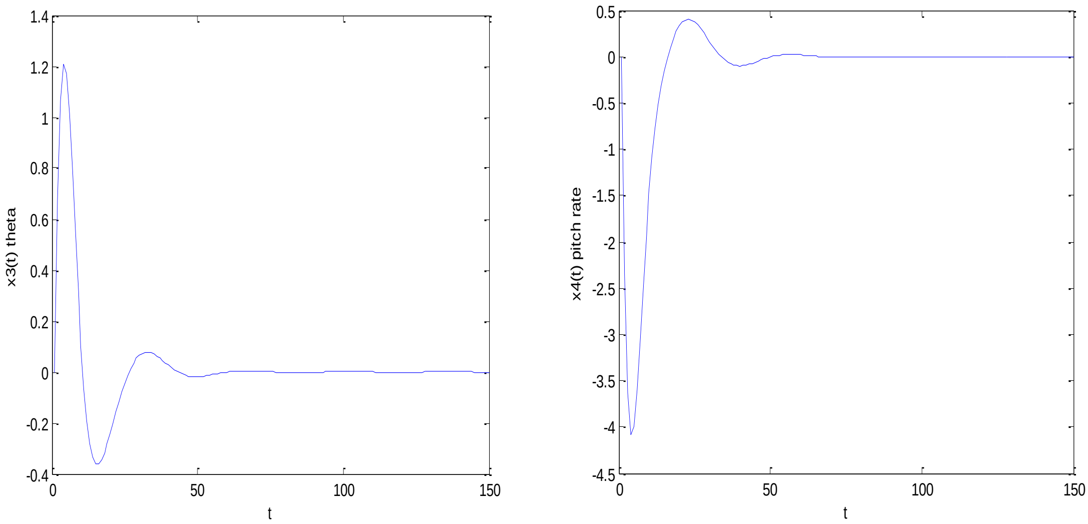

Figure 4.8 Response of state variable $x_3(k)$ pitch angle and $x_4(k)$ pitch rate

4.6 Conclusions

In this chapter, U-state space model is defined as the similar representation in terms of U-model polynomial framework. Then the linear state feedback controller design method is proposed to design on the nonlinear dynamic model directly within U-state space model. The zero dynamics of the control system is discussed to guarantee the stability of the design control system.

U-state space model design approach was originally proposed to simplify nonlinear control system design on the linear approach available U-platform, it has still advantages over classical approaches while dealing with linear control system design. One of the attraction points is to split accumulated bulk inversion into separate inversions in the design. U state space design is an approach where a feedback linearisation loop is applied to the tracking outputs of interest.

In the simulation section, two mathematical models and one F-16 aircraft model are selected to test the proposed U-state space design approach. The simulation results shows that the state variables are converged to initial values in a short response time with satisfied overshoot. The proposed approach is an effective and efficient tool for control system design.

In the next section, a quadrotor craft model is introduced to conduct the U-state space design approach for MIMO system.

Chapter 5

U-State Space Control Systems Design for a Quadrotor

5.1 Introduction

5.1.1 Quadrotor configuration and requirements

Unmanned Aerial Vehicles (UAVs) are gaining increasing interest because of a wide area of possible applications. The multi-rotor vehicle is a mechatronic system with four (or more) propellers in typically in a symmetrical configuration (shown in Figure 5.1). In this project, the quad-rotor vehicle (quadcopter) is used as the platform for the studies and discussion. The quadcopter different from a conventional helicopter, has two motors (the front and the rear) rotate clockwise, and the rest of two (the left and the right) rotate counter-clockwise. The overall thrust is the summation of the thrusts generated by the four single rotors. During trimmed flight thrust forces from motors nearly cancels gyroscopic effects and aerodynamic torques. One additional advantage of the quad-rotor is the simplified rotor mechanics structure and kinetic analysis (Voos, 2009). By varying the rotating speed of each motor, vertical and lateral motion can be controlled by changing the lift force. For example on basic rotations, the different rotating speed of a pair motors (usually it says front and rear) provide a lift force larger/smaller than gravity for the pitch movement. Another pair of motors can produce a roll motion by generating a difference in rotating speed. Yaw rotation, slightly different from above two movements, is generated due to the difference in the induced counter torque of the paired motors (front/rear and left/right).

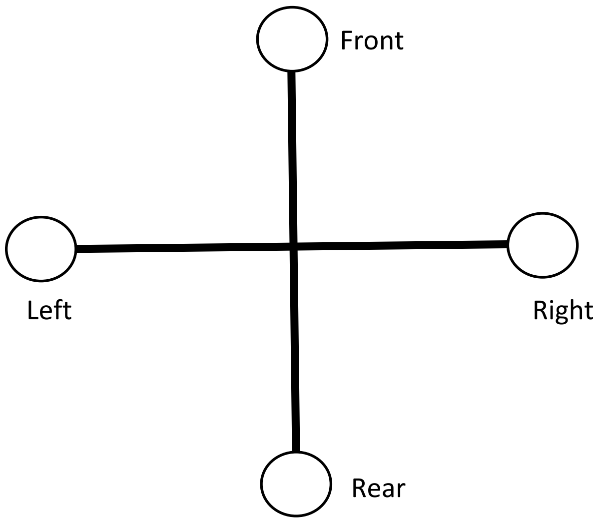

Figure 5.2 Quad-Rotor configuration

Nowadays, the quad-rotor rotorcraft is designed to operate with high agility and rapid manoeuvring. It is required to be capable of working in degraded environments such as strong and gusting wind conditions and so on (Das et al., 2008). Thus UAVs have been successfully deployed for a wide range of applications including: search and rescue operations for missing people and natural disasters, surveillance for illegal imports and exports, inspection of power lines, aerial photography for mapping, fire detection and control, tracking shooting of TV program, traffic monitoring in urban areas, crop monitoring and spraying, border patrol and atmospheric analysis for weather forecasts (Zulu and John, 2014).

The focus on the development of powerful control strategies for quadrotors has correspondingly become more and more popular as a research area. For these applications, quad-rotor control often requires holding a particular trimmed operation such as hovering or tracking targets as well as controlling the motors to provide commanded velocity and acceleration in the desired way (Koo and Sastry, 1998). Thus, it is significant to enhance the ability for quadrotors to achieve hover precisely and manoeuvre sharply.

From an attitude control aspect, it can be said that rotorcraft control is very similar to aircraft control which involves controlling the pitch, yaw, and roll motion of the rigid body. However, the pitch, yaw and roll dynamics of the rotorcraft are strongly coupled because of its unique body structure. Therefore, it is challenging to design decoupled control laws that stabilise the faster and slower dynamics simultaneously within this commonly under-actuated design configuration. In addition, the dynamics of the quadrotor are highly nonlinear and several uncertainties are encountered during its missions, which makes it a challenging venture for the related control design tasks (Lee et al., 2013).

The 6 Degree of Freedom (DoF) airframe dynamics of a typical rotorcraft involves the typical translational and rotational kinetic equations. As briefly mentioned earlier the kinetic dynamic of a quad-rotor rotorcraft is essentially a simplified form of helicopter that exhibits the same basic problems including under-actuation, strong coupling, multi-input/multi-output, and unknown nonlinearities (Das et al., 2008). The quad-rotor is classified as a rotorcraft where overall thrust is derived from the rotation speed of four motors and its movements are characterised by the resultant force and moments of the four rotors. Therefore the control algorithms designed for a quadrotor could be applied to a helicopter with relatively straightforward modifications. The control techniques applied to quadrotor has been extensively explored in academic research and as such, there are many different control algorithms successful for rotorcraft control systems design.

5.1.2 Quadrotor control

The PID controller, one of the most popular approaches in modern control engineering, has been applied to a broad range of industrial applications. The advantages of the classical PID controller is that parameters are flexible to adjust within fixed low cost structure and has good robustness (Ogata, 2009). However, the performance of PID controllers when applied to quad-rotors is variable due to several challenges including the nonlinearity associated with the mathematical model and the imprecise nature of the model due to uncertain or inaccurate mathematical modelling of some of the dynamics (Zulu and John, 2014).

A PID controller is proposed for the attitude control of a quadrotor (Lee et al., 2012). The stable hovering conditions of quadrotors is uniformly ultimately bounded for all signals by applying Lyapunov stability criteria. From the simulation and experimental work presented for this PID control system, it results in a better performance on the task of pitch angle tracking, but it can be observed that there are large steady state errors in the roll angle tracking performance. In another study, a PID controller was applied to regulate both position and orientation of a quad-rotor (Li and Li, 2011). The PID parameter gains were determined intuitively which result in a good performance during attitude stabilisation. In addition, the system performance has almost zero steady state error with slight overshoot and suitable response time. From many kinds of literature, the PID controller has been successfully applied to the quadrotor meanwhile with some limitations. For example, it is required to conduct PID tuning around the equilibrium point to demonstrate a good performance.

Most literature (Bijnens et al., 2005; Mokhtari, et al., 2006) deal with either input output linearisation for decoupling pitch, yaw, roll or backstepping to deal with the under-actuation problem. The problem of coupling in the yaw, pitch, and roll of a helicopter, as well as the problem of coupled dynamics-kinematic under-actuated system, can be solved by back-stepping (Slotine and Li, 1991). Generally, backstepping control is a recursive algorithm that breaks down the controller into steps and progressively stabilises each subsystem. The advantage is that this algorithm quickly converges and results in less computational calculations with high performance against external disturbances. However, the robustness against internal uncertainties is very poor. It is applied to stabilise a quad-rotor system consisting of an under-actuated, fully actuated and propeller subsystems (Madani and Benallegue, 2006). Combined with Lyapunov stability theory, roll and pitch angle stabilisation can be guaranteed with proper tracking performance for position and yaw angle. The backstepping approach is also applied for attitude stabilisation of a quadrotor (Huo et al., 2014). By using Lyapunov stability analysis, it is found that the closed loop system of attitude is asymptotical stable, where all the state vectors are ultimately bounded in the presence of external disturbances. To increase robustness against external disturbances, an integrator is added to the general system to develop an algorithm named Integrator Backstepping Control as articulated (Fang and Gao, 2011). The integral approach is shown to eliminate the steady state errors of the system, reduce response time and restrain overshoot of the control parameters (Zulu and John, 2014).

The most popular method is feedback linearization which is a generic nonlinear control system design approach and it is also known as Dynamic Inversion (DI). DI control is a methodology for designing closed loop control laws for nonlinear systems by which existing undesirable dynamics are cancelled out and replaced by designer specified appropriate ones through the inversion of the model dynamics. The central idea of this approach was applied to linear control techniques for those nonlinear systems whose nonlinear dynamics could be fully or partly transformed into linear one. This technique can be regarded as way of deriving simpler equivalent models from the original system model.

The nonlinear dynamics are cancelled in the closed loop and the mature linear design methods can be applied to the equivalent systems. Dynamic inversion is effective in the control of both linear and nonlinear systems and involves an inner inversion loop (similar to feedback linearization) which results in tracking if the residual or internal dynamics is stable. Typical usage requires the selection of the output control variables so that the internal dynamics is guaranteed to be stable. This implies that the tracking control cannot always be guaranteed for the original outputs of interest (Das et al., 2008).

The application of dynamic inversion on UAV’s and other flying vehicles such as missiles, fighters and aircraft are proposed in several research works. Output feedback linearization is implemented as an adaptive control strategy for stabilisation and trajectory tracking on a quadrotor with a centre of gravity that could dynamically change (Palunko and Fierro, 2011). The controller is able to stabilise the quadrotor and reconfigure it in real time when the centre of gravity changed. Feedback linearization and input dynamic inversion are implemented to design a path-following controller which allowed the designer to specify the speed profile and yaw angle as a function of displacement along the path (Roza and Maggiore, 2012). Two simulation cases with the quadrotor travelling at different speeds along the path were considered. Both cases showed the convergence of velocity and yaw angle.

In this chapter, the nonlinear quadrotor model is introduced in section 5.2. The U-state space control system design for MIMO is proposed in section 5.3. In the next section, the case studies for the U-state space control system is demonstrated though computational experiments. A conclusion of this chapter is given in the last section.

5.2 The Nonlinear Quadrotor Model

5.2.1 Quadrotor model preliminaries

For a typical quad-rotor (Figure 5.2), it is given that the front and the rear motors rotate counter-clockwise while the other two rotate clockwise, gyroscopic effects and aerodynamic torques tend to cancel in trimmed flight (Castillo et al., 2005). For the task of controlling the quad-rotor rotorcraft, the control targets are the rotation speed of each motor instead of any blade pitch control. In this way, the throttle input is obtained from the summation of the thrusts providing by motors. Pitch movement is obtained by increasing/reducing the speed of the rear motor and reducing/increasing the speed of the front motor. The roll movement is obtained similarly using the lateral motors. The yaw movement is obtained by increasing/decreasing the speed of the front and rear motors and decreasing/increasing the speed of the lateral motors (Das et al., 2008).

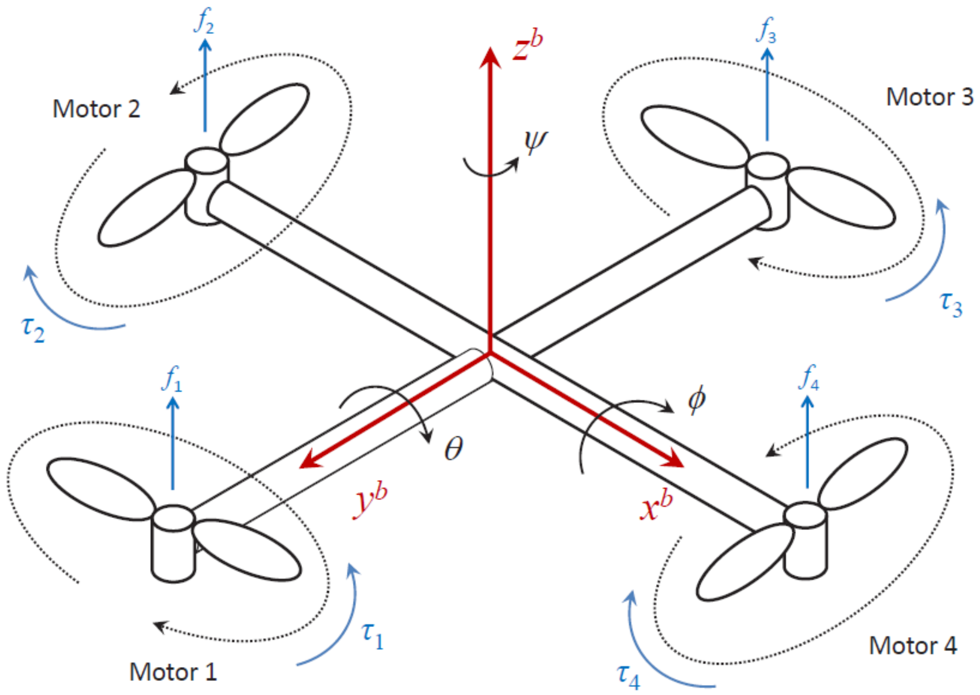

Figure 5.2 the body frame of quadrotor

This quadcopter has four identical rotors located at the corners of a square body (Figure 5.2), and its propellers or blades are connected with each rotor at a fixed angle of attack. The location can pair the rotors, and each pair of them rotates in a different direction (see the arrow in Figure 5.2). When inspecting from above view, motors 1 and 3 rotate clockwise, whereas motors 2 and 4 have a counter clockwise rotation. Especially, all the motors rotate at the same angular velocity to generate the torques τ1, τ2, τ3 and τ4. These torques are the counter torques as a consequence of the rotation of the motors. The quadcopter will cancel each other out and not spin about its vertical direction (z^b axis $\psi = 0$). The hover condition is satisfied if the total thrust (generated by the four rotors) is equal to the force of gravity.

To formalise the description on movement trajectory and attitude of the quad-rotor rotorcraft, the inertial frame and the body frame are referred. Firstly, the inertial frame is defined by the ground, with gravity pointing in the negative *z* direction. Secondly, the body frame is defined by the orientation of the quadcopter, with the rotor axes pointing in the positive z^b direction and the arms pointing in the x^b and y^b directions (Agudelo and Moor, 2014). The attitude of the quad-rotor rotorcraft is determined by three angles, normally denoted as roll (ϕ), pitch (θ) and yaw (ψ). The way of varying these angles by determining the angular velocities of the rotors is illustrated in Figure 5.3. For example, the roll and pitch angles changes are accompanied by translational movement. The reason is a quad-rotor rotorcraft is an under-actuated vehicle. There are using only 4 actuators (four rotors located in the corners) for controlling 6 degrees of freedom (including three translational positions *x*, *y*, *z* and three rotational angles ϕ, θ and ψ).

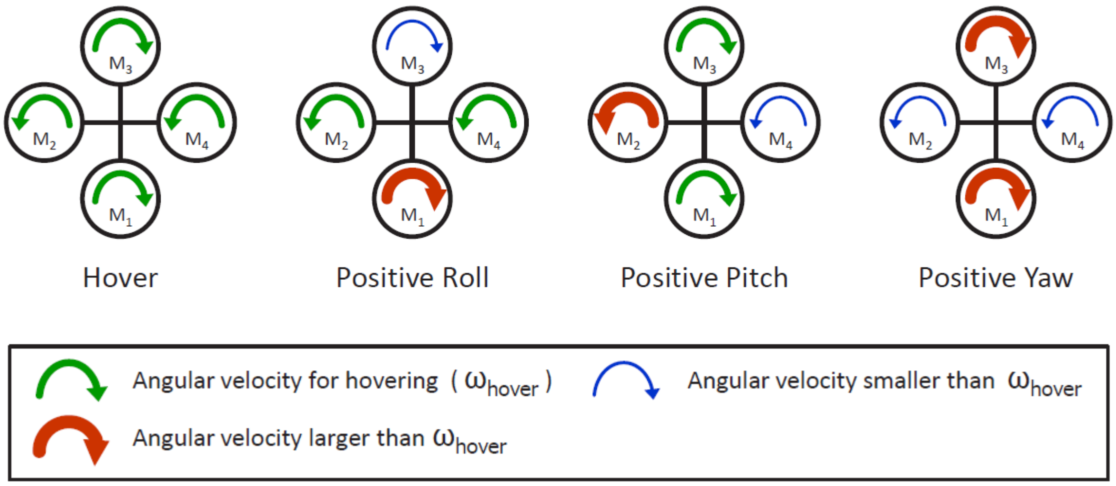

Figure 5.3 Motion of the quadrotor (Agudelo and Moor, 2014)

The angular velocity of the motors are denoted as ω_i, $i = 1, 2, 3, 4$, and the angular velocity of the quad-rotor rotorcraft in the hover condition is denoted as ω_h. From Figure 5.3, it can be inspected that the rolling motion corresponds to a rotation of the quad-rotor rotorcraft about the x^b axis. It is obtained when $\omega_2 = \omega_4 = \omega_h$ and ω_1, ω_3 are changed. For a positive rolling, the angular of velocities of motors is specified as $\omega_1 > \omega_h$ and $\omega_3 < \omega_h$. A negative rolling action is produced when the angular velocities of ω_1, ω_3 are set up in the opposite conditions.

5.2.2 Quadrotor kinematic equations

There are many existing studies to described the nonlinear dynamic model of the quad-rotor rotorcraft. Referring to previous studies, the nonlinear dynamic model is reviewed and presented in the following section.

Consider the motion of quad-rotor as described by Newton laws:

$$F_{tot} = f_1 + f_2 + f_3 + f_4 - mg = ma \qquad (5.1)$$

The force and acceleration along the coordinate directions

$$\begin{aligned} F_x &= F_{tot} \sin\theta = ma_x \\ F_y &= F_{tot} \sin\varphi = ma_y \\ F_z &= F_{tot} \cos\theta \cos\varphi = ma_z \end{aligned} \qquad (5.2)$$

The translational motion of the quad-rotor in the inertial frame is described by the following equation:

$$m\begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ -mg \end{bmatrix} + RT^b + F_D \qquad (5.3)$$

where x, y and z are the coordinates of the position of the quad-rotor rotorcraft in the inertial frame, m is the mass of the vehicle, g is the acceleration due to gravity, F_D is the drag force because of air friction, $T_b \in \mathbb{R}^3$ is the thrust vector in the body frame, and $R \in \mathbb{R}^{3\times 3}$ is the rotation matrix. Within the body frame with the inertial frame, transformation matrix R is defined as:

$$R = \begin{bmatrix} \cos\psi\cos\theta & \cos\psi\sin\theta\sin\varphi - \cos\varphi\sin\psi & \sin\sin + \cos\cos\sin \\ \cos\theta\sin\psi & \cos\psi\cos\varphi + \sin\psi\sin\theta\sin\varphi & \cos\sin\sin - \cos\sin \\ -\sin\theta & \cos\sin & \cos\cos \end{bmatrix} \qquad (5.4)$$

The force drag F_D due to air friction is supposed to a force proportional to the linear velocity in each direction and is described as

$$F_D = -k_d \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} \quad (5.5)$$

where k_d is the air friction coefficient. The thrust f_i generated by the i th rotor is given by the following expression

$$f_i = k\omega_i^2 \quad for\ i = 1,2,3,4 \quad (5.6)$$

where k is the propeller lift coefficient and ω_i is the angular velocity of the i th motor. The set of control inputs (speed of motors) are described as

$$F_R = \begin{pmatrix} 0 \\ 0 \\ u \end{pmatrix} \quad (5.7)$$

where the main thrust is modelled as:

$$u = f_1 + f_2 + f_3 + f_4 \quad (5.5)$$

and f_i is described as $f_i = k_i\omega_i^2$ (positive constant and angular speed of the motor)

The total thrust T^b generated by the four rotors is defined as

$$T^b = \sum_{i=1}^{4} f_i = k \begin{bmatrix} 0 \\ 0 \\ \sum_{i=1}^{4} \omega_i^2 \end{bmatrix} \quad (5.8)$$

It is assumed that the dynamics of the motors is determined as the motion of the quad-rotor, and therefore is not taken into account to be different between them.

Typically, the relationship of the angular velocity and applied voltage of motor is regarded as proportional. It is given as

$$\omega_i^2 = c_m v_i^2 \quad for\ i = 1,2,3,4 \qquad (5.9)$$

where c_m is a constant and v is the voltage applied to the motor.

The rotation of quad-rotor can be expressed about the centre of itself instead of about inertial center, because in the inertial frame, it is convenient to obtain linear motion equations and the rotational equations of motion are useful in the body frame (Agudelo and Moor, 2014). Thus, from Euler's equation, the equations for rigid body dynamics are defined as follows:

$$I\dot{\Omega} + \Omega \times (I\Omega) = \tau \qquad (5.10)$$

where $I \in \mathbb{R}^{3\times3}$ is the inertia matrix, $\Omega = \begin{bmatrix}\omega_x & \omega_y & \omega_z\end{bmatrix}^T$ is the angular velocity vector and $\tau = \begin{bmatrix}\tau_1 & \tau_2 & \tau_3\end{bmatrix}^T$ is the vector of the external torques.

The quad-rotor can be modelled as two thin uniform rods crossed at the origin with a point mass (motor) at each end. The inertia matrix is formed in a diagonal matrix of the following form

$$I = \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix} \qquad (5.11)$$

where $I_{xx},\ I_{yy}\ and\ I_{zz}$ are the moments of inertia of the quadcopter about $x^b,\ y^b\ and\ z^b$ axes respectively. The equation (5.12) reduces to

$$\begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix} \begin{bmatrix} \dot{\omega}_x \\ \dot{\omega}_y \\ \dot{\omega}_z \end{bmatrix} = \begin{bmatrix} \tau_\varphi \\ \tau_\theta \\ \tau_\psi \end{bmatrix} - \begin{bmatrix} (I_{yy} - I_{zz})\omega_y\omega_z \\ (I_{zz} - I_{xx})\omega_x\omega_z \\ (I_{xx} - I_{yy})\omega_x\omega_y \end{bmatrix} \qquad (5.13)$$

The torques of roll τ_ϕ and pitch τ_ϕ are derived from standard mechanics as

$$\tau_\varphi = L\left(f_1 - f_3\right) = Lk(\omega_1^2 - \omega_3^2) = Lkc_m(v_1^2 - v_3^2)$$
$$\tau_\theta = L\left(f_2 - f_4\right) = Lk(\omega_2^2 - \omega_4^2) = Lkc_m(v_2^2 - v_4^2) \quad (5.14)$$

where L represents the distance from the rotor to the centre of quad-rotor. The total torque about the z^b axis that is the yaw τ_ψ torque is given below as

$$\tau_\psi = b\left(\omega_1^2 - \omega_2^2 + \omega_3^2 - \omega_4^2\right) = bc_m\left(v_1^2 - v_2^2 + v_3^2 - v_4^2\right) \quad (5.15)$$

where b is the propellers drag coefficient.

The roll, pitch and yaw rates are related to the components of the angular velocity vector by means of the following expression:

$$\begin{bmatrix} \dot{\varphi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & \sin\varphi\tan\theta & \cos\varphi\tan\theta \\ 0 & \cos\varphi & -\sin\varphi \\ 0 & \sin\varphi/\cos\theta & \cos\varphi/\cos\theta \end{bmatrix} \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} \quad (5.16)$$

Finally, the nonlinear equations of motion (quad-rotor rotorcraft) can be expressed in state space form as follows

$$
\begin{aligned}
&\dot{x} = v_x \\
&\dot{y} = v_y \\
&\dot{z} = v_z \\
&\dot{v}_x = \frac{-k_d}{m} v_x + \frac{kc_m}{m}\left(\sin\psi \sin\phi + \cos\psi \cos\phi \sin\theta\right)\left(v_1^2 + v_2^2 + v_3^2 + v_4^2\right) \\
&\dot{v}_y = \frac{-k_d}{m} v_y + \frac{kc_m}{m}\left(\cos\phi \sin\psi \sin\theta + \cos\psi \sin\phi\right)\left(v_1^2 + v_2^2 + v_3^2 + v_4^2\right) \\
&\dot{v}_z = \frac{-k_d}{m} v_z - g + \frac{kc_m}{m}\left(\cos\theta \cos\phi\right)\left(v_1^2 + v_2^2 + v_3^2 + v_4^2\right) \\
&\dot{\phi} = \omega_x + \omega_y\left(\sin\phi \frac{\sin\theta}{\cos\theta}\right) + \omega_z\left(\cos\phi \frac{\sin\theta}{\cos\theta}\right) \\
&\dot{\theta} = \omega_y \cos\theta - \omega_z \sin\theta \\
&\dot{\psi} = \omega_y \frac{\sin\phi}{\cos\theta} + \omega_z \frac{\cos\phi}{\cos\theta} \\
&\dot{\omega}_x = \frac{Lkc_m}{I_{xx}}\left(v_1^2 - v_3^2\right) - \left(\frac{I_{yy} - I_{zz}}{I_{xx}}\right)\omega_y \omega_z \\
&\dot{\omega}_y = \frac{Lkc_m}{I_{yy}}\left(v_2^2 - v_4^2\right) - \left(\frac{I_{zz} - I_{xx}}{I_{yy}}\right)\omega_x \omega_z \\
&\dot{\omega}_z = \frac{bc_m}{I_{zz}}\left(v_1^2 - v_2^2 + v_3^2 - v_4^2\right) - \left(\frac{I_{xx} - I_{yy}}{I_{zz}}\right)\omega_x \omega_y
\end{aligned}
\tag{5.17}
$$

where the state vector of the system, the control input and system output vectors are denoted respectively as $x = \begin{bmatrix} x & y & z & v_x & v_y & v_z & \phi & \theta & \psi & \omega_x & \omega_y & \omega_z \end{bmatrix}^T, x \in \mathbb{R}^{12}$, $u = \begin{bmatrix} v_1^2 & v_2^2 & v_3^2 & v_4^2 \end{bmatrix}^T$ and $y = \begin{bmatrix} x & y & z & \phi & \theta & \psi \end{bmatrix}^T, y \in \mathbb{R}^6$. Notice that the control input vector is in terms of the squared voltages of the rotors, and therefore the control system should compute v_1^2, v_2^2, v_3^2 *and* v_4^2 instead of v_1, v_2, v_3 *and* v_4. The maximum voltage that can be applied to the motors is dependent on the hardware capacity.

5.3 U-State Space Control System Design for Quad-rotor Dynamic Model

A UAV control system contains two main control loops. The first loop is the underlying control loop (inner loop) called ‘vehicle control loop’. This control loop is responsible for the generation and stabilisation of a currently required movement of the UAV (Voos, 2007).

The second loop is the mission control loop (outer loop) that comprises the stabilised vehicle as a platform for mission related sensors and actuators and the mission control system. The mission control loop computes the desired flight path, e.g. given by waypoints, and commands current required movements to the vehicle control loop (Voos, 2007). The remaining question is how to develop controller based on U-state space approach to achieve control targets properly.

Consider a discrete time state space system

$$\begin{aligned} x(k+1) &= f(x(k),u(k)) \\ y(k) &= h(x(k)) \end{aligned} \qquad (5.18)$$

Differentiating the output vector until control input u appears.

$$\frac{y(k+1)-y(k)}{Ts} = h(x(k+1)) - h(x(k)) \qquad (5.19)$$

The continuous time state space system can be discretised by the factor

$$\frac{x(k+1)-x(k)}{Ts} = f(x) \qquad (5.20)$$

The discrete time state space system is

$$x(k+1) = x(k) + Ts * f(x) \qquad (5.21)$$

where T_s is sampling time interval.

Assume a linear dynamical system

$$\begin{aligned} \dot{x} &= Ax + Bu \\ y &= Cx \end{aligned} \qquad (5.22)$$

where $x(t)$ is the state vector and $u(t)$ is control input.

The control law will be state feedback, and a standard form can be expressed as:

$$u = -Kx \qquad (5.23)$$

where K is a matrix of constant feedback coefficients to be determined by the design procedure. The objective of state regulation for the quadcopter is to drive any initial condition error to zero, thus guaranteeing stability. This may be achieved by selecting the control input $u(t)$ to minimise a quadratic cost of the type. The linear quadratic regulator (LQR) is used to design the feedback gain K to obtain the closed-loop matrix A_d. This design method calculates the optimal gain vector K such that the feedback law $u=-Kx$ minimizes the cost function

$$J=\frac{1}{2}\sum_{k=0}^{N}\left[x^T(k)Qx(k)+u^T(k)Ru(k)\right] \qquad (5.24)$$

where Q and R are symmetric positive semidefinite weighing matrices to be selected by the designer. In this way, the optimal control can be calculated by weighting each state and control input through the matrices Q and R respectively. Considering by the control goal, the choice of these matrices can be done by trial and error.

U-state space model design approach was originally proposed to simplify nonlinear control system design on the linear approach available U-platform, it has still advantages over classical approaches while dealing with linear control system design. One of the attractive points is the ability to split accumulated bulk inversion into separate inversions in the design. U state space design is an approach where a feedback linearization loop is applied to the tracking outputs of interest. This approach is designed in discrete time domain directly where initially the convenient output vector $y=[x,y,z,\psi]^T$ is selected for the position control.

Accordingly the task of the design is to determine the desired state variable $x_d(t)$ according to specified performance index A_d.

$$A_d = A-BK \quad (5.25)$$

With reference to U-state space expression (4.8), in simple mathematical expression, it is clear to express U-state space equation as

$$x_{dn}(t)=\sum_{j=0}^{M}\lambda_j(t)u^j(t-1) \quad (5.26)$$

where $\lambda_j(t)$ contains the state variable $x(t)$. Assume that the state variables within $x(t)$ are measurable or obtained by a proper observer, the desired state space equation can be updated from equation (5.26). As mentioned previously, the remaining design task is to resolve one of the roots of the following equation (5.27) to obtain the controller output. That is

$$u(t-1) = \Psi^{-1}\left[x_d(t) - \sum_{j=0}^{M} \lambda_j(t) u^j(t-1) = 0\right] \quad (5.27)$$

where $\Psi^{-1}[*]$ is a root-solving algorithm, such as Newton-Raphson algorithm or other root solver algorithms (Zhu et al., 1999).

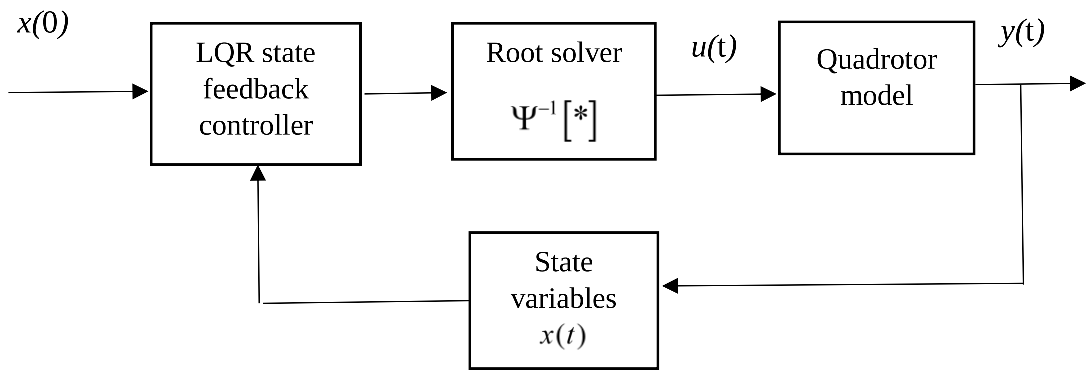

Figure 5.4 block diagram of U-state space design

The U-model approach is applied to the tracking outputs and requires that the residual dynamics (internal dynamics) are stable. Initially, the nonlinear dynamic system is selected to achieve tracking control for position outputs (x, y, z, ψ) or (z, ψ). The output vector $y = (z, \varphi, \theta, \psi)$ is easy to solve the inverse root. This root solver loop yields effectively an inner control loop that linearises the system from control input $u = (u, \tau_\varphi, \tau_\theta, \tau_\psi)$ to system output.

A step-by-step procedure for the U-state space control system design for MIMO aircraft system can be specified as the follows:

Step 1. Select a flight condition to obtain aerodynamic coefficients and corresponding continuous state space equations.

Step 2. Discretise the continuous state space equations into discrete time state space equation by Zero Order Hold equations.

Step 3. Determine the desired closed loop matrix A_d by LQR design

Step 4. Obtain the controller output by root solver (5.27)

5.4 Case Studies

The dynamic model of the quadrotor is derived and implemented in MATLAB/Simulink software. With the help of that computational simulation, the nonlinear vehicle control system is tested and demonstrates the proposed control strategy and its effectiveness and efficiency for the quadrotor dynamic model.

5.4.1 Quadrotor Parameters

In the case studies, a linearised quadrotor model in trim flight condition is selected to demonstrate the proposed U-state space control system design approach. The parameters for a typical quadcopter is shown in Table 5.1.

Table 5.1 Parameters of the quadrotor model (Agudelo and Moor, 2014)

Parameter	Symbol	Value	unit
Mass of the quadcopter	m	0.5	Kg
Radius of the quadcopter	L	0.25	m
Propeller lift coefficient	k	$3 \cdot 10^{-6}$	$N\ s^2$
Propeller drag coefficient	b	$1 \cdot 10^{-7}$	$N\ ms^2$
Acceleration of gravity	g	9.81	m/s^2

Air friction coefficient	k_d	0.25	*kg/s*
Quadcopter inertia about the x^b axis	I_{xx}	$5\cdot10^{-3}$	*kg m*^{2}
Quadcopter inertia about the y^b axis	I_{yy}	$5\cdot10^{-3}$	*kg m*^{2}
Quadcopter inertia about the z^b axis	I_{zz}	$1\cdot10^{-2}$	*kg m*^{2}
Motor constant	c_m	$1\cdot10^{4}$	$V^{-2}s^{-2}$

A reference trajectory is derived that minimises the rate of change of acceleration over the time horizon. The trajectory ensures that the velocities and accelerations at the end point are zero while meeting the position tracking objective. The advantage of this trajectory generation method lies in the fact that more demanding changes in position can be accommodated by varying the final time, that is, acceleration/torque ratio can be controlled smoothly as per requirement and constraints explicitly enforced on the achievable accelerations.

The linear model of the quad-rotor is expressed as

$$\begin{aligned} \dot{x} &= Ax + Bu \\ y &= Cx \end{aligned} \qquad (5.28)$$

where the state vector is $x = \begin{bmatrix} x & y & z & v_x & v_y & v_z & \phi & \theta & \psi & \omega_x & \omega_y & \omega_z \end{bmatrix}^T, x \in \mathbb{R}^{12}$, and control input vector and output are respectively $u = \begin{bmatrix} v_1^2 & v_2^2 & v_3^2 & v_4^2 \end{bmatrix}^T$ and $y = \begin{bmatrix} x & y & z & \phi & \theta & \psi \end{bmatrix}^T, y \in \mathbb{R}^6$. Substituting the parameters from table 5.1, the system matrices are shown as below

$$A = \begin{bmatrix} * & \\ & * \end{bmatrix}_{12*12} \qquad (5.29)$$

$$B = \begin{bmatrix} 0 & \cdots & \cdots & 0 \\ \vdots & & & \vdots \\ 0.6 & 0.6 & 0.6 & 0.6 \\ \vdots & & & \vdots \\ 0.0038 & & -0.0038 & \\ & 0.0038 & & -0.0038 \\ -0.001 & 0.001 & -0.001 & 0.001 \end{bmatrix} \quad (5.30)$$

The continuous time quadcopter model (5.28) is converted into a discrete time expression with a sampling time $T = 0.05s$, given as:

$$x(k+1) = Gx(k) + Hu(k) \quad (5.31)$$

where

$$G = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ & & & & 1 & & & & & & & 0 \\ & & & & & 1 & & & & & & 0 \\ & & & -0.5 & & & & 0.24 & & & & 0 \\ & & & & -0.5 & & 0.24 & & & & & 0 \\ & & & & & -0.5 & & & & & & 0 \\ & & & & & & & & & 1 & & 0 \\ & & & & & & & & & & 1 & 0 \\ & & & & & & & & & & & 1 \\ & & & & & & & & & & & 0 \\ & & & & & & & & & & & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (5.32)$$

$$H = \begin{bmatrix} 0 & \cdots & \cdots & 0 \\ \vdots & & & \vdots \\ 0.03 & 0.03 & 0.03 & 0.03 \\ \vdots & & & \vdots \\ 0.0019 & & -0.0019 & \\ & 0.0019 & & -0.0019 \\ -0.0005 & 0.0005 & -0.0005 & 0.0005 \end{bmatrix} \quad (5.33)$$

5.4.2 Linear model of quadrotor and control

Using LQR design, matrices Q and R are setup and the feedback gain matrix can be determined as $K \in \mathbb{R}^{4\times 12}$. Thus, the desired closed loop state space model can be expressed as:

$$x_d(k+1) = A_d x(k) + B_d w(k) \qquad (5.34)$$

where

$$A_{cl} = \begin{bmatrix} 1 & 0 & 0 & 0.1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ & 1 & & & 0.1 & & & & & & & 0 \\ & & 1 & & & 0.1 & & & & & & 0 \\ & & & 0.95 & & & & 0.024 & & & & 0 \\ & & & & 0.95 & & 0.024 & & & & & 0 \\ & & & & & 0.95 & & & & & & 0 \\ & & & & & & 1 & & & 0.1 & & 0 \\ & & & & & & & 1 & & & 0.1 & 0 \\ & & & & & & & & 1 & & & 0.1 \\ & & & & & & & & & 1 & & 0 \\ & & & & & & & & & & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad B_d = 0$$

It can be found that:

$$x_d(k+1) = A_d x(k) = \lambda_0(k) + \lambda_1(k)u(t) \qquad (5.35)$$

where the initial state variables are assumed to be:

$$x(k) = \begin{bmatrix} 0 & 0 & -1 & 0 & 0 & 0 & 45*(\frac{\pi}{180}) & 45*(\frac{\pi}{180}) & 45*(\frac{\pi}{180}) & 0 & 0 & 0 \end{bmatrix}^T$$

From the controller output is determined by:

$$u(t) = \frac{x_d(t+1) - \lambda_0(t)}{\lambda_1(t)} \qquad (5.36)$$

For U-state space method, the first design task is to determine the desired state variables which are impacted by the control inputs. Consider a simplified discrete-time quadrotor model from (5.35)

$$\Delta x(k+1) = G\Delta x(k) + H\Delta u_c(k) \qquad (5.37)$$

where the state vector and control inputs are respectively $\Delta x = \begin{bmatrix} \Delta z & \Delta\omega_x & \Delta\omega_y & \Delta\omega_z \end{bmatrix}^T$ and $\Delta u_c = \begin{bmatrix} \Delta u_1 & \Delta u_2 & \Delta u_3 & \Delta u_4 \end{bmatrix}^T$. Substitute the related parameters from Table 5.1, thus the state space equation can be written as:

$$\begin{bmatrix} \Delta v_z(k+1) \\ \Delta\omega_x(k+1) \\ \Delta\omega_y(k+1) \\ \Delta\omega_z(k+1) \end{bmatrix} = \begin{bmatrix} 0.75 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \Delta z(k+1) \\ \Delta\omega_x(k+1) \\ \Delta\omega_y(k+1) \\ \Delta\omega_z(k+1) \end{bmatrix} + \begin{bmatrix} 0.03 & 0.03 & 0.03 & 0.03 \\ 1.9\times10^{-3} & 0 & -1.9\times10^{-3} & 0 \\ 0 & 1.9\times10^{-3} & 0 & -1.9\times10^{-3} \\ -5\times10^{-4} & 5\times10^{-4} & -5\times10^{-4} & 5\times10^{-4} \end{bmatrix} \begin{bmatrix} \Delta u_1(k) \\ \Delta u_2(k) \\ \Delta u_3(k) \\ \Delta u_4(k) \end{bmatrix} \qquad (5.38)$$

By using LQR design, the matrices Q and R are setup and the desired closed loop state space equation is determined as

$$A_d = G - HK \qquad (5.39)$$

It can be found that the desired closed loop state equations

$$x_d(k+1) = A_d x(k) = \lambda_0(k) + \lambda_1(k)u(k) \qquad (5.40)$$

where

$$\lambda_0(\mathrm{t}) = \begin{bmatrix} 0.75 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \text{ and } \lambda_1(\mathrm{t}) \begin{bmatrix} 0.03 & 0.03 & 0.03 & 0.03 \\ 1.9\times10^{-3} & 0 & -1.9\times10^{-3} & 0 \\ 0 & 1.9\times10^{-3} & 0 & -1.9\times10^{-3} \\ -5\times10^{-4} & 5\times10^{-4} & -5\times10^{-4} & 5\times10^{-4} \end{bmatrix}.$$

From (5.4), the control input can be obtained by

$$u(k)=\frac{x_d(k+1)-\lambda_0(k)}{\lambda_1(k)} \qquad (5.41)$$

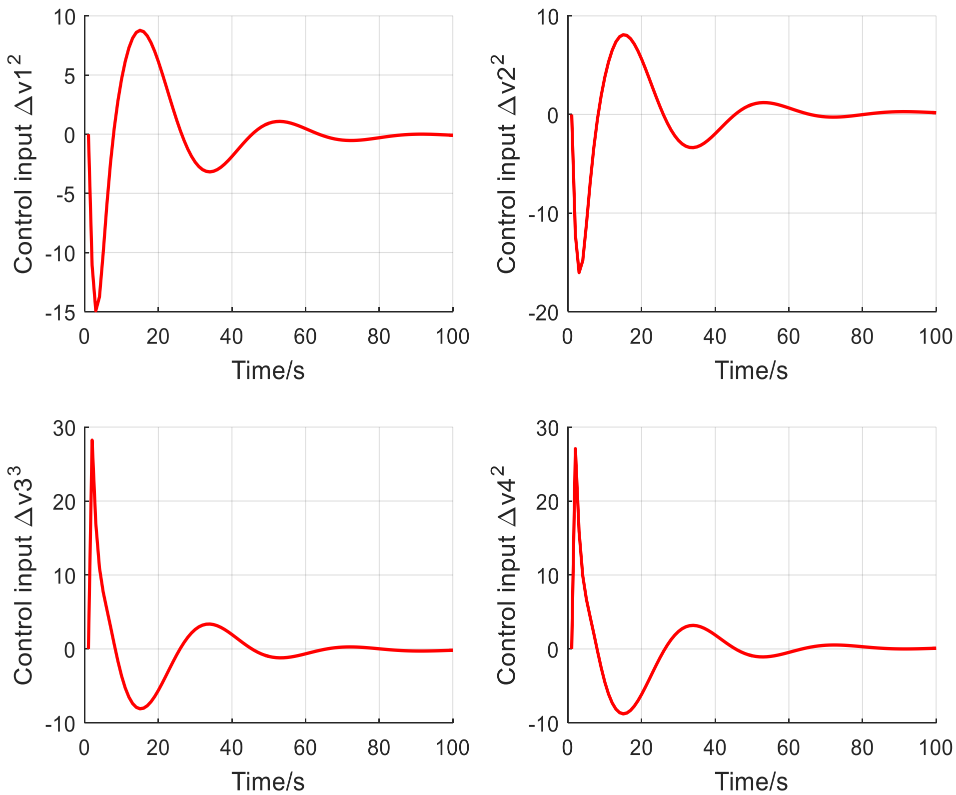

Figure 5.5 Control inputs Δu

The initial states are given as $\Delta x=\begin{bmatrix}-1 & 0.7854 & 0.7854 & 0.7854\end{bmatrix}^T$. Note that the control input vector $\Delta u=\begin{bmatrix}\Delta v_1^2 & \Delta v_2^2 & \Delta v_3^2 & \Delta v_4^2\end{bmatrix}^T$ is in terms of the square voltages of the rotors, where every control input should be positive and constraint in the range of $0\ V^2 \le u \le 100\ V^2$.The simulation results are shown in figure 5.5.

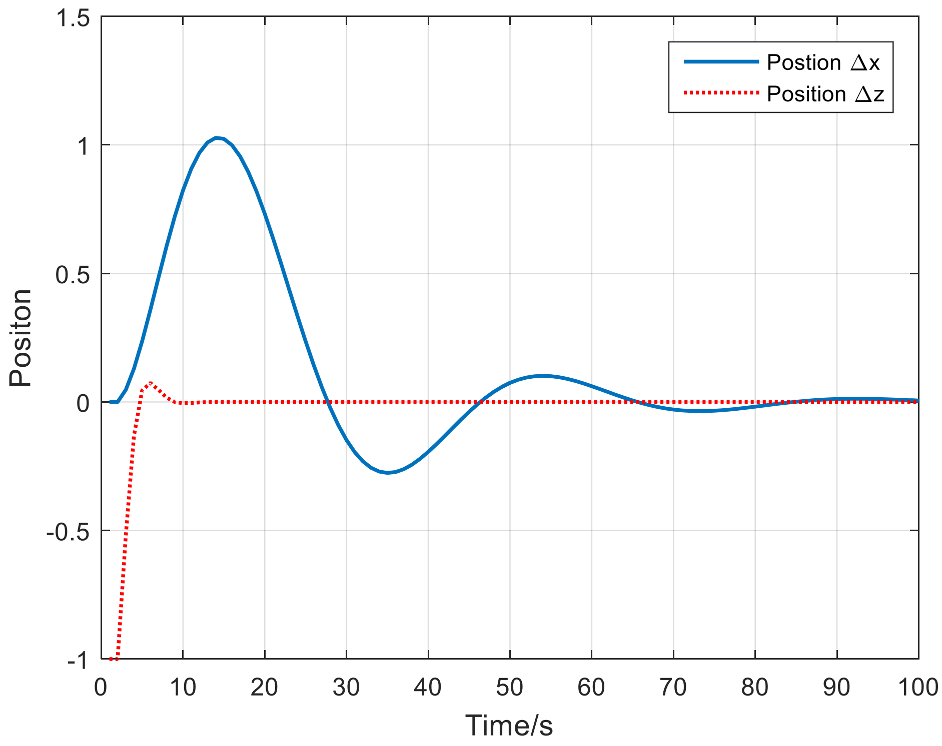

Figure 5.6 Resultant in position

Although U-state space model design approach was originally proposed to simplify single-input single-output (SISO) nonlinear control system design, by using linear method. Under the U-state space platform, it has still advantages over classical approaches while dealing with the linear design approaches for SISO and MIMO systems. This study from theory to simulation confirms again this superiority. In the next section, the original nonlinear dynamic model will be studied to apply the U-state space design approach.

5.4.3 Nonlinear quadrotor control

In this section, the same LQR controller (in section 5.4.1) is used to control the nonlinear dynamic model (5.16).

The nonlinear model is rewritten as:

$$\begin{bmatrix}\dot{v}_x\\ \dot{v}_y\\ \dot{v}_z\end{bmatrix}=\begin{bmatrix}\frac{-k_d}{m} & & \\ & \frac{-k_d}{m} & \\ & & \frac{-k_d}{m}v_z\end{bmatrix}\begin{bmatrix}v_x\\ v_y\\ v_z\end{bmatrix}+\begin{bmatrix}0\\ 0\\ -g\end{bmatrix}+\begin{bmatrix}\frac{kc_m}{m}\left(\sin\psi\sin\phi+\cos\psi\cos\phi\sin\theta\right)\\ \frac{kc_m}{m}\left(\cos\phi\sin\psi\sin\theta+\cos\psi\sin\phi\right)\\ \frac{kc_m}{m}\left(\cos\theta\cos\phi\right)\end{bmatrix}u_1 \quad (5.42)$$

$$\begin{aligned}\dot{\omega}_x&=\frac{Lkc_m}{I_{xx}}\left(v_1^2-v_3^2\right)-\left(\frac{I_{yy}-I_{zz}}{I_{xx}}\right)\omega_y\omega_z\\ \dot{\omega}_y&=\frac{Lkc_m}{I_{yy}}\left(v_2^2-v_4^2\right)-\left(\frac{I_{zz}-I_{xx}}{I_{yy}}\right)\omega_x\omega_z\\ \dot{\omega}_z&=\frac{bc_m}{I_{zz}}\left(v_1^2-v_2^2+v_3^2-v_4^2\right)-\left(\frac{I_{xx}-I_{yy}}{I_{zz}}\right)\omega_x\omega_y\end{aligned} \quad (5.43)$$

The discretise operator is used for conversion nonlinear plant from continuous time into discrete time. The desired closed loop is the same as (5.34)

$$x_d(k+1)=A_d x(k)+B_d w(k) \quad (5.44)$$

where the initial state variables are:

$$x(k)=\begin{bmatrix}0 & 0 & -1 & 0 & 0 & 0 & 45*(\frac{\pi}{180}) & 45*(\frac{\pi}{180}) & 45*(\frac{\pi}{180}) & 0 & 0 & 0\end{bmatrix}^T$$

The U-state space expression of nonlinear plant (5.43) and (5.44) as:

$$x(k+1)=\lambda_i(k)+\lambda_{i+1}(k)u(k) \quad i=0,1,2,3... \quad (5.45)$$

where

$$\lambda_0(t)=\begin{bmatrix}\frac{-k_d}{m} & & \\ & \frac{-k_d}{m} & \\ & & \frac{-k_d}{m}v_z\end{bmatrix}\begin{bmatrix}v_x(t)\\ v_y(t)\\ v_z(t)\end{bmatrix}+\begin{bmatrix}0\\ 0\\ -g\end{bmatrix}$$

$$\lambda_1(t) = \begin{bmatrix} \frac{kc_m}{m}\left(\sin\psi(t-1)\sin\phi(t-1) + \cos\psi(t-1)\cos\phi(t-1)\sin\theta(t-1)\right) \\ \frac{kc_m}{m}\left(\cos\phi(t-1)\sin\psi(t-1)\sin\theta(t-1) + \cos\psi(t-1)\sin\phi(t-1)\right) \\ \frac{kc_m}{m}\left(\cos\theta(t-1)\cos\phi(t-1)\right) \end{bmatrix}$$

$$\lambda_2(t) = -\left(\frac{I_{yy} - I_{zz}}{I_{xx}}\right)\omega_y(t-1)\omega_z(t-1)$$

$$\lambda_3(t) = \frac{Lkc_m}{I_{xx}}$$

$$\lambda_4(t) = -\left(\frac{I_{zz} - I_{xx}}{I_{yy}}\right)\omega_x(t-1)\omega_z(t-1)$$

$$\lambda_5(t) = \frac{Lkc_m}{I_{yy}}$$

$$\lambda_6(t) = -\left(\frac{I_{xx} - I_{yy}}{I_{zz}}\right)\omega_x(t-1)\omega_y(t-1)$$

$$\lambda_7(t) = \frac{bc_m}{I_{zz}}$$

The inner loop controller output u is determined by:

$$u(t) = \begin{bmatrix} u_1(t) \\ u_2(t) \\ u_3(t) \\ u_4(t) \end{bmatrix} = \begin{bmatrix} \frac{x_{d3}(t) - \lambda_0(t)}{\lambda_1(t)} \\ \frac{x_{d4}(t) - \lambda_2(t)}{\lambda_3(t)} \\ \frac{x_{d5}(t) - \lambda_4(t)}{\lambda_5(t)} \\ \frac{x_{d6}(t) - \lambda_6(t)}{\lambda_7(t)} \end{bmatrix} \qquad (5.46)$$

The block diagram of U-state space nonlinear design is shown on figure 5.8. The outer loop is similar design process as the inner loop.

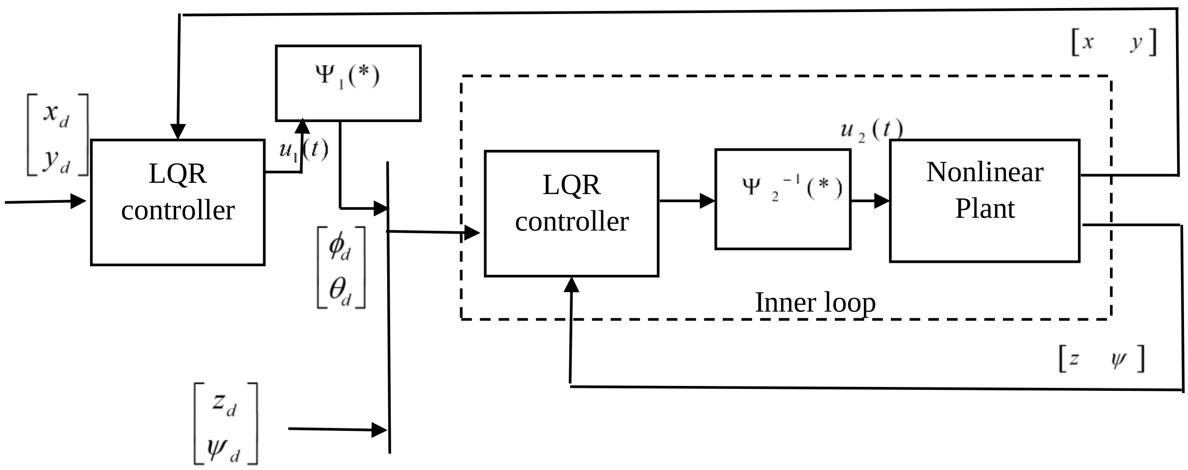

Figure 5.7 Block diagram of U-state space nonlinear control

The simulation results are shown on figures 5.8 and 5.9.

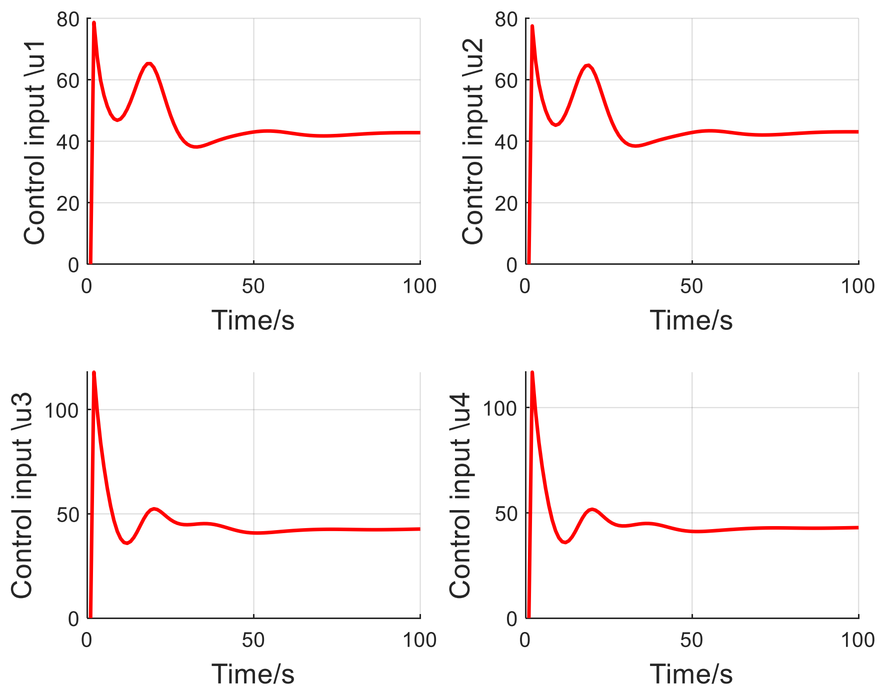

Figure 5.8 Control inputs *u*

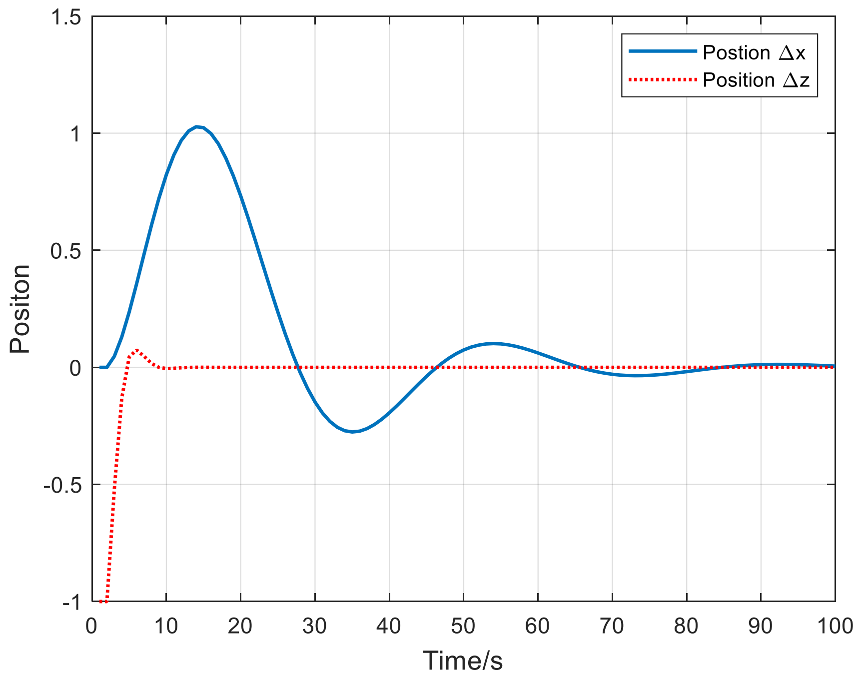

Figure 5.9 Resultant in position

From simulation results, it can be inspected that the nonlinear system outputs achieve the same performance as those in the linear case. The nonlinear control inputs are different from the control inputs in linear case. The reason is that the nonlinear model has been linearised around equilibrium point which results state variables and control inputs are $\Delta x,\ \Delta u$. However, the nonlinear simulation shows the results of original state variables and control inputs $x,\ u$.

5.5 Conclusions

In this chapter, a nonlinear dynamic model of four rotors quadrotor is introduced to be controlled subject for proposed U-state space control system design. U-state space model is extended from SISO system to the square MIMO state space system. The U-state space design approach for MIMO system is similar to that for SISO design. The closed loop characteristic equation is determined by the linear control approach. Then the control inputs can be obtained by solving the related function. In simulation studies, the control performance is given as computational experiment results via MATLAB/Simulink. Note that the conducted control input should follow the range of the hardware values.

In the next section, an inverted pendulum system will be introduced. As a Single Input Multi Output (SIMO) system, U-state space control will be applied to control and stabilise the unstable system.

Chapter 6

U-State Space Enhanced Control of Inverted Pendulum

6.1 Introduction

6.1.1 Overview of inverted pendulum control systems

The Inverted Pendulum is recognised as a classical example in the control research domain, as it contains a range of dynamic characteristics such as high order dynamic, nonlinearity, inherently unstable, multivariate and tight coupling.

The stabilisation and control of the inverted pendulum is the world recognised challenges on control theory research and application development. This is typically due to its low cost, simple structure, easy simulation and implementation of control in the two different ways (analogue and digital). Moreover, it is a quite complex controlled object with many different characteristics, and it only is well stabilised using an effective control strategy. Although an inverted pendulum has a simple structure (a cart and pendulum), the difficulty of stabilisation and control such systems is dramatically increasing due to the numbers of connected pendulums. Inverted pendulum system is not only an ideal experiment device for verifying the performance of control strategy, but also the dynamic of this system behaves as similar as many applications.

The industrial applications based on the research of inverted pendulum control systems include:

- Robot movement (such as standing and walking) like a double inverted pendulum system. The first humanoid robot, known as Elektro, was exhibited at the New York World's Fair eighty years ago (Schaut, 2006). It is still the objective of many researchers to develop key control strategies to achieve smoothing movement for humanoid robot.
- The real-time control for rockets to maintain the desired attitude during flight processing. For example, the multi-stage rocket is developed to prevent the crash of single-stage rocket during launch. The flight attitude control for such rocket is often studied by multi-stage inverted pendulum system.

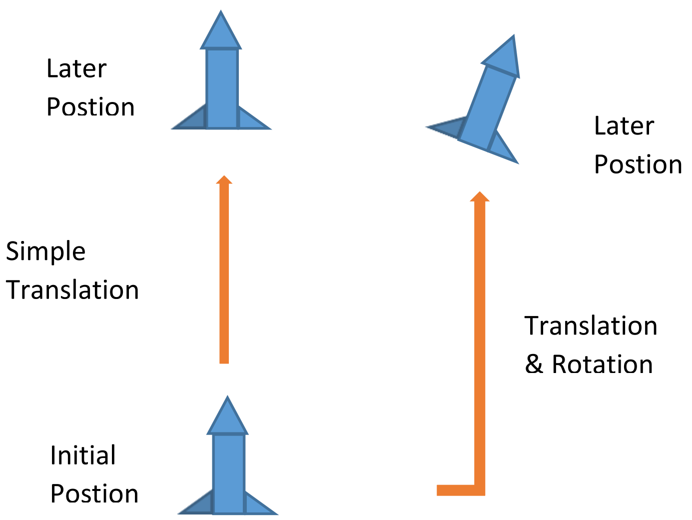

Figure 6.1 Basic rocket motion (NASA, 2014)

- The communications satellite maintains its stable posture while tracking of the pre-calculated orbit at a fixed location in order to keep the satellite antenna pointing to the Earth and its solar panels always pointing towards the sun. For reconnaissance satellite, slight jitter will have a great impact on the image quality of the camera. To guarantee the quality of the camera, it must eliminate vibration by automatically maintaining the stability of the servo attitude. The research of inverted pendulum provides useful theoretical references and laboratory results underpinning this application.

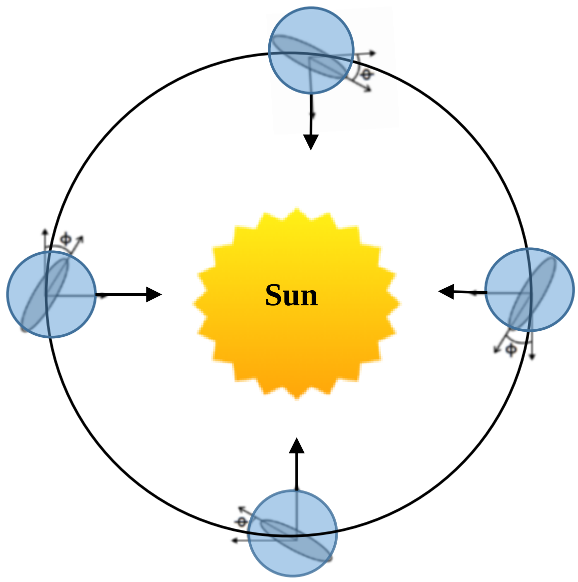

Figure 6.2 Sun synchronous orbit (Alvenes, 2012)

- The control of a tower crane is a classic problem that requires the balancing of an inverted pendulum by moving a cart along a horizontal track (Mladenov, 2011). When moving the shipping containers back and forth, the cranes move the box accordingly so that it never swings or sways. It will stays perfectly positioned under the operator even when moving or stopping quickly.

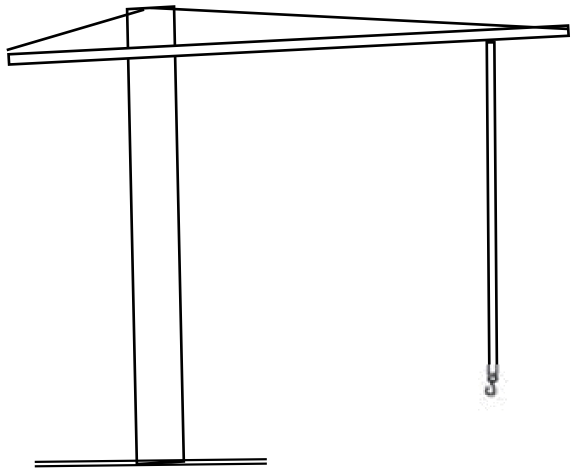

Figure 6.3 Tower Crane

- The Segway Transporter, known as a unicycle, is a popular vehicle nowadays which is a vehicle extended from the inverted pendulum system and balancing robot (Kim and Park, 2016). This system is an uncertain nonlinear system and has an unknown time-varying control coefficient. In a Segway Transporter, the pivot of the pendulum is the axle of a wheel or pair of wheels where the wheel is powered by an electric motor. The movement of Segway is stabilised by the designed controller to dynamically balance the pendulum.

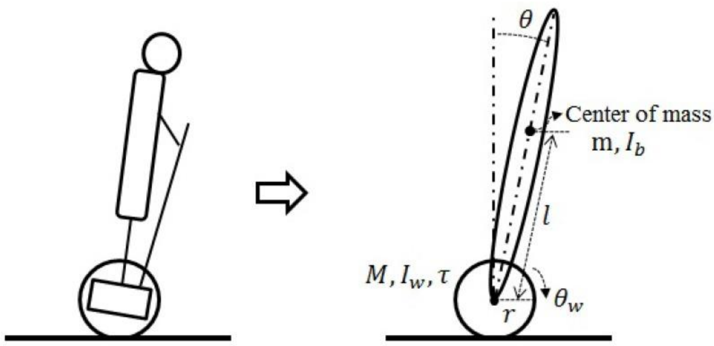

Figure 6.4 Segway model (Kim and Park 2016)

From the early 1990s, the inverted pendulum has been became a hot topic in the control domain and has led to the development of many new control strategies and approaches for more complex systems. The reasons for selecting the inverted pendulum as the system are:

- It is the one of the most easily available systems for laboratory usage.
- It is originally a nonlinear system. However, it can be studied and tested as linear system without too much error in a wide range of variation.
- It can provide a significant reference and practice experiment for prospective control engineers.

6.1.2 Inverted pendulum control strategies

The performance of the designed control systems for inverted pendulum can be either directly observed by its stability or can be measured by the performance indexes (also state variables) such as angle of pendulum, position of the cart and settling time. The experimental results are intuitive and significant to verify the accuracy and practicality of applied controller and to compare the performance of various different control methods.

Moreover, many control problems (such as stabilisation problems, nonlinear problems, robustness such as follow-up problems and tracking problems) can be studied and analysed with the design of inverted pendulum control systems. The inverted pendulum system is a popular demonstration of using feedback control to stabilise an open loop unstable system. The first solution to this problem was described by Roberge (1960) in his aptly named thesis, “The Mechanical Seal”. Subsequently, it has been used in many books and papers as an example of an unstable system (Ogata, 2009). Based on the pole placement methods of classical and modern control theory, many researchers have successfully designed the effective analogue controllers for the stabilisation of single and double inverted pendulum systems (Lan and Fei, 2011; Rajak, 2015).

Proportional Integral Derivative (PID) control is one of the simplest implementations for designed controller and sufficient capacity to solve lots of industrial control problems. Many researchers have proposed the PID controller to stabilise the inverted pendulum. More than one PID controllers were designed for stabilisation and tracking control for three types of inverted pendulum (Wang, 2011). The first PID controller was to control the angle of the pendulum and the other one was to track the position of the cart (shown in Figure 6.5). This control system is not only solving stabilisation and tracking problems for inverted pendulum, but also having robustness against large and fast disturbances.

Based on this PID control system, another PID controller is designed to deal with the combination of horizontal and vertical control forces in x-z axis of the inverted pendulum (Wang, 2015). In case studies, simulation results are compared with the classical inverted pendulum (one horizontal control force), which are shown to enlarge the stability domain and robustness margin by the vertical control force. This proposed PID control system not only realises the stabilisation and tracking control of the inverted pendulum in the horizontal and vertical space, but also has better performance and flexibility than the standard inverted pendulum control system.

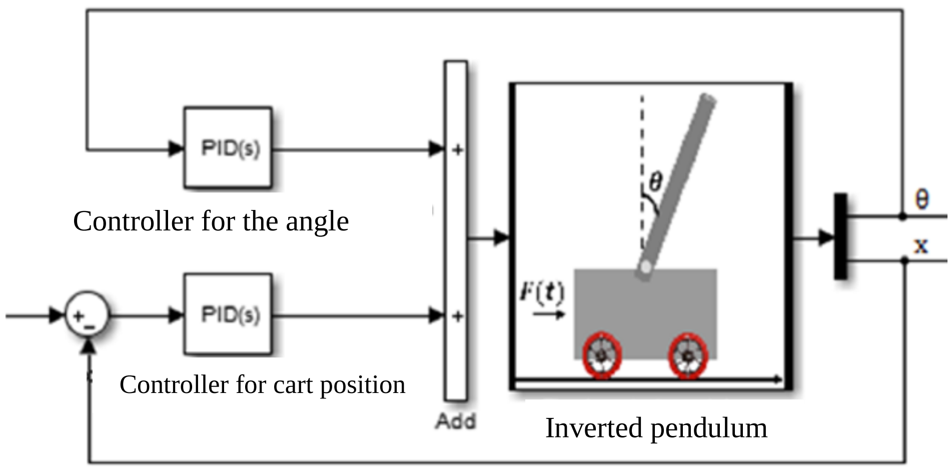

Figure 6.5 PID controllers for the inverted pendulum (Krafes et al., 2016)

The LQR control is one of the optimal control techniques that looks for a feedback gain for state feedback control. Based on state feedback control, the angle and position is stabilised and minimized by the quadratic criterion of inverted pendulum (Krafes et al., 2016). The weighting matrices Q and R is significant to determine the positions and velocities in order to stabilise of the system. Figure 6.6 shows the schematic of a standard LQR control system for the inverted pendulum. To stabilise the system, LQR controller is designed to compare with different weighting matrices Q and R (Wang et al., 2010). It shows that the larger feedback gain (determined by weighting coefficients) results a quick response time and reduced overshoot.

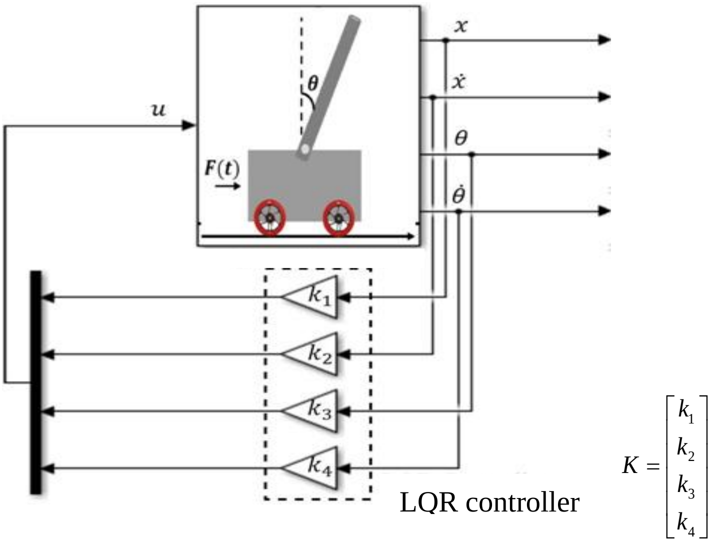

Figure 6.6 Standard LQR control system for the inverted pendulum

A combined PID controller with LQR optimal control system has been implemented to control the nonlinear inverted pendulum system with disturbance input (Prasad et al., 2011). The case studies compared the system performance of three different control system with disturbance input including purely PID control for pendulum angle and cart position, LQR stabilisation combined with PID controllers (angle and position) and LQR stabilisation with PID control cart position. Although the feedback gain (LQR) is obtained by using linearised models, the simulation results show that a combination control system response for nonlinear inverted pendulum is effective and robust (better than PID control).

Based on linearisation, nonlinear effects are neglected so that linear controllers are designed to stabilise the inverted pendulum. Some linear controllers can also work on nonlinear model with satisfied performance. However, the dynamics provided by these nonlinear effects is richer than linear systems (Krafes et al., 2016). It should be necessary to have better robustness and accuracy of controllers. Thus, numerous nonlinear control approaches have been proposed for the inverted pendulum to solve the stabilisation problem.

The Sliding Mode Control (SMC) is a nonlinear control technique with numerous advantages including quick response, insensitivity to parameter variation and disturbance, and easy tuning and implementation. A discontinuous state feedback control signal was designed to force the system dynamics to move toward an adjacent region (state trajectories). A sliding mode controller design was proposed for a rotational inverted pendulum (Grossimon et al., 1996). In this case, the tip of the pendulum arm could be placed at any reachable point so that the position of this point will be a function of $f(\beta,\theta)=\theta_{des}(\beta)$. This gives a unique surface design by SMC. Figure 6.7 shows a block diagram of an SMC system.

To resolve chattering phenomenon problem of SMC, there is continued research to develop algorithms of the second order sliding mode control. Twisting and a super-twisting algorithms (continuous sliding mode) were presented to ensure main properties of the first order sliding mode control for systems with Lipschitz continuous matched uncertainties or disturbances with bounded gradients (Mahjoub et al., 2013; Krafes et al., 2016). Numerical simulation results show that higher order SMC performs better compared to the first order sliding mode controller.

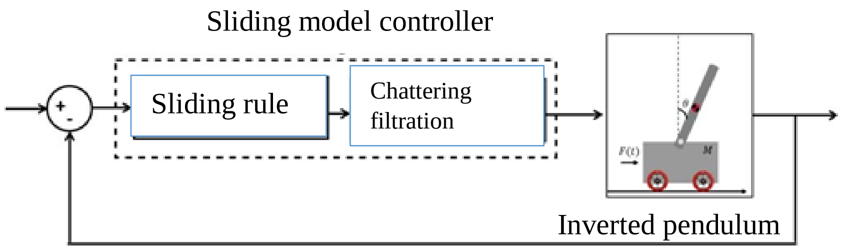

Figure 6.7 Sliding mode controller for the inverted pendulum (Krafes et al., 2016)

Backstepping is another popular nonlinear control method applied to the development of inverted pendulum control systems. To guarantee overall stability, backstepping is a systematic method for nonlinear control design where it decomposes the system into several subsystems where each subsystem is stabilised by Lyapunov stability criterions.

A control system that combines the feedback linearisation and backstepping has been also proposed to control the non-minimum phase nonlinear systems (Yakoub et al., 2013) where the system is decomposed into two strict feedback subsystems. Then, the control strategy is synthesised by the sum of two types of control laws that represented the controller obtained by the backstepping and the controller obtained by the input-output feedback linearisation respectively. For an inverted system, the backstepping algorithm is synthesised to regulate the rod angle without regard to the cart motion; and then synthesised to regulate the cart position. Figure 6.8 shows a block diagram of a backstepping and feedback linearisation controller for the inverted pendulum.

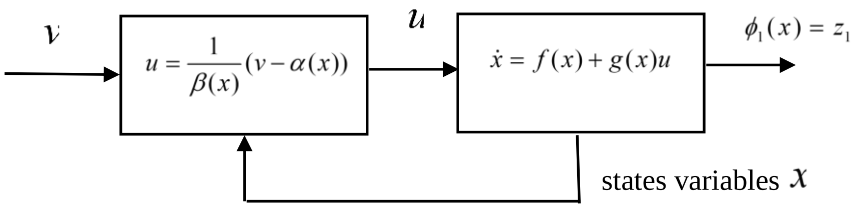

Figure 6.8 Block diagram of feedback linearisation (Slotine and Li, 1991)

In a comparative study of trajectory tracking control, three different controllers respectively SMC, backstepping SMC and feedback linearisation SMC were designed to test the system performance (Sassi and Abdelkrim, 2015). The simulation results shows that in the case of third controller the presence of the chattering phenomenon is noticed. A slower convergence to the desired trajectory was noticed with the use of the first controller. Using the second controller, the system response was faster and the system reached the desired trajectory in a shorter period of time (Krafes et al., 2016).

An exact linearization is proposed based on differential geometry techniques in order to algebraically obtain the equivalent linear equations from the original nonlinear system.so that linear control techniques (such as feedback control) can be applied for further controller design (Zhang and Wang, 2011). This combination gives a good response and good robustness for both the pendulum and the cart position.

From the above researches, the main idea is to obtain an approximated linear model and then to design with linear control approaches. U-model methodology, which is a generic systematic approach to convert the nonlinear model into a controller output based time-varying expression model (Quan et al., 2016), aims to transform the nonlinear model into an equivalent expression without any approximation so that those well-known linear approaches developed can be directly applied to such nonlinear U-model expression.

In this chapter, the nonlinear inverted pendulum model is introduced in section 6.2. The U-state space control system design for SIMO is proposed in section 6.3. In the next section, the case studies for the U-state space control system is demonstrated using a computational experiment. Finally, the conclusions of this chapter is presented.

6.2 Inverted Pendulum System

A standard inverted pendulum system is a motorized cart connected with a pendulum also known as stick balancer; a schematic diagram of the inverted pendulum is shown in Figure 6.9. In this example it will be considered as a two-dimensional problem where the pendulum is constrained to move in the vertical direction. For this system, the control input is the force F that can move the cart horizontally and the outputs are the angular position of the pendulum θ and the horizontal position of the cart *x*.

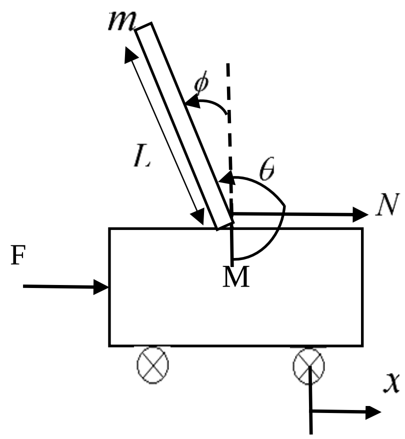

Figure 6.9 Schematic diagram of inverted pendulum

Based on Newton's law, summing the forces in the free body diagram (shown in Figure 6.10) of the cart in the horizontal direction, the motion of the cart can be expressed as:

$$M\ddot{x} = F - b\dot{x} - N \qquad (6.1)$$

where abbreviation of inverted pendulum is listed in Table 6.1.

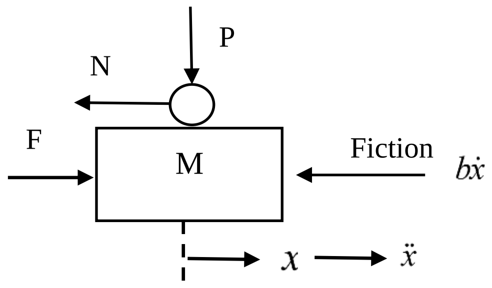

Figure 6.10 Force analysis of the cart

Similarly from the horizontal force analysis (shown in Figure 6.11) of the pendulum force ,it can be derived the following equation:

$$N = m\frac{d^2}{dt^2}(x - l\sin\Phi) \qquad (6.2)$$

where Φ represent the deviation of the pendulum's position from equilibrium, that is, $\theta = \pi - \Phi$. That is:

$$N = m\ddot{x} + ml\ddot{\theta}\cos\theta - ml\dot{\theta}^2\sin\theta \qquad (6.3)$$

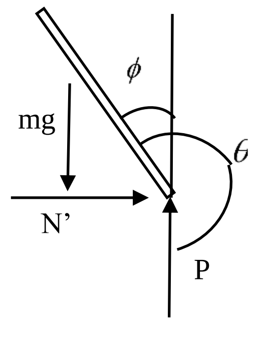

Figure 6.11 Force anaylsis of pendulum

By substituting equation (6.3) into equation (6.1), the reaction force *N* can be expressed as:

$$(M+m)\ddot{x} + b\dot{x} + ml\ddot{\theta}\cos\theta - ml\dot{\theta}^2\sin\theta = F \qquad (6.4)$$

Table 6.1 Abbreviation of inverted pendulum

x	The position of the cart on the horizontal axis
F	Force applied the cart
m	Mass of the pendulum
M	Mass of the cart
L	Length of the pendulum
l	Length to pendulum centre of mass
φ	Angle between pole and vertical upward position
θ	Angle between pole and vertical downward position
N	Interactive force's components for cart and pole on horizontal position
P	Interactive force's components for cart and pole on vertical position
b	Coefficient of friction for cart
g	Gravitational force

Similarly, from force analysis of the pendulum (shown in Figure 6.11), it gives:

$$P\sin\theta + N\cos\theta - mg\sin\theta = ml\ddot{\theta} + m\ddot{x}\cos\theta \qquad (6.5)$$

Summating of the moments about the centroid of the pendulum achieves following equation:

$$-Pl\sin\theta - Nl\cos\theta = I\ddot{\theta} \qquad (6.6)$$

Combining equation (6.5) and (6.6), it gives:

$$(I + ml^2)\ddot{\theta} + mgl\sin\theta = -ml\ddot{x}\cos\theta \qquad (6.7)$$

The nonlinear model of inverted pendulum can then be expressed as:

$$\begin{cases} (I + ml^2)\ddot{\theta} + mgl\sin\theta = -ml\ddot{x}\cos\theta \\ (M+m)\ddot{x} + b\dot{x} + ml\ddot{\theta}\cos\theta - ml\dot{\theta}^2\sin\theta = u \end{cases} \qquad (6.8)$$

The following small angle approximations can be applied to simplify the non-linear functions in our system equations:

$$\begin{aligned} \cos\theta &= \cos(\pi+\Phi) = -1 \\ \sin\theta &= \sin(\pi+\Phi) = -\Phi \\ \dot{\theta}^2 &= \dot{\Phi}^2 = 0 \end{aligned} \qquad (6.9)$$

It can be concluded the following linearised equations of motion as:

$$\begin{aligned} (I+ml^2)\ddot{\Phi} - mgl\Phi &= ml\ddot{x} \\ (M+m)\ddot{x} + b\dot{x} - ml\ddot{\Phi} &= u \end{aligned} \qquad (6.10)$$

The state vector is $x = \begin{bmatrix} x & \dot{x} & \Phi & \dot{\Phi} \end{bmatrix}$, and the linearised equations of motion from (6.10) can also be represented in state-space form as:

$$\begin{aligned} \dot{x} &= \dot{x} \\ \ddot{x} &= \frac{-(I+ml^2)b}{I(M+m)+Mml^2}\dot{x} + \frac{m^2gl^2}{I(M+m)+Mml^2}\Phi + \frac{(I+ml^2)}{I(M+m)+Mml^2}u \\ \dot{\varphi} &= \dot{\varphi} \\ \ddot{\varphi} &= \frac{-mlb}{I(M+m)+Mml^2}\dot{x} + \frac{mgl(M+m)}{I(M+m)+Mml^2}\Phi + \frac{ml}{I(M+m)+Mml^2}u \end{aligned} \qquad (6.11)$$

It is assumed here that the pendulum rod is mass-less, and the hinge is frictionless. Thus, equation (6.4) can be expressed as:

$$(M+m)\ddot{x} + ml\ddot{\Phi}\cos\Phi - ml\dot{\Phi}^2\sin\Phi = u \qquad (6.12)$$

6.3 U-State Space Control System Design for Linear System

6.3.1 Linear controller design

In order to use linear state space model-based design approaches, the desired state vector is defined as $x_d(t)$, where the specified feedback gain is defined by designers in advance (such as pole placement or LQR). Therefore, the relationship between a specified state vector $x_d(t)$ and the requested corresponding controller output $u(t-1)$ can be expressed in terms of the U-state space model as:

$$x_d(t) = \sum_{j=0}^{M} \lambda_j(t) u^j(t-1) \qquad (6.13)$$

Based on U-state space design, the proposed design procedure can be classified into two steps. The first task of the U-state space design is to determine the desired state vector as $x_d(t)$ according to a specified performance index.

A linear feedback controller is:

$$\begin{aligned} x(t+1) &= Ax(t) + Bu(t) \\ u(t) &= -Kx(t) \end{aligned} \qquad (6.14)$$

where A, B are state space matrices and $u(t)$ is feedback controller; K is a matrix of constant feedback coefficients. The LQR control minimises the cost function as:

$$J = \frac{1}{2} \sum_{k=0}^{N} \left[x^T(k) Q x(k) + u^T(k) R u(k) \right] \qquad (6.15)$$

The desire state vector can be derived as:

$$x_d(t) = A_c x(t) + B_c r(t) \qquad (6.16)$$

Assume that the state variables in the inverted pendulum system are measurable or obtained by a proper observer. Then, the following task is to obtain the desired controller output $u(t-1)$. By resolving one of the roots of equation (6.13), it can be expressed in terms of:

$$u(t-1) = \Psi^{-1} \left[x_d(t) - \sum_{j=0}^{M} \lambda_j(t) u^j(t-1) = 0 \right] \qquad (6.17)$$

where Ψ^{-1} is root solving algorithm.

Figure 6.14 shows a general U-state space feedback control system structure with the proposed design procedure.

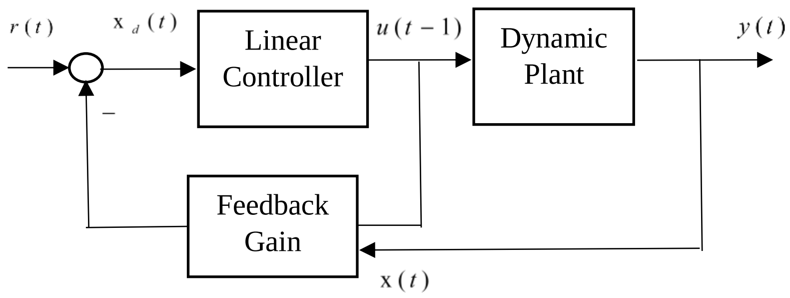

Figure 6.12 Block diagram of U-state space feedback control

The U-state space model (6.13) is regarded as a type of time-varying polynomials, where the values of these time-varying parameters are necessary to know in order to derive the desired equation. Moreover, the related controller is actually an online algorithm because its control input must be updated from the time-varying parameter vector in each sampling interval.

For the classical design procedure, the performance index of the classical control is needed to determine the criterion function. Then, the controller output is obtained by resolving the criterion function. For a linear system case, it requests for the plant model to obtain the solution from inversion of the equations based on such procedure. For the case of the nonlinear systems, there are more difficulties on the inversion calculation (Quan et al, 2016).

For a state feedback, the controller is:

$$u = -Kx \qquad (6.18)$$

where u is control input, K is feedback gain and x is state vector.

The state space equations for the closed-loop feedback system are expressed as:

$$\dot{x} = Ax + B(-Kx) = (A - BK)x \qquad (6.19)$$

From equation (6.19), it can be found that criterion function in this example relies on the system matrices A and B, because the desired poles determine the closed loop matrix $A - BK$.

Compared with U-state space procedure, the first task is to resolve the criterion function to obtain the designed/desired state vector. Then, the controller output can be obtained by resolving desired state vector through U-state space expression (equation (6.13)). For some linear approaches (such as pole placement), it does not require the plant of model to derive the desired criterion function. The plant model is only used for conversion of the original model into the U-state space expression.

6.3.2 Case study

The dynamic model of the inverted pendulum is derived and implemented in MATLAB/Simulink software. With the help of that computational simulation, the LQR feedback control system is tested and demonstrates the system performance by the proposed control strategy for the inverted pendulum.

In the case studies, a two-dimensional version of the inverted pendulum system consists of a cart and pendulum in which the pendulum is constrained to move in the vertical plane. A linearised model is determined presuming a small deviation from equilibrium and is selected to demonstrate the proposed U-state space control system design approach. The control input is the force F that moves the cart horizontally and outputs are the angular position θ and the position of the cart x. The parameters for a typical inverted pendulum is shown in Table 6.2.

Table 6.2 Parameters of inverted pendulum

m	Mass of the pendulum	0.2 Kg
M	Mass of the cart	0.5 Kg
l	Length to pendulum centre of mass	0.3 m
I	Inertia of the pendulum	$0.006\ \mathrm{Kg.m^2}$
b	Coefficient of friction for cart	0.1 N/M/s
g	Gravitational force	$9.81\ \mathrm{m/s^2}$

Consider an inverted pendulum system in (6.11) as:

$$\begin{bmatrix} \dot{x} \\ \ddot{x} \\ \dot{\Phi} \\ \ddot{\Phi} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & A_1 & A_2 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & A_3 & A_4 & 0 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ \Phi \\ \dot{\Phi} \end{bmatrix} + \begin{bmatrix} 0 \\ B_1 \\ 0 \\ B_2 \end{bmatrix} u \qquad (6.20)$$

where

$$A_1 = \frac{-(I+ml^2)b}{I(M+m)+Mml^2} \quad A_2 = \frac{m^2gl^2}{I(M+m)+Mml^2}$$

$$A_3 = \frac{-mlb}{I(\mathrm{M+m})+\mathrm{Mml}^2} \quad A_4 = \frac{mgl(\mathrm{M+m})}{I(\mathrm{M+m})+\mathrm{Mml}^2}$$

$$B_1 = \frac{(I+ml^2)}{I(M+m)+Mml^2} \quad B_2 = \frac{ml}{I(M+m)+Mml^2}$$

Substituting the parameters into equation (6.20), it gives:

$$\begin{bmatrix} \dot{x} \\ \ddot{x} \\ \dot{\Phi} \\ \ddot{\Phi} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -0.1818 & 2.6755 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & -0.9091 & 31.2136 & 0 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ \Phi \\ \dot{\Phi} \end{bmatrix} + \begin{bmatrix} 0 \\ 1.8182 \\ 0 \\ 4.5455 \end{bmatrix} u \qquad (6.21)$$

The continuous time state space system can be discretised by the factor:

$$x(k+1) = x(k) + Ts * f(x) \qquad (6.22)$$

Let $Ts = 0.01\mathrm{s}$, the discrete time model is obtained as:

$$\begin{bmatrix} x_1(t+1) \\ x_2(t+1) \\ x_3(t+1) \\ x_4(t+1) \end{bmatrix} = \begin{bmatrix} 1 & 0.01 & 0 & 0 \\ 0 & 0.9982 & 0.0268 & 0 \\ 0 & 0 & 1 & 0.01 \\ 0 & -0.0091 & 0.3121 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} 0 \\ 1.8182 \\ 0 \\ 4.5455 \end{bmatrix} u(t) \qquad (6.22)$$

where state vector $x(\mathrm{t}) = \begin{bmatrix} x_1 & x_2 & x_3 & x_4 \end{bmatrix}^T = \begin{bmatrix} x & \dot{x} & \Phi & \dot{\Phi} \end{bmatrix}^T$. The eigenvalues are 1.0000, 0.9437, 0.9990 and 1.0555. This system is unstable because of an eigenvalue larger than one.

Using LQR design, matrices Q and R are setup and the feedback gain matrix can be determined as $K \in \mathbb{R}^{4\times1}$. Thus, the desired closed loop state space model can be expressed as:

$$x_d(\text{t+1}) = A_d x(t) + B_d r(t) \qquad (6.23)$$

where $A_d = \begin{bmatrix} 1 & 0.01 & 0 & 0 \\ 0.0171 & 1.0328 & -0.3156 & -0.0635 \\ 0 & 0 & 1 & 0.01 \\ 0.0427 & 0.0773 & -0.5438 & 0.8412 \end{bmatrix}$ and $B_d = \begin{bmatrix} 0 \\ 0.0182 \\ 0 \\ 0.0455 \end{bmatrix}$.

For equation (6.13), the U-state space model expression of inverted pendulum is derived as:

$$x_{dn}(t+1) = \lambda_{0n}(t) + \lambda_{1n}(t)u(t) \qquad (6.24)$$

where $\lambda_{02}(t) = \begin{bmatrix} 0.0171 & 1.0328 & -0.3156 & -0.0635 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}$ and $\lambda_{12}(t) = 0.0182$.

Then the controller output $u(t-1)$ is obtained by resolving the following equation:

$$u(t) = \frac{x_d(t+1) - \lambda_0(t)}{\lambda_1(t)} \qquad (6.25)$$

The simulation results are presented in Figure from 6.13 to Figure 6.15.

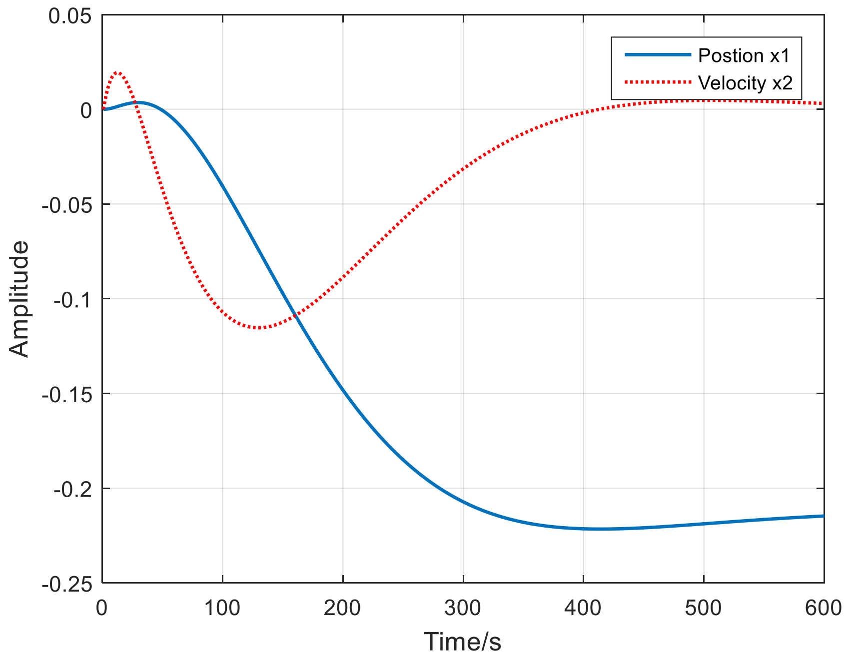

Figure 6.13 Response of cart position and speed

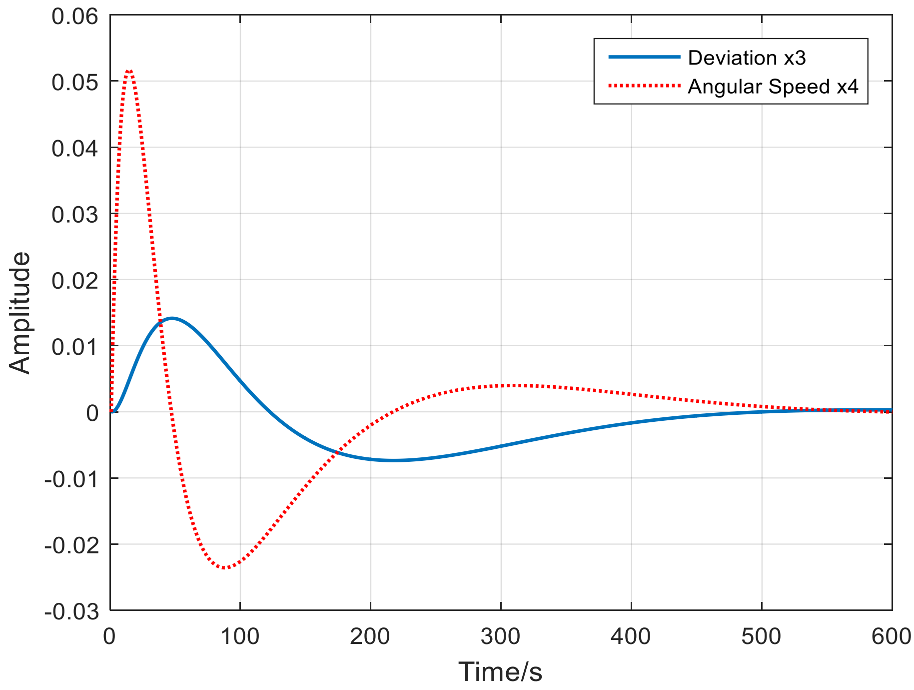

Figure 6.14 Response of pendulum angle and angular speed

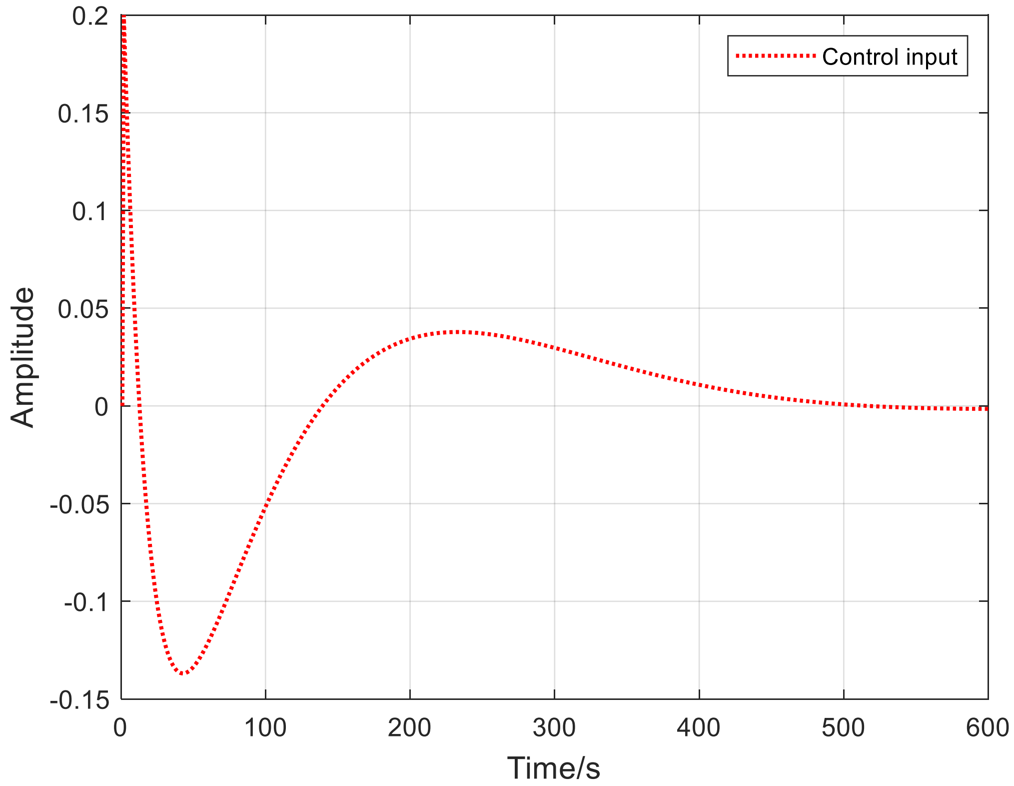

Figure 6.15 Control input u

6.4 U-State Space Control System Design for Nonlinear System

6.4.1 Nonlinear control design

Consider a nonlinear inverted pendulum system (6.8) as:

$$\begin{aligned}(I+ml^2)\ddot{\theta}+mgl\sin\theta &= -ml\ddot{x}\cos\theta \\ (M+m)\ddot{x}+b\dot{x}+ml\ddot{\theta}\cos\theta-ml\dot{\theta}^2\sin\theta &= u\end{aligned} \tag{6.26}$$

where u is control input and the state variables are $x=\begin{bmatrix} x & \dot{x} & \theta & \dot{\theta}\end{bmatrix}$, respectively position, velocity, phase angle and angular speed.

The system (6.26) can be rewritten into nonlinear state space expression as:

$$\Theta_\theta \ddot{x} = (I + ml^2)(\frac{u}{ml\cos\theta} + \theta^2 \frac{\sin\theta}{\cos\theta} - \frac{b\dot{x}}{ml\cos\theta}) + gml\sin\theta$$
$$\Gamma_\theta \ddot{\theta} = \frac{(M+m)g\sin\theta}{\cos\theta} - b\dot{x} + ml\dot{\theta}^2\sin\theta + u \qquad (6.27)$$

where

$$\Theta_\theta = \frac{I(M+m)}{ml\cos\theta} + \frac{l(M+m)}{\cos\theta} - ml\cos\theta$$
$$\Gamma_\theta = \frac{(M+m)(I+ml^2)}{-ml\cos\theta} + ml\cos\theta \qquad (6.28)$$

The discrete time operator is used to converted (6.28) into discrete time system:

$$f(t+1) - f(t) = T_s\dot{f} \qquad (6.29)$$

To clarify the state variables in discrete time system, there are redefined as $x = \begin{bmatrix} x & \dot{x} & \theta & \dot{\theta} \end{bmatrix} = \begin{bmatrix} x & v & \theta & \omega \end{bmatrix}$Then the U-state space prototype can be represented as:

$$\frac{v(t+1)-v(t)}{T_s} = \lambda_0(t) + \lambda_1(t)u_1(t)$$
$$\frac{\omega(t+1)-\omega(t)}{T_s} = \lambda_2(t) + \lambda_3(t)u_2(t) \qquad (6.30)$$

where

$$\lambda_0(t) = \left[(I+ml^2)(\theta(t)^2 \frac{\sin\theta(t)}{\cos\theta(t)} - \frac{bv(t)}{\cos\theta(t)}) + mgl\sin\theta(t) \right] \div \Theta_\theta$$
$$\lambda_1(t) = \frac{(I+ml^2)}{ml\cos\theta(t)\Theta_\theta}$$
$$\lambda_2(t) = \left[\frac{(M+m)g\sin\theta(t)}{\cos\theta(t)} - bv(t) + ml(\omega(t))^2\sin\theta(t) \right] \div \Gamma_\theta$$
$$\lambda_3(t) = \frac{1}{\Gamma_\theta}$$

For example, the linear design (LQR linear design (6.15)) method is used to control the nonlinear system (6.30). The desired closed loop is:

$$\boldsymbol{x}_d(t) = \boldsymbol{A}_c\boldsymbol{x}(t) + \boldsymbol{B}_c\boldsymbol{r}(t) \qquad (6.31)$$

where $A_c = A - BK_{LQR}$ and $\boldsymbol{B}_c = \boldsymbol{B}$ are linearised state space matrices.

The following task is to obtain the desired controller output $\boldsymbol{u}(t-1)$. By resolving one of the roots of equation (6.30), it can be expressed in terms of:

$$\begin{aligned} u_1(t) &= \left[\frac{v_d(t+1) - v(t)}{T_s} - \lambda_0(t)\right] \div \lambda_1(t) \\ u_2(t) &= \left[\frac{\omega_d(t+1) - \omega(t)}{T_s} - \lambda_2(t)\right] \div \lambda_3(t) \end{aligned} \qquad (6.32)$$

The nonlinear control must consider of two control inputs rather than one control input in linearized case.

6.4.2 Case study

Substituting the parameters (Table 6.2) into equation (6.27), it gives:

$$\begin{aligned} \Theta_\theta \ddot{x} &= 0.024\left(\frac{u}{0.06\cos\theta} + \theta^2 \frac{\sin\theta}{\cos\theta} - \frac{0.1\dot{x}}{0.06\cos\theta}\right) + 0.5886\sin\theta \\ \Gamma_\theta \ddot{\theta} &= \frac{6.867\sin\theta}{\cos\theta} - 0.1\dot{x} + 0.06\dot{\theta}^2 \sin\theta + u \end{aligned} \qquad (6.33)$$

Using the same LQR controller in section 6.3.2, matrices Q and R are setup and the feedback gain matrix can be determined as $K \in \mathbb{R}^{4\times 1}$. Thus, the desired closed loop state space model can be expressed as:

$$\boldsymbol{x}_d(\mathrm{t+1}) = \boldsymbol{A}_d\boldsymbol{x}(t) + \boldsymbol{B}_d\boldsymbol{r}(t) \qquad (6.34)$$

where $A_d = \begin{bmatrix} 1 & 0.01 & 0 & 0 \\ 0.0171 & 1.0328 & -0.3156 & -0.0635 \\ 0 & 0 & 1 & 0.01 \\ 0.0427 & 0.0773 & -0.5438 & 0.8412 \end{bmatrix}$ and $B_d = \begin{bmatrix} 0 \\ 0.0182 \\ 0 \\ 0.0455 \end{bmatrix}$.

Then the U-state space expression in (6.30) can be written as:

$$
\begin{aligned}
x_{d2}(t+1) &= \lambda_0(t) + \lambda_1(t)u_1(t) \\
x_{d4}(t+1) &= \lambda_2(t) + \lambda_3(t)u_2(t)
\end{aligned}
\qquad (6.35)
$$

where

$$
\lambda_0(t) = \left[0.024(\theta(t)^2 \frac{\sin\theta(t)}{\cos\theta(t)} - \frac{0.1v(t)}{\cos\theta(t)}) + 0.5886\sin\theta(t) \right] \div (\frac{0.0042}{0.06\cos\theta(t)} + \frac{0.21}{\cos\theta(t)} - 0.06\cos\theta(t))
$$

$$
\lambda_1(t) = \frac{0.024}{0.06\cos\theta(t)(\frac{0.0042}{0.06\cos\theta(t)} + \frac{0.21}{\cos\theta(t)} - 0.06\cos\theta(t))}
$$

$$
\lambda_2(t) = \left[\frac{6.867\sin\theta(t)}{\cos\theta(t)} - 0.1v(t) + 0.06(\omega(t))^2 \sin\theta(t) \right] \div (\frac{0.203}{-0.06\cos\theta(t)} + 0.06\cos\theta(t))
$$

$$
\lambda_3(t) = \frac{1}{\frac{0.203}{-0.06\cos\theta(t)} + 0.06\cos\theta(t)}
$$

Then the controller output $u(t-1)$ is obtained by resolving the following equation:

$$
\begin{aligned}
u_1(t) &= \left[\frac{v(t+1) - v(t)}{T_s} - \lambda_0(t) \right] \div \lambda_1(t) \\
u_2(t) &= \left[\frac{\omega(t+1) - \omega(t)}{T_s} - \lambda_2(t) \right] \div \lambda_3(t)
\end{aligned}
\qquad (6.36)
$$

The simulation results are presented in Figures (6.16 to 6.18) below:

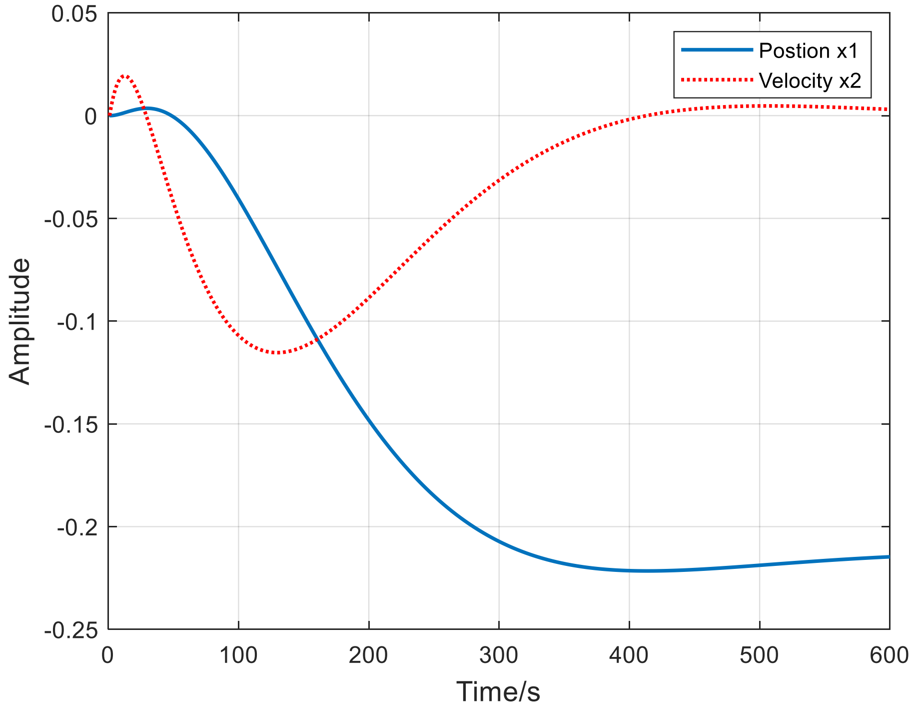

Figure 6.16 Response of cart position and speed (nonlinear model)

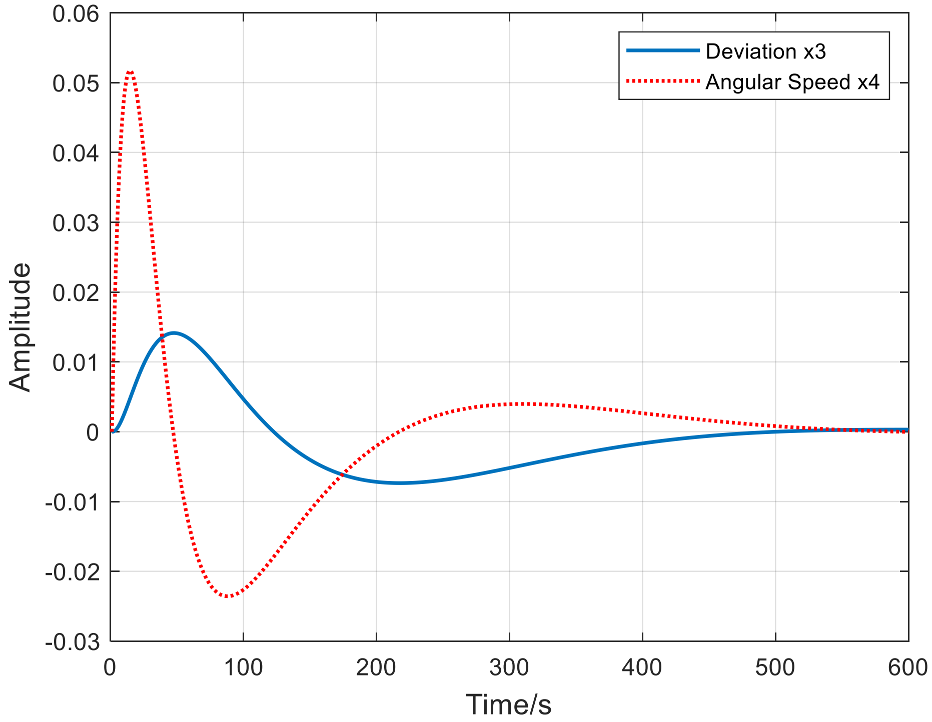

Figure 6.17 Response of pendulum angle and angular speed (nonlinear model)

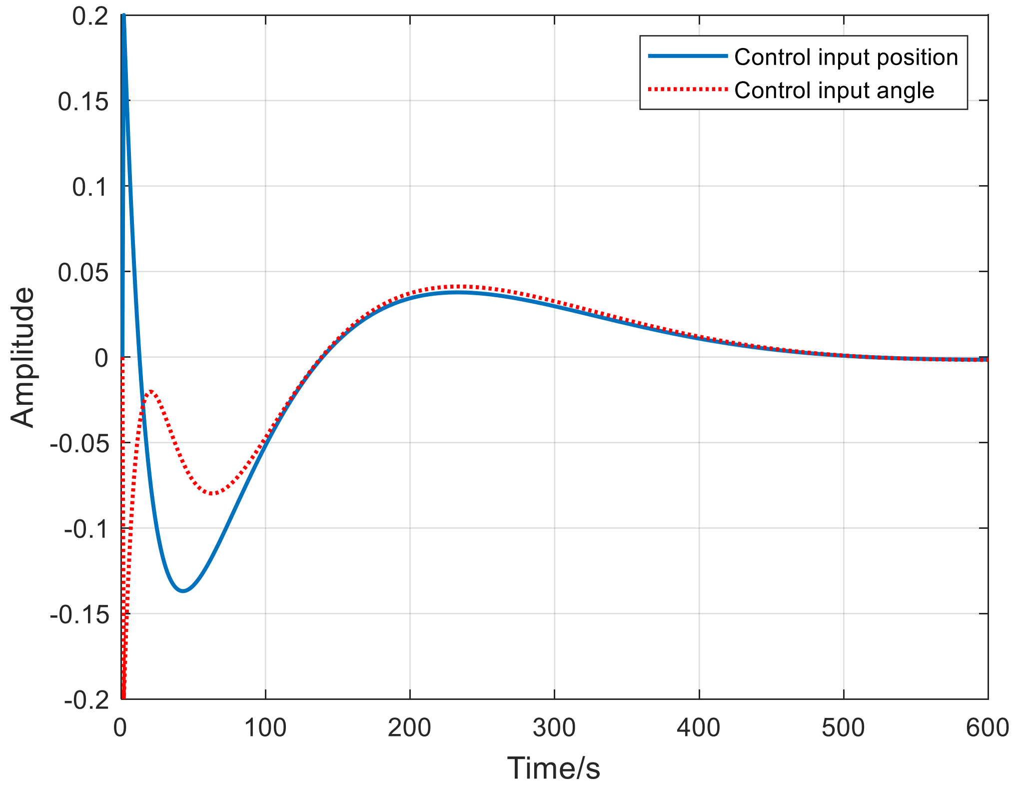

Figure 6.18 Control input cart position and angle (nonlinear model)

From the simulation results, it can be found that both linear and nonlinear control system perform the same system output based on U-state space control. For the nonlinear control system, it needs the stabilisation control loop to guarantee the stable internal dynamic.

6.5 Conclusions

In this chapter, a nonlinear dynamic model of an inverted pendulum is introduced as the controlled subject for the proposed U-state space control system design. U-state space model is extended from SISO system to stabilise the SIMO state space system. The U-state space design approach for SIMO system is similar to that for SISO design. The closed loop characteristic equation is determined by the LQR optimal control approach. Then the control input can be obtained by solving the related function. In simulation studies, the control performance is given as computational experiment results via MATLAB/Simulink. There may be some constraints for the control signals depending on the different types of hardware.

From simulation results, it can be inspected that the curves of the pendulum's angle and the cart's position is not satisfactory. The settling times and rise time are needed to be further improvement. The cart's final position is also not near the desired location (0.2). It should be mentioned that these errors are leading by LQR design approach (desired target).

Chapter 7

Conclusions and Further Work

7.1 Conclusions

To overall aim of this PhD research was to propose the robust analysis and design of U-model based control systems. Moreover, to extend the U-model approaches to state space form and to establish an enhanced U-model based state space platform to use the mature linear controller design approaches directly on nonlinear control system design and give a case study on the development of control system design for the standard quad-rotor dynamic model.

In this project, a general control-oriented polynomial model framework called U-model and the corresponding pole placement control system design has been introduced to be the fundamental methodologies. Based on U-model, this design approach is not only simplified by using the linear controller design method directly for the nonlinear dynamic model but also can be used to obtain the closed loop system with linearised input-output relationship called U-block model. Considering the uncertainty U-model based pole placement control system; the robust stability margin is obtained by the feasibility of LMI conditions. Therefore, the LMI based robust control system is designed to improve the robustness of the original U-model based control system.

Many types of research of the U-model based control system design are focused on how to demonstrate different linear control methods on nonlinear control systems for nonlinear polynomial models. In modern control engineering, the state space realisation is widely used for presenting dynamic industrial applications. This study is extended to the U-model techniques into the state space control system design for establishing the U-state space platform. The new platform provides a generalised representation of a broad range of nonlinear state space models and simplifies nonlinear control design procedures.

The contents of this PhD thesis can be summarised as follows.

Chapter 2 briefly introduces the description of U-model, which is followed by the literature review of U-model based pole placement control system design; introduced to represent the fundamental methodologies. Also, other U-model based control system designs are also analysed to show the development of the U-model approach during last decade.

In chapter 3, a framework named U-block model, is defined as an input output closed loop transfer function of the U-control systems (such as U-pole placement design). It is easily converted into an equivalent linear transfer function or state space realisation. Within the U-block model, the procedure for LMI based robust stability analysis of U-model pole placement control system is presented to determine the stability range. Then, an enhanced U-model LMI based robust control system is designed to enlarge this robust stability range. An LMI based enhanced H_∞ output feedback controller of U-block model control system has been proposed to improve robust stability for the developed U-model based control system. The LMI based robust controller design approach is difficult to directly implement on nonlinear polynomial models. Finally, the computational simulation results are presented to verify the effectiveness of the enlarged robust stability bound.

Chapter 4 establishes a U-state space realisation which is converted from the nonlinear state space dynamic model. Within the U-state space platform, the controller for the nonlinear control system is developed using linear state feedback approach. The stability of the designed U-state space control system is analysed by using zero dynamic and relative degree. In order to implement the U-model design approach in state space control system, U-state space expression is established to apply linear control system design method directly for linear/nonlinear state space model. Through the numerical simulations, it can be inspected that the system performance of designed U-state space control system achieves the desired requirements/targets.

In chapter 5, the proposed U-state space control system design approach is applied to develop the controller for a nonlinear quad-rotor rotorcraft model. Firstly, a brief introduction to quad-rotor modelling is studied to test the viability of U-state space design approach. A typical nonlinear quad-rotor model (5.17) is selected as the dynamic plant for implementation. The optimal control algorithm LQR is applied to find a desired closed loop dynamic matrix. Then the simulation results of the navigation and control architecture for the quad-rotor are presented to highlight the application and performance of the proposed control laws. Finally, an application study of U-state space control system design for the standard nonlinear quad-rotor model has been proposed to validate the effectiveness and efficiency of the proposed U-state space approach.

In chapter 6, inverted pendulum control system is developed using U-state space control approach. The modelling of the inverted pendulum is presented to be controlled subject for U-control method. The desired state vector (closed loop specification) is determined by LQR design. The simulation results has been proposed to stabilise the inverted pendulum with satisfied performance.

7.2 List of publications based on this study

Conference papers

Peng, Y., Liu, X., Zhu, Q.M., Xu, F. and Zhao, D. (2013) The initial robustness analysis of designed U-pole placement control systems. In: 2013 *International Conference of Modelling, Identification & Control* (ICMIC), Cairo, Egypt, 31 August - 2 September 2013., pp. 253-258 Available from: http://eprints.uwe.ac.uk/27533

Liu, X., Peng, Y., Zhu, Q.M. and Narayan, P. (2014) U-model based LMI robust controller design. In: 2014 33rd Chinese Control Conference (CCC), Nanjing, China, 28-30 July 2014, pp. 2200-2205 Available from: http://eprints.uwe.ac.uk/27530

Liu, X., Zhu, Q.M., Narayan, P. and Yang, Y. (2015) The first stage studies of U-state space control system design. In: 2015 34th Chinese Control Conference (CCC), Hangzhou, China, 28-30 July 2015, pp. 946-951 Available from: http://eprints.uwe.ac.uk/27532

Liu, X., Zhu, Q.M. and Narayan, P (2015) Case studies on U-state space control system design for aircraft dynamic model, *Modelling, Identification and Control (ICMIC), 2015 7th International Conference on*, Sousse, Tunisia, 2015, Available from: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=7409439&isnumber=7409325

Liu, X., Zhu, Q.M. and Narayan, P (2016) Case Studies on U-State Space Control System Design for Quadrotor Model, *International Conference of Modelling, Identification & Control*, 2016 8th International Conference on, Algiers, Algeria, 2016, Available from: http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=7804234

Liu, X., Zhu, Q.M. and Narayan, P. U-model enhanced control of inverted pendulum. to be submitted

Zhu, Q.M., Liu, X. and Narayan, P. U-model based control applications-simulated (Book). to be edited

7.3 Future work

Since the first official publication in 2002, U-model based control system design procedures have gone through a decade to research and development. Almost all the researchers have focused on control system design within U-model polynomial framework. The study of U-state space model is in the first stage. There are many potential expansions of this study to be summarised in this section. The potential expansion of the present study can be summarised as follows.

- With U-block realisation, nonlinear polynomial models can be easily converted into linear state space models (Zhu, 2016). This linear state space form provides a solution for the further robust control system analysis and design using linear design approaches. Although some of the existing robust control algorithms (ElBsat and Yaz, 2013; Zemouche and Boutayeb, 2013) will provide a useful reference for the new development of U-block model, more stability analysis methods/theorems should be studied to give more powerful evidence for U-control approaches.

- Instead of linearisation at the operating point, nonlinear modelling approaches could be further analysed, developed and applied to quad-rotor rotorcraft system analysis and stabilised design, e.g. nonlinear adaptive controller, nonlinear robust controller.

- The improved tracking controller design algorithm could be applied to backstepping approach to determine the better stability and performance for the designed U-model based control systems.

- Even though the bottleneck problem has been resolved in U-state space platform design, the selected feedback design in this project is just for feasibility test. Further comprehensive studies and simulation bench tests should be conducted in the future concerning many leading research results (Bartolini and Punta, 2012; Moreno and Osorio, 2012).

- Within such U-state space framework, proper stability analysis and stabilisation methods should be considered in conjunction with Lyapunov stability analysis theorem (Johansen, 2000; Li and Khalil, 2012; Zhou et al., 2011; Zhu, 2016).

Appendix A

Procedures of Linearization for Quadrotor Dynamic Model

In order to design an LQR controller, it is necessary to have a linear approximation of the nonlinear model around an operating point (equilibrium point). The following procedures are used for obtaining the linearized model of quadrotor.

Determine the linearization point of the vehicle for $\varphi = \theta = \psi = 0$ and $x = y = z = 0$.

The dynamic model of quadrotor can be expressed as

$$\begin{bmatrix} x \\ y \\ z \\ v_x \\ v_y \\ v_z \\ \phi \\ \theta \\ \psi \\ \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} \rightarrow \begin{bmatrix} x_0 + \Delta x \\ y_0 + \Delta y \\ z_0 + \Delta z \\ v_{x_0} + \Delta v_x \\ v_{y_0} + \Delta v_y \\ v_{z_0} + \Delta v_z \\ \phi_0 + \Delta\phi \\ \theta_0 + \Delta\theta \\ \psi_0 + \Delta\psi \\ \omega_{x_0} + \Delta\omega_x \\ \omega_{y_0} + \Delta\omega_y \\ \omega_{z_0} + \Delta\omega_z \end{bmatrix} = \begin{bmatrix} \Delta x \\ \Delta y \\ z_0 + \Delta z \\ \Delta v_x \\ \Delta v_y \\ \Delta v_z \\ \Delta\phi \\ \Delta\theta \\ \Delta\psi \\ \Delta\omega_x \\ \Delta\omega_y \\ \Delta\omega_z \end{bmatrix}$$

Consider the differential equation of v_x

$$\dot{v}_x = \frac{-k_d}{m} v_x + \frac{kc_m}{m}(\sin\psi\sin\phi + \cos\psi\cos\phi\sin\theta)(v_1^2 + v_2^2 + v_3^2 + v_4^2)$$

It can be rewritten as

$$v_{x_0} + \Delta\dot{v}_x = \frac{-k_d}{m}\left(v_{x_0} + \Delta v_x\right) + \frac{kc_m}{m}(\Delta\psi\,\Delta\phi + \Delta\theta)\left(v_{1_0}^2 + \Delta v_1^2 + v_{2_0}^2 + \Delta v_2^2 + v_{3_0}^2 + \Delta v_3^2 + v_{4_0}^2 + \Delta v_4^2\right)$$

$$\Delta\dot{v}_x = \frac{-k_d}{m}(\Delta v_x) + \frac{kc_m}{m}(\Delta\psi\,\Delta\phi + \Delta\theta)\left(v_{1_0}^2 + \Delta v_1^2 + v_{2_0}^2 + \Delta v_2^2 + v_{3_0}^2 + \Delta v_3^2 + v_{4_0}^2 + \Delta v_4^2\right)$$

$$\Delta\dot{v}_x = \frac{-k_d}{m}(\Delta v_x) + \frac{kc_m}{m}(\Delta\theta)\left(v_{1_0}^2 + v_{2_0}^2 + v_{3_0}^2 + v_{4_0}^2\right)$$

Set Disturbances to zero

$$0 = -0 + 0\left(v_{1_0}^2 + v_{2_0}^2 + v_{3_0}^2 + v_{4_0}^2\right)$$

Substitute into Linearized Equation

$$\Delta\dot{v}_x = \frac{-k_d}{m}(\Delta v_x) + \frac{kc_m}{m}\left(v_{1_0}^2 + v_{2_0}^2 + v_{3_0}^2 + v_{4_0}^2\right)(\Delta\theta)$$

Similarly it can be found that

$$\Delta\dot{v}_y = \frac{-k_d}{m}\left(\Delta v_y\right) + \frac{kc_m}{m}\left(v_{1_0}^2 + v_{2_0}^2 + v_{3_0}^2 + v_{4_0}^2\right)(\Delta\phi)$$

Consider the differential equation of v_z

$$\dot{v}_z = \frac{-k_d}{m} v_z - g + \frac{Kc_m}{m}(\cos\theta\cos\phi)(v_1^2 + v_2^2 + v_3^2 + v_4^2)$$

$$v_{z_0} + \Delta\dot{v}_z = \frac{-k_d}{m}\left(v_{z_0} + \Delta v_z\right) - g_0 + \frac{Kc_m}{m}(\cos(\theta_0 + \Delta\theta)\cos(\phi_0 + \Delta\phi))\left(v_{1_0}^2 + \Delta v_1^2 + v_{2_0}^2 + \Delta v_2^2 + v_{3_0}^2 + \Delta v_3^2 + v_{4_0}^2 + \Delta v_4^2\right)$$

$$v_{z_0} + \Delta\dot{v}_z = \frac{-k_d}{m}(\Delta v_z) - g_0 + \frac{Kc_m}{m}\left(v_{1_0}^2 + \Delta v_1^2 + v_{2_0}^2 + \Delta v_2^2 + v_{3_0}^2 + \Delta v_3^2 + v_{4_0}^2 + \Delta v_4^2\right)$$

Set Disturbances to zero

$$0 = -g_0 + \frac{Kc_m}{m}\left(v_{1_0}^2 + v_{2_0}^2 + v_{3_0}^2 + v_{4_0}^2\right)$$

Substitute into Linearised Equation

$$\Delta\dot{v}_z = \frac{-k_d}{m}(\Delta v_z) + \frac{Kc_m}{m}(\Delta v_1^2 + \Delta v_2^2 + \Delta v_3^2 + \Delta v_4^2)$$

Appendix B

Program of U-model based Control System Design and Analysis

The following programmes are used for computational simulation of the proposed U-model based control system design approaches.

LMI based robust control system design for developed U-model based pole placement

```
% LMI design
% the script started 28/01/2014
% updated 18/05/2014
% build up state space matrices
a=[0 1;-0.4966 1.3205]; b1= [1;0];b2=[0;1];c1=[1
0];c2=[0.1761 0];d11=0;d12=0;d21=1;d22=0;I=[1 0;0 1];
P=ltisys(a,[b1 b2],[c1 ;c2],[d11 d12;d21 d22]);
[gopt,K]=dhinflmi(P,[1 1],3);
```

```
[Ak,Bk,Ck,Dk]=ltiss(K);
sysc=ss(Ak,Bk,Ck,Dk);
[num,den]=tfdata(sysc,'v');
sysczz=tf(num,den,1);
setlmis([])

%setup LMI equations
setlmis([])
X=lmivar(1,[2,1]);
Y=lmivar(1,[2,1]);
%LMI 1
lmiterm([1 1 1 X],a,a');%ARA'
lmiterm([1 1 1 X],-1,1);
%lmiterm([1 1 2 X],1,b1);
%lmiterm([1 1 3 0],c1');
lmiterm([1 2 1 X],c1,a');%C1RA'
lmiterm([1 2 2 0],-3);
lmiterm([1 2 2 X],c1, c1');
%lmiterm([1 2 3 0],d11');
lmiterm([1 3 1 0],b1');
lmiterm([1 3 2 0],d11');
lmiterm([1 3 3 0],-3);

%LMI 2
lmiterm([2 1 1 Y],a',a);
lmiterm([2 1 1 Y],-1,1);
lmiterm([2 2 1 Y],b1',a);
lmiterm([2 2 2 0],-3);
lmiterm([2 2 2 Y],b1',b1);
%lmiterm([2 2 1 Y],c1,1);
%lmiterm([2 2 2 0],-1);
%lmiterm([2 2 3 0],d11);
lmiterm([2 3 1 0],c1);
lmiterm([2 3 2 0],d11');
lmiterm([2 3 3 0],-3);
%LMI 3
lmiterm([-3 1 1 X],1,1);
lmiterm([-3 2 1 0],1);
lmiterm([-3 2 2 Y],1,1);
%Compute solution to given system of LMIs
[copt,xopt]=feasp(lmisys);
X=dec2mat(lmisys,xopt,X);
Y=dec2mat(lmisys,xopt,Y);
%display results
X
Y
```

U-state space model base control system design for nonlinear mathematical models

```
%U state space control system (Case I)
% the script started 28/01/2015
% updated 03/02/2015
% Reference:
%           1.  Q.M. Zhu and L.Z. Guo, a pole placement
controller
%                    for nonlinear dynamic plants, 2002.
%  Nonlinear state space model
%   x(t+1)=f(x(t),u(t))
%  Desired state function
%  Bench test model
%  Nonlinear state space model
%       x1(t+1)=x1(t)^2+x2(t)
%       x2(t+1)=x2(t)*cos(x1(t))+u(t)
%  Specified desired Ad Bd
%  Desired state space equations
%       xd(t+1)=Ad*x(t)+Bd*W(t)
%  obtain controller output u(t)
%       u(t)=xd2(t+1)-x2(t)*cosx1(t)-v(t)
%

clc,clear
ns=100;
x=zeros(1,3);x1=x;x2=x;xd1=x;xd2=x;u=x;w=0;
%initialization
x1(1)=-0.5;x2(1)=-0.5;%inital state variables x1,x2
xd1(1)=0;xd2(1)=0;%desired state xd
u(1)=0;
for t=1:ns
%step 1 obtain desire state variables xd(t)

xd1(t+1)=x2(t);
xd2(t+1)=-0.4966*x1(t)+1.3205*x2(t)+w;

%step 2 solve controller output u(t)
u(t)=xd2(t+1)-x2(t)*cos(x1(t));

%step 3 update state variables from state equations
x1(t+1)=0.1*x1(t)^2+x2(t);
x2(t+1)=x2(t)*cos(x1(t))+u(t);
%y(t)=0.1761*x1(t);
end

%step 4 display simulation results
t=1:ns;
figure(1)
```

```
plot(t,x1(1:ns))
xlabel('Time')
ylabel('x1(t)')
figure(2)
plot(t,x2(1:ns))
xlabel('Time')
ylabel('x2(t)')
figure(3)
plot(t,u(1:ns))
xlabel('Time')
ylabel('Controller output u(t)')
%figure(4)
%plot(t,y(1:ns))


%% U state space control system design (case II)
% the script started 28/01/2015
% updated 18/03/2015
% Reference:
%           1.  Zhu, Q.M.and Guo, L.Z., a pole placement
controller
%                   for nonlinear dynamic plants, 2002.
%           2.  Slotine, J.J.E. and Li, W., Applied
Nonlinear Control,
%                   Prentice-Hall, 1991.
%  Nonlinear discrete time state space model
%   x(t+1)=f(x(t))+g(x)u(t)
%  Desired state function
%  xd(t+1)=Ax(t)+Bw(t)
%  y(t)=h(x)
%  obtain controller output
%  u(t)=(xd(t+1)-f(x(t))-w(t))/g(x)
%  Bench test model
%  Nonlinear continuous state space model
%       x1'=-2x1+x2+sinx1
%       x2'=x2(t)-x2*cos(x1)+u*cos(2x1)
%  Discretization by Eural method
%  y_n+1=y_n+h*f_n
%  Nonlinear discrete time state space model
%       x1(t+1)=-x1(t)+x2(t)+sin(x1(t))
%       x2(t+1)=x2(t)-x2(t)*cos(x1(t))+u(t)*cos(2x1(t))
%  Specified desired Ad Bd
%  Desired state space equations
%       xd(t+1)=Ad*x(t)+Bd*V(t)
%  obtain controller output u(t)
%       u(t)=(xd2(t+1)-x2(t)+x2(t)*cosx1(t))/cos(2*x(x1))-
w(t)
%
```

```
clc,clear
ns=100;%length of sample
x=zeros(1,3);x1=x;x2=x;xd1=x;xd2=x;u=x;
w=0;%reference input

%initialisation
x1(1)=1;x2(1)=0.5;%inital state variables x1,x2
xd1(1)=0;xd2(1)=0;%desired initial state xd
u(1)=0;%u(0)=0
for t=1:ns

%step 1 obtain desired state variables xd(t)
xd1(t+1)=x2(t);
xd2(t+1)=-0.4966*x1(t)+1.3205*x2(t)+w;

%step 2 determine controller output u(t)
u(t)=(xd2(t+1)-x2(t)+x2(t)*cos(x1(t))-w)/cos(2*x1(t));

%step 3 update state variables from state equations
x1(t+1)=-x1(t)+x2(t)+sin(x1(t));
x2(t+1)=x2(t)-x2(t)*cos(x1(t))+u(t)*cos(2*x1(t));

end

%step 4 display simulation results
t=1:ns;
figure(1)
plot(t,x1(1:ns))
xlabel('Time')
ylabel('x1(t)')
figure(2)
plot(t,x2(1:ns))
xlabel('Time')
ylabel('x2(t)')
figure(3)
plot(t,u(1:ns))
xlabel('Time')
ylabel('Controller output u(t)')

% U-state space design for f-16 model
% desired closed loop Ad
% the script started 08/05/2015
% updated 18/09/2015
% Ad=[0.5 -0.4 -0.3 1;
%     0.1 0.8 0.9 -0.2;
%     0.3 -0.6 0.7 1;
%     0.6  0  0  0.3 ]
```

```
%% initialization
x=zeros(1,3);x1=x;x2=x;x3=x;x4=x;xd1=x;xd2=x;xd3=x;xd4=x;u=x
;w=0;
%state varariables are respectively V_T_trim, alpha_trim,
q_trim and z_E_trim
x1(1)=0;x2(1)=1;x3(1)=0;x4(1)=0;u(1)=0;
ns=100;
%sysd1=ss(Ad,Bd,Cd,Dd,dt);
for t=1:ns
%step 1 obtain desired state variables xd(t)
%  Ad=[0.5000   -0.4000   -0.3000    1.0000
%      0.1000    0.8000    0.9000   -0.2000
%      0.3000   -0.6000    0.7000    1.0000
%     -0.2208    0.7125    0.1532   -0.8282]
% xd1(t+1)=x2(t);
% xd2(t+1)=x3(t);
% xd3(t+1)=x4(t);
xd4(t+1)=-0.2208*x1(t)+0.7125*x2(t)+0.1532*x3(t)-
0.8282*x4(t);

%step 2 determine controller output u(t)
u(t)=xd4(t+1)-0.6*x1(t)-0.3*x4(t);

% update state variables
x1(t+1)=0.5*x1(t)-0.4*x2(t)-0.3*x3(t)+x4(t) ;
x2(t+1)=0.1*x1(t)+0.8*x2(t)+0.9*x3(t)-0.2*x4(t)     ;
x3(t+1)= 0.3*x1(t)-0.6*x2(t)+0.7*x3(t)+x4(t)   ;
x4(t+1) =0.6*x1(t)+0.3*x4(t)+u(t);

end
%step 4 display simulation results
t=1:ns;
% subplot(321),plot(t,u1,'r'),box off
% ylabel('u1(t)'),xlabel('t')
subplot(322),plot(t,u,'r'),box off
ylabel('u(t)'),xlabel('t')
subplot(323),plot(t,x1(1:ns)),box off
ylabel('x1(t)'),xlabel('t')
subplot(324),plot(t,x2(1:ns)),box off
ylabel('x2(t)'),xlabel('t')
subplot(325),plot(t,x3(1:ns)),box off
ylabel('x3(t)'),xlabel('t')
subplot(326),plot(t,x4(1:ns)),box off
ylabel('x4(t)'),xlabel('t')
```

U-state space model based control system design for MIMO quadrotor model

```
% U state space control system design for quadrotor
% the script started 28/01/2015
% updated 28/05/2016
% Reference:
%           1.  Zhu, Q.M.and Guo, L.Z., a pole placement
controller
%                   for nonlinear dynamic plants, 2002.
%           2.  Stevens, B.L. and Lewis, F.L, Aircraft
control and
%                   simulation, 2003.
%
%  Nonlinear discrete time state space model
%   x(t+1)=f(x(t))+g(x)u(t)
%  Desired closed loop state function
%  xd(t+1)=Ax(t)+Bw(t)
%  y(t)=h(x)
%  obtain controller output
%  u(t)=(xd(t+1)-f(x(t))-w(t))/g(x)
%  Specified desired Ad Bd by LQR design
%  Desired state space equations
%       xd(t+1)=Ad*x(t)+Bd*V(t)
%  obtain controller output u(t)
%       u(t)=(xd2(t+1)-lamda_0(t))/lamda_1(t))
%

clc,clear
%% Quadcopter parameters
% mass of the quadcopter - m (kg)
m = 0.5;

% radius of the quadcopter - L (m)
L = 0.25;

% propellor lift coefficient - k (N s^2)
k = 3e-6;

% propellor drag coefficient - b (N m s^2)
b = 1e-7;

% Gravity - g (m/s^2)
g = 9.81;

% Air friction coefficient - kd (Kg/s)
kd = 0.25;

% Inertia about xb axis - Ixx (Kg m^2)
```

```
Ixx = 5-3;

% Inertia about yb axis - Iyy (Kg m^2)
Iyy = 5-3;

% Inertia about zb axis - Izz (Kg m^2)
Izz = 1-2;

% Motor constant - cm (v^-2 s^-2)
Cm = 1e4;

v1 = 1;
v2 = 1;
v3 = 1;
v4 = 1;
VS = v1^2+v2^2+v3^2+v4^2;
%% initialisation
xs=zeros(12,3);
x=zeros(1,3);
% state variables
x1=x;x2=x;x3=x;x4=x;x5=x;x6=x;x7=x;x8=x;x9=x;x10=x;x11=x;x12
=x;xd=x;
xs(1,:)=x1;xs(2,:)=x2;xs(3,:)=x3;xs(4,:)=x4;xs(5,:)=x5;xs(6,
:)=x6;xs(7,:)=x7;xs(8,:)=x8;xs(9,:)=x9;xs(10,:)=x10;xs(11,:)
=x11;xs(12,:)=x12;
% control inputs v1 v2 v3 v4
u1=x;u2=x;u3=x;u4=x;w=0;
%state vararariables are respectively V_T_trim, alpha_trim,
q_trim and z_E_trim
%x1(1)=0;x2(1)=1;x3(1)=1;x4(1)=0;u1=1;u2(1)=0;
Ts=0.5;%sampling time
ns=100;
%% linear model
% x = [x, y, z, vx, vy, vz, phi, theta, psi, wx, wy, wz]

A = [0, 0, 0, 1,     0,     0,     0, 0, 0, 0, 0, 0;...
     0, 0, 0, 0,     1,     0,     0, 0, 0, 0, 0, 0;...
     0, 0, 0, 0,     0,     1,     0, 0, 0, 0, 0, 0;...
     0, 0, 0, -kd/m, 0,     0,     0, ((k*Cm)/m)*VS, 0, 0,
0, 0;...
     0, 0, 0, 0,     -kd/m, 0,     ((k*Cm)/m)*VS, 0, 0, 0,
0, 0;...
     0, 0, 0, 0,     0,     -kd/m, 0, 0, 0, 0, 0, 0;...
     0, 0, 0, 0,     0,     0,     0, 0, 0, 1, 0, 0;...
     0, 0, 0, 0,     0,     0,     0, 0, 0, 0, 1, 0;...
     0, 0, 0, 0,     0,     0,     0, 0, 0, 0, 0, 1;...
     0, 0, 0, 0,     0,     0,     0, 0, 0, 0, 0, 0;...
     0, 0, 0, 0,     0,     0,     0, 0, 0, 0, 0, 0;...
```

```
     0,  0,  0,  0,     0,      0,      0,  0,  0,  0,  0,  0];


B = [0,            0,             0,              0;...
     0,            0,             0,              0;...
     0,            0,             0,              0;...
     0,            0,             0,              0;...
     0,            0,             0,              0;...
     (k*Cm/m),     (k*Cm/m),      (k*Cm/m),       (k*Cm/m);...
     0,            0,             0,              0;...
     0,            0,             0,              0;...
     0,            0,             0,              0;...
     (L*k*Cm)/Ixx, 0,             -(L*k*Cm)/Ixx,  0;...
     0,            (L*k*Cm)/Iyy,  0,              -
(L*k*Cm)/Iyy;...
     (b*Cm)/Izz,   -(b*Cm)/Izz,   (b*Cm)/Izz,     -
(b*Cm)/Izz];

C = eye(12);
D = zeros(12, 4);
%% discrete time state space model
%  sys = ss(A,B,C,D);
%  sysd = c2d(sys,Ts);
%  [A_Hov,B_Hov,C_Hov,D_Hov,Ts]= ssdata(sysd);
A_Hov=eye(12)+Ts*A;
B_Hov=Ts*B;
C_Hov=eye(12)+Ts*C;
D_Hov=D;
%% LQR Control for Desired closed loop
v_max = 10;
r_input = (1/v_max)^2;

xy_max = 0.5;
xy_max_opt = (1/xy_max)^2;
z_max = 0.2;
z_max_opt = (1/z_max)^2;
v_max = 0.1;
v_max_opt = (1/v_max)^2;
angle_max = 5;
ang_max_opt = (1/angle_max)^2;
w_max = 5;
w_max_opt = (1/w_max)^2;

Q = zeros(12);
Q(1,1) = xy_max_opt;
Q(2,2) = xy_max_opt;
Q(3,3) = z_max_opt;
Q(4,4) = v_max;
```

```
Q(5,5) = v_max;
Q(6,6) = v_max;
Q(7,7) = ang_max_opt;
Q(8,8) = ang_max_opt;
Q(9,9) = ang_max_opt;
Q(10,10) = w_max_opt;
Q(11,11) = w_max_opt;
Q(12,12) = w_max_opt;

R = [r_input, 0, 0, 0;...
     0, r_input, 0, 0;...
     0, 0, r_input, 0;...
     0, 0, 0, r_input];

%[K,S,e] = lqr(A_Hov,B_Hov,Q,R);

%% discrete LQR
[Kd,Sd,ed] = dlqr(A_Hov,B_Hov,Q,R);
%% Closed loop matrix
 Ac=A_Hov-B_Hov*Kd*C_Hov; %%% X_dot=(A-BKC)X+B*V
 Bc=0;
 Cc=C_Hov;
 Dc=D_Hov;
%% Initial States
xs(:,1) = [0 0 -1 0 0 0 45*(pi/180) 45*(pi/180) 45*(pi/180)
0 0 0]';
%sysd1=ss(Ad,Bd,Cd,Dd,dt);
for t=2:ns
%step 1 obtain desired state variables xd(t)
xd1(t)=Ac(6,:)*xs(:,t-1);
xd2(t)=Ac(10,:)*xs(:,t-1);
xd3(t)=Ac(11,:)*xs(:,t-1);
xd4(t)=Ac(12,:)*xs(:,t-1);
%step 2 determine controller output u(t)
%% U expression yd=lamda_0+lamda_1*u
 lamda1=[A_Hov(6,:)*xs(:,t-1), k*Cm/m];
 lamda2=[A_Hov(10,:)*xs(:,t-1), (L*k*Cm)/Ixx];
 lamda3=[A_Hov(11,:)*xs(:,t-1), (L*k*Cm)/Iyy];
 lamda4=[A_Hov(12,:)*xs(:,t-1), (b*Cm)/Izz];

%% root solver to obtain delta_u %
% here  u1=delta_v1^2
%       u2=delta_v2^2
%       u3=delta_v3^2
%       u4=delta_v4^2
% yd=Ac(6,:)*y(t);
% u=(yd-lamda0)/lamda1;
% v= root(u/4);
```

```
ud1(t)=(xd1(t)-lamda1(1))/lamda1(2);
ud2(t)=(xd2(t)-lamda2(1))/lamda2(2);
ud3(t)=(xd3(t)-lamda3(1))/lamda3(2);
ud4(t)=(xd4(t)-lamda4(1))/lamda4(2);
u1(t)=((ud1(t)+ud4(t))/2+ud2(t))/2;
u2(t)=((ud1(t)-ud4(t))/2+ud3(t))/2;
u3(t)=u1(t)-ud2(t);
u4(t)=u2(t)-ud3(t);
% constraint for u should be in range of 0<<u=v^2<<100
% if u1(t)<-10
%     u1(t)=-10;
% elseif u1(t)^2>100
%     u1(t)=10;
% end
% if u2(t)<-10
%     u2(t)=-10;
%  elseif u2(t)^2>100
%     u2(t)=10;
% end
% if u3(t)<-10
%     u3(t)=-10;
%  elseif u3(t)^2>100
%     u3(t)=10;
% end
% if u4(t)<-10
%     u4(t)=-10;
% elseif u4(t)^2>100
%     u4(t)=10;
% end
u(:,t)=[u1(t),u2(t),u3(t),u4(t)]';
%step 3 update state variables from state equations
xs(:,t)=A_Hov*xs(:,t-1)+B_Hov*u(:,t);

end
%step 4 display simulation results
t=1:ns;

%% plot results in the same figure
 figure (3)
 subplot(221),plot(t,u1(t),'r','LineWidth',1.5),box off,grid
on
 ylabel('Control input \Deltav1^2', 'FontSize',
12),xlabel('Time/s', 'FontSize', 12)
 subplot(222),plot(t,u2(t),'r','LineWidth',1.5),box off,grid
on
 ylabel('Control input \Deltav2^2', 'FontSize',
12),xlabel('Time/s', 'FontSize', 12)
```

```
 subplot(223),plot(t,u3(t),'r','LineWidth',1.5),box off,grid
on
 ylabel('Control input \Deltav3^3', 'FontSize',
12),xlabel('Time/s', 'FontSize', 12)
 subplot(224),plot(t,u4(t),'r','LineWidth',1.5),box off,grid
on
 ylabel('Control input \Deltav4^2', 'FontSize',
12),xlabel('Time/s', 'FontSize', 12)
%  subplot(325),plot(t,xs(6,t)),box off
%  ylabel('vz'),xlabel('t')
%  subplot(326),plot(t,xs(5,t)),box off
%  ylabel('vy(t)'),xlabel('t')
%% polt figures
figure (1)
plot(t,xs(1,t),t,xs(3,t),'r:','LineWidth',1.5),grid on
xlabel('Time/s', 'FontSize', 12);
ylabel('Positon', 'FontSize', 12);
legend('Postion \Deltax','Position \Deltaz')
%axis([0 50 -1 1.5]);
figure (2)
plot(t,xs(4,t),t,xs(6,t),'r:','LineWidth',1.5),grid on
xlabel('Time/s', 'FontSize', 12);
ylabel('Positon', 'FontSize', 12);
legend('\Deltavx','\Deltavz')
```

USER MANUAL

Introduction

This program aims to demonstrate and simulate the robust stability study of the designed U-Model based pole placement control systems. MATLAB simulation program can test the robust stability margin of the internal parameter uncertain system. The simulation program procedure includes a U-Model based pole placement control system section, a least squares algorithm function and a robust margin test. Under the determined parameters variation, the robust stability margin of U-Model based pole placement control system can be tested.

Guide

Several steps should be done to run this program and to discover the performance of the U-model based control system design. Here the * simulation is introduced as an example.

- Run the MATLAB software;

- Change the direction point to the **related folder path** and add to the MATLAB path;
- Run the ***.m** and the simulation results will disappear automatically.

References

Ali, S.S.A., Fouad, M.A. and Muhammad, S. (2006) U-model based adaptive tracking scheme for unknown MIMO *bilinear* systems. *1st IEEE Conference on Industrial Electronics and Applications*. Singapore, 24-26 May 2006. pp. 1-5.

Ali, S.S.A., *Fouad*, M.A. and Jamil, M.B. (2008) Feedback analysis of U-model via small gain theorem. *10th WSEAS International Conference on Automatic Control, Modelling and Simulation*. Istanbul, 27-29 May 2008. pp. 63-68.

Alvenes, F. (2012) Satellite attitude control system. Thesis.

Astrom, K.J. and Wittenmark, B. (1995) Adaptive control (2nd ed.), Reading, PA: Addison Wesley.

Agudelo, O.M. and Moor, B.D. (2014) Computergestuurde Regeltechniek exercise session Case study: Quadcopter [online]. Available from: http://homes.esat.kuleuven.be/~maapc/static/files/CACSD/exercises/Session%203/quadcopter_exercise.pdf [Accessed 18 Mar 2017].

Bartolini, G. and Punta, E. (2012) Sliding mode output-feedback stabilization of uncertain nonlinear nonaffine systems, *Automatica,* 48(12), pp. 3106 – 3113.

Bijnens, B., Chu, Q. P., Voorsluijs, G. M. and Mulder, J. A. (2005) Adaptive feedback linearization flight control for a helicopter UAV, *AIAA Guidance, Navigation, and Control Conference and Exhibit*, San Francisco, California.

Bouabdallah, S., Noth A.E. and Siegwart, R. (2004) PID vs LQ control techniques applied to an indoor micro quadrotor'. Int. Conf. Intelligent Robots and Systems (IEEE, Sendai, Japan, 2004), pp. 2451–2456.

Butt, N.R. and Shafiq, M. (2006) Adaptive tracking of nonlinear dynamic plants, using the U-model. *Proceedings IMechE, Part 1: J. Systems and Control Engineering*, 220(16). pp. 473–487.

Calafiore, G. and Dabbene, F. (2002) A probabilistic framework for problems with real structured *uncertainty* in system and control, *Automatica*, 38(8). pp. 1265-1276.

Castillo P., Lozano R. and Dzul A. (2005) Stabilization of a mini rotorcraft having four rotors, *IEEE Control System Magazine*, 25(6), pp. 45-55.

Chaal, H. (2007) A chemical reactor benchmark for adaptive control using U-model and NARMA-L2 techniques. *International Conference on Control*, *Automation and Systems*, pp. 2871–2875.

Chang, W.C., Wang W.J. and Jia H.R. (2011) Radial Basis Functions Neural Network of vary learning rate based *stochastic* U-model. *International Conference on Electrical and Control Engineering (ICECE 2011)*. Yichang, 16-18 September 2011. pp. 278-281.

Chen, C.L. (2012) Linear system theory and design (4th ed.), Oxford: University Press.

Chen, W.Y. and Tu, F.S. (2000) The strict bounded real lemma for linear time-varying systems, *Journal of Mathematical Analysis and Applications*, 244(1), pp. 120-132.

Chong, E.K.P. and Zak, S.H. (2013) An introduction to optimization (4th ed.). New York, NY: Wiley.

Cimen, T. (2010) Systematic and effective design of nonlinear feedback controllers via the state-dependent Riccati equation (SDRE) method. *Annual Reviews in Control*, 34(1), pp. 32-51.

Curtain, R.F. (1993) The strict bounded real lemma in infinite dimensions, *System and Control Letters*, 20, pp. 113-116.

Das, A., Lewis, F. and Subbarao, K. (2009) Back-stepping approach for controlling a quadrotor using Lagrange form dynamics, *Journal of Intelligent and Robotic Systems*, 56, pp. 127–151.

Das, A., Subbarao, K. and Lewis, F. (2009) Dynamic inversion with zero-dynamics stabilisation for quadrotor control, IET *Control Theory Application*, 3, pp. 303-314.

Davis, L.D., Collins Jr, E.G. and Haddad, W. M. (1996) Discrete-time mixed-norm H_2/ H_∞ controller synthesis. *Optimal Control Applications & Methods*, 17, pp. 107-121.

Dorf, R.C. and Bishop, R.H. (2011) Modern Control Systems (12^{th} ed.), Pearson.

Du, W.X., Wu X.L. and *Zhu* Q.M. (2012) Direct design of a U-model-based generalized predictive controller for a class of nonlinear (polynomial) dynamic plants.*Proceedings of the Institution of Mechanical Engineers. Part I: Journal of Systems and Control Engineering* 226 (1). pp. 27-42.

Duarte, M. A. and Narendra, K. S. (1987) Combined Direct and Indirect Adaptive Control of Plants with Relative Degree Greater than One, Technical Report No. 8715, Center for Systems Science, Yale University, New Haven, CT.

ElBsat, M.N. and Yaz, E.E. (2013). Robust and resilient finite-time bounded control of discrete-time uncertain nonlinear systems, Automatica, 49(7), pp. 2292–2296.

Fang C.H. (2002) Stability robustness analysis of uncertain descriptor systems - an LMI approach. *Proceedings of the 41st IEEE Conference on Decision and Control*, vol. 2. pp. 1459-1460. ISSN 0191-2216.

Fang, Z. and Gao, W. (2011) Adaptive integral backstepping control of a micro-quadrotor.

Proceedings of the 2nd International Conference on Intelligent Control and Information Processing (ICICIP), Harbin, 25-28 July 2011, pp. 910-915.

Gahinet, P. and Apkarian, P. (1994) A linear matrix inequality approach to H_∞ control. *Int. J. of Robust and Nonlinear Control*, 4, pp. 421-448.

Gajic, Z. and Qureshi, M.T.J. (2008) *Lyapunov Matrix Equation in System Stability and Control*, Courier Corporation.

Grossimon, P. G. Barbieri, E. and Drakunov, S. (1996) Sliding mode control of an inverted pendulum, *Proceedings of 28th Southeastern Symposium on System Theory*, Baton Rouge, LA, 1996, pp. 248-252.

Haddad, W.M. and Chellaboina, V. (2008) *Nonlinear dynamical* systems *and control: A Lyapunov-based approach*. New Jersey: Princeton University Press.

Hagan, M.T., Demuth, H.B. and Beale M.H. (2002) *Neural network design*. Campus Pub. Service, University of Colorado Bookstore.

Huo, X., Huo, M. and Karimi, H.R. (2014) Attitude stabilization control of a quadrotor UAV by using backstepping approach. *Mathematical Problems in Engineering*, 2014, pp. 1-9.

Isidori, A., (1995) *Nonlinear control systems (3rd ed.)*, Berlin: Springer-Verlag.

Johansen, T.A. (2000) Computation of Lyapunov functions for smooth nonlinear systems using convex optimization, *Automatica*, 36(11), pp. 1617 – 1626.

Kalsi, K., Lian, J.M., Hui, S. and Zak, S.H. (2010) Sliding-mode observers for systems with unknown inputs: A high-gain approach, *Automatica*, 46(2), pp. 347 – 353.

Karimi, H.R. (2006) Dynamic output feedback stabilisation with H_∞ performance for a class of LTPID systems using PPDQ functions, *International Journal of Modelling Identification and Control*, 1(4), pp. 308-315.

Kim, B. W., and Park, B. S. (2016). Robust Control for the Segway with Unknown Control

Coefficient and Model Uncertainties. Sensors (Basel, Switzerland), 16(7), 1000. http://doi.org/10.3390/s16071000.

Khalil, H.K. (2002) *Nonlinear systems* (3rd ed.), Prentice Hall. pp. 30-69.

Koo, T.J. and Sastry S. (1998) Output tracking control design of a helicopter model based on approximate linearization. *Proceedings of 37th Conference Decision and Control (IEEE, Tampa, FL, 1998)*, pp. 3635-3640.

Kravaris, C. and Soroush, M. (1990) Synthesis of multivariable nonlinear controllers by input/output linearization. *AIChE Journal,* 36 (2). pp. 249-264. ISSN 0001-1541.

Lan, Y. and Fei, M.R. (2011) Design of state-feedback controller by pole placement for a coupled set of inverted pendulums, *IEEE 2011 10th International Conference on Electronic Measurement & Instruments*, Chengdu, 2011, pp. 69-73. Doi: 10.1109/ICEMI.2011.6037857.

Langtangen, H.P. (2012) *Computational Partial Differential Equations: Numerical Methods and Diffpack Programming (2nd ed.)*, Springer Berlin Heidelberg.

Lee, B.Y., Lee, H.I. and Tahk, M.J. (2013) Analysis of Adaptive Control Using On-Line Neural Networks for a Quadrotor UAV, *Proceedings of 13th International Conference on Control*, *Automation and Systems* (*ICCAS*), 20-23 October 2013, pp. 1840-1844.

Lee, K.U., Kim, H.S., Park, J.B. and Choi, Y.H. (2012) Hovering control of a quadrotor, *12th International Conference on Control, Automation and Systems (ICCAS)*, 17-21 October 2012, pp. 162-167.

Lee, H.G. and Marcus, S.I. (1987) On input-output linearization of discrete-time nonlinear system. *Systems & Control Letters*, 8(3). pp. 249-259.

Leenaerts, D.M.W. and Bokhoven, W.M.G.V. (1998) *Piecewise linear modelling and analysis*. Boston: Kluwer Academic Publishers, pp. 5-30.

Leontarities, I.J. and Billings. S.A. (1985) Input-output parametric models for nonlinear systems-Part I & II. *International* Journal *of Control*, 41(2). pp. 303-344.

Li, J. and Li, Y. (2011) Dynamic analysis and PID control for a quadrotor. *International Conference on Mechatronics and Automation (ICMA)*, 7-10 August 2011, pp. 573-578.

Li, R.R. and Khalil, H.K. (2012). Nonlinear output regulation with adaptive conditional servocompensator. *Automatica*, 48(10), pp. 2550 – 2559.

Lu, B. (2004) Linear Parameter-Varying Control of an F-16 Aircraft at High Angle of Attack. Ph.D. thesis, North Carolina State University.

Lu, X. and Zhang, W.H. (2008) Mixed H_2/H_∞ output-feedback control for stochastic discrete-time systems. *Control and Decision Conference*, 2-4 July 2008. China. pp. 4792-4796.

Maciejowski, J.M. (1994).Multivariable feedback design. Reading, PA: Addison Wesley.

Madani, T. and Benallegue, A. (2006) Backstepping control for a quadrotor helicopter. *Proceedings of the 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Beijing, 9-15 October 2006, pp. 3255-3260

Mei, S.W., Shen, T.L. and Liu, K.Z. (2003) *Modern Robust Control Applications*, Tsing Hua University Publisher.

Meng, D.Y., Jia, Y.M., Du, J.P. and Zhang, J. (2014) On iterative learning algorithms for the formation control of nonlinear multi-agent systems, *Automatica*, 50(1), pp. 291–295.

Mladenov, V. (2011) Application of Neural Networks for Control of Inverted Pendulum. *Wseas Transactions on Circuits and Systems.* 10(2), pp. 49-58.

Mokhtari, A., Benallegue, A. and Orlov, Y. (2006) Exact linearization and sliding mode observer for a quadrotor unmanned aerial vehicle. *International Journal of Robotics and Automation*, 21, pp. 39-49.

Moreno, J.A. and Osorio, M. (2012) Strict Lyapunov Functions for the Super-Twisting Algorithm. *IEEE Transactions on Automatic Control*, 57(4), pp.1035 – 1040.

Muhammad, S. and Butt, N.R. (2005) U-model based adaptive IMC for nonlinear dynamic plant. Proceedings *of the 10th IEEE International Conference on Emerging Technologies & Factory Automation*. Catania, 19-22 September 2005. 1. pp. 955-959.

Muhammad, S. and Haseebiddon, M. (2005). U-model-based internal model control for non-linear dynamic plants. *Journal of Systems and Control Engineering*, 219(6), pp. 449–458.

Nash, S. (2000) A survey of truncated-Newton methods. *Journal of Computational and Applied Mathematics*, 124, pp. 45-59.

Nejati, A., Shahrokhi, M. and Mehrabani, A. (2012) Comparison between backstepping and input-output linearization techniques for pH process control. *Journal of Process Control*, 22(1). pp. 263-271.

Ogata, K. (2009) *Modern Control Engineering (5th ed.)*, Prentice-Hall.

Oktem, H. (2005) A survey on piecewise-linear models of regulatory dynamical systems. *Nonlinear Analysis* 63 (3). pp. 336-349.

Palunko, I. and Fierro, R. (2011) Adaptive control of a quadrotor with dynamic changes in the center of gravity. *Proceedings of the 18th IFAC World Congress*, Milan, 28 August-2 September 2011, pp. 2626-2631.

Pei, H. and Zhou, Q. (1998) Approximate linearization of nonlinear systems: A neural network approach. *Control Theory & Applications*, 15(1). pp. 31-38.

Peng, Y., Liu, X., Zhu, Q.M., Xu, F.X. and Zhao, D.Y. (2013) The initial robustness analysis of designed U-pole placement control systems, *The Proceedings of 2013 International Conference on Modelling Identification and Control (ICMIC 2013)*. Aug. 31-Sept. 2, 2013, Cairo, Egypt. pp. 253-258.

Petlenkov, E. (2007) NN-ANARX structure based dynamic output feedback linearization for control of nonlinear MIMO systems. P*roceedings of the 15th Mediterranean Conference Control & Automation*, Athens, Greece, 27-29 June 2007, pp. 1132-1137.

Prasad, L.B., Tyagi, B. and Gupta, H. O. (2011) Optimal control of nonlinear inverted pendulum dynamical system with disturbance input using PID controller & LQR, *2011 IEEE International Conference on Control System, Computing and Engineering*, Penang, 2011, pp. 540-545.

Rajak, N. (2015) Stabilization of cart-inverted pendulum using pole-placement method. *International Journal of Advanced Engineering Research and Studies,* 4 (1). 308-312.

Roberge, J.K. (1960) *The mechanical seal*. Bachelor's thesis, Massachusetts Institute ofTechnology.

Roza, A. and Maggiore, M. (2012) Path following controller for a quadrotor helicopter. *Proceedings of the American Control Conference (ACC)*, Montreal, 27-29 June 2012, pp. 4655-4660.

Shen, T. and Zhu, J. (2004) Nonlinear discrete time disturbance system feedback control. *Journal of Zhejiang University*, 38 (4). pp. 425- 428 (In Chinese).

Skogestad, S., and Postlethwaite, I. (2007) *Multivariable Feedback Control*, John Wiley & Sons.

Slotine, J.E. and Li, W. (1991). *Applied nonlinear control*. London: Prentice Hall.

Stevens, B.L. and Lewis, F.L. (2003) *Aircraft control and simulation* (3rd ed.). New Jersey: John Wiley & Sons.

Stipanovie, D.M and Siljakd, D.D. (2001) Robust stability and stabilization of discrete-time nonlinear system: the LMI approach. *International Journal of Control*, 74 (9). pp. 873-879. ISSN 1366-5820.

Taylor, C.J., Chotai, A. and Young, P.C. (2009) Non-linear control by input output state variable feedback pole assignment. International Journal of Control, 82(6), pp. 1029–1044.

Tahir, K. and Muhammad, S. (2006) Adaptive feedforward control of thermal heating process. *IEICE Electronics Express*, 13 (9). pp. 184-190. ISSN 1349-2543

Tan Y., Dai, H.H., Huang, D.Q. and Xu, J.X. (2012) Unified iterative learning control schemes for nonlinear dynamic systems with nonlinear input uncertainties, *Automatica*, 48(12), pp. 3173-3182.

Turner, M. C., Herrmann, G., and Postlethwaite, I. (2004) An introduction to linear matrix inequalities in control, *University of Leicester Department of Engineering Technical Report* on 02-04.

Utkin, V., Guldner, J. and Shi. J. (2009) Sliding Mode Control in Electro-Mechanical Systems (2nd ed.), New York: Taylor & Francis.

Voos, H. (2007) Nonlinear and Neural Network-based Control of a Small Four-Rotor Aerial Robot. In *Proc. of the* IEEEIASME *Int. Conference on Advanced Intelligent Mechatronics*, Zurich, Switzerland, 4-7 Sept. 2007. pp. 1-6.

Voos, H. (2009) Nonlinear control of a quadrotor micro-UAV using feedback-linearization. In *Mechatronics, 2009. ICM 2009. IEEE International Conference on,* pp. 1-6.

Wang, J.J. (2011) Simulation studies of inverted pendulum based on pid controllers. *Simulation Modelling Practice and Theory*. 19(1). pp. 440-449.

Wang, J.J., (2015) Stabilization and tracking control of x-z inverted pendulum based on pid controllers. *Proceedings of the 34th Chinese Control Conference (CCC)*, Hangzhou, China, 2015. pp. 4202-4207.

Wang, L. and Wang, Q.L. (2009) The feedback linearization based on backstepping technique. In *Proc. of the IEEE International Conference on Intelligent Computing and Intelligent Systems*. Shanghai, China, 20-22 Nov. 2009. Pp. 282-286.

Wu, X.L. et al. (2011) U-model based adaptive control for a class of stochastic nonlinear dynamic plants with unknown parameter. *International Journal of Modelling, Identification and Control*, 13 (3). pp. 135-143. ISSN 1746-6172

Xu, J.X. and Tan, Y. (2003) *Linear and nonlinear iterative learning control, lecture notes in control and information sciences*. Berlin: Springer.

Xu, J.X. and Xu, J. (2013) State-constrained iterative learning control for a class of MIMO systems. *IEEE Transactions on Automatic Control*, 58(5), pp. 1322-1327.

Yakoub, Z., Charfeddine, M., Jouili, K. and Braiek, N. B. (2013)A combination of backstepping and the feedback linearization for the controller of inverted pendulum, *10th International Multi-Conferences on Systems, Signals & Devices 2013 (SSD13)*, Hammamet, 2013, pp. 1-6.

Young, P.C. (2000) *Stochastic, dynamic modelling and signal processing: time variable and state dependent parameter estimation, in Nonlinear and Nonstationary Signal Processing*, ed. Fitzgerald, W.J., Cambridge: Cambridge University Press.

Yu, L. (2002) Robust control - Linear matrix inequality techniques. Tsinghua University Publishing House.

Peng, Y. (2014) Novel robust control of damping oscillations in power systems with multi-machine wind farms. PhD thesis, University of the West of England. Available from: http://eprints.uwe.ac.uk/22983

Zemouche, A. and Boutayeb, M. (2013) On LMI conditions to design observers for Lipschitz nonlinear systems, *Automatica*, 49(2), pp. 585 – 591.

Zhai, G., Murao, S., Koyama, N., and Yoshida, M. (2003) Low order H-infinity controller design: An LMI approach, *European Control Conference (ECC)*, pp. 3070-3075.

Zhou, B., Zheng, W.X. and Duan, G.R. (2011) Stability and stabilization of discrete-time periodic linear systems with actuator saturation, *Automatica*, 47(8), pp. 1813-1820.

Zhou, J. and Wen, C. (2008) Adaptive backstepping control of uncertain systems: Nonsmooth nonlinearities, interactions or timevariations. Part of the *Lecture Notes in Control and Information Sciences*, 372, pp. 9-31.

Zhu, Q.M., Ma, Z. and Warwick, K. (1999) Neural network enhanced generalised minimum variance self-tuning controller for nonlinear discrete-time systems. *IEE Proceedings-Control Theory and Applications*, 146 (4). pp. 319-326. ISSN 1350-2379

Zhu, Q.M. and Guo, L.Z. (2002) A pole placement controller for nonlinear dynamic plants.

Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering, 216 (6). pp. 467-476. ISSN 0959-6518.

Zhu, Q.M., Li, S.Y. and Zhao, D.Y., (2014) A universal U-model based control system design, *In Proceedings of the 33rd Chinese Control Conference, 28-30 July, Nanjing, China*, pp. 1839-1844.

Zhu, Q.M., Zhao, D.Y. and Zhang, J.H. (2016) A general U-block model based design procedure for nonlinear polynomial control systems, *International Journal of Systems Science*, 47(14), pp. 3465-3475.

Zulu, A. and John S. (2014) A Review of Control Algorithms for Autonomous Quadrotors, *Open Journal of Applied Science*, 4, pp. 547-556.