DEVELOPMENT OF ADAPTIVE CONTROL METHODOLOGIES AND ALGORITHMS FOR NONLINEAR DYNAMIC SYSTEMS BASED ON U-CONTROL FRAMEWORK

JI QIU

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

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

July 2019

Abstract

Inspired by the U-model based control system design (or called U-control system design), this study aims to develop the U-model based control system design by establishing pulling theorems to deal with non-minimum phase problem and enhance the U-model based control system design for nonlinear dynamic systems by MIT normalised rules and Lyapunov algorithms. This study is clearly novel as the methods we proposed have not been described and defined before.

The study initially proposed a U-model based control system for unstable non-minimum phase system. Pulling theorems are proposed to apply zeros pulling filters and poles pulling filters to pass the unstable non-minimum phase characteristics of the plant model/system. The zeros pulling filters and poles pulling filters are derived from a defined desired minimum phase plant model. The remaining controller design can be any classic control systems or U-model based control system. The difference between classic control systems and U-model based control system for unstable non-minimum phase will be shown in the case studies.

In the second part, the U-model framework is proposed to integrate the direct model reference adaptive control with MIT normalised rules for nonlinear dynamic systems. The U-model based direct model reference adaptive control is defined as an enhanced direct model reference adaptive control expanding the application range from linear system to nonlinear system. The estimated parameter of the nonlinear dynamic system will be placement as the estimated gain of a customised linear virtual plant model with MIT normalised rules. The defined linear virtual plant model is the same form as the reference model. Moreover, the U-model framework is designed for the nonlinear dynamic system within the root inversion.

Thirdly, similar to the structure of the U-model based direct model reference adaptive control with MIT normalised rules, the U-model based direct model reference adaptive control with Lyapunov algorithms proposes a linear virtual plant model as well, estimated and adapted the particular parameters as the estimated gain which of the nonlinear plant model by Lyapunov algorithms. The root inversion such as Newton-Ralphson algorithm provides the simple and concise method to obtain the inversion of the nonlinear system without the estimated gain. The proposed U-model based direct control system design approach is applied to develop the controller for a nonlinear system to implement the linear adaptive control. The computational experiments are presented to demonstrate the effectiveness and efficiency of the proposed U-model based direct model reference adaptive control approach and stabilise with satisfied performance as applying for the linear plant model.

Acknowledgement

Firstly, I would like to express my gratitude to my parents, Dr Qingming Qiu and Prof Yi Li, for their financial support and constant love.

I would like to take this opportunity to express my sincere gratitude to my supervisors, Prof Quanmin Zhu, Dr Mokhtar Nibouche and Prof Yufeng Yao, for their encouragement and guidance during the past years of my studies. I would like to thank them for their supervision throughout the period of this research work.

I gratefully acknowledge Engineering Modelling and Simulation Group (EMSG), UWE support me studentship from 2015-2017.

I also would like to thank Dr Pritesh Narayan, Dr Budi Chandra and Dr Steve Wright, who is critical review my research progresses every year and providing useful suggestions for my research. I also give my thanks to staff and PhD colleagues from the Faculty of Environment and Technology of UWE with whom I share the excellent and comfortable research environment.

Many thanks to my colleagues and friends, Dr Xin Liu, Dr Yizhi Wang, Dr Rui Li, Mr Liwei Xu, Miss Chen Chen for their help and useful discussions on many occasions.

List of publications

Conference presentations

Qiu, J., Zhu, Q.M., and Zhao, D.Y. (2015), Application of Hidden Markov Model to locate soccer robots. IEEE *34th Chinese Control Conference (CCC),* Hangzhou, China, pp. 1716-1721.

Qiu, J., Zhu, G., Zhu, Q.M., Nibouche, M., and Yao, Y., (2016). Comparative study on pole placement controller design with U-Model approach & classical approach. IEEE *35th Chinese Control Conference (CCC),* Chengdu, China, pp. 10436-10441.

Zhu, G., Qiu, J., Zhu, Q., Narayan, P., and Wright, S., (2016). Design of U-PPC-Type II for nonlinear systems. IEEE 35th Chinese Control Conference (CCC), Chengdu, China, pp. 10431-10435.

Qiu, J., Zhu, Q., Nibouche, M., and Yao Y., (2016). U-model enhanced control of nonminimum-phase dynamic systems. IEEE *8th International Conference on Modelling, Identification and Control (ICMIC),* Algiers, Algeria, pp. 648-653.

Qiu, J., Delshad, I.S., Zhu, Q., Nibouche, M., and Yao, Y., (2017). A U-model based controller design for non-minimum phase systems: *Application to Boeing 747 altitude-hold autopilot.* IEEE *9th International Conference on Modelling, Identification and Control (ICMIC),* Kunming, China, pp. 122-127.

Journal paper

Qiu, J., Zhu, Q.M., Nibouche, M. and Yao, Y., *U-model Based Model Reference Adaptive Control of Nonlinear Systems.* (to be submitted)

Zhu, Q.M., Qiu, J., Delshad, I.S., Nibouche, M., and Yao, Y.F., *U-model based control of non-minimum phase systems* (to be submitted)

Zhu, Q.M., Zhang, W.C., S. Mobayen, Yan, H., Qiu, J. and Narayan, P., *U-Control: A model-independent control system design framework* (to be submitted)

Contents

List of Figures

Abbreviations and Nomenclature

Abbreviations:

BIBO - bounded input bounded output

FEM - Finite-element method

IIR - Infinite impulse response

ILC - Iterative learning control

LQG – Linear quadratic Gaussian control

MAC - modified adaptive control

MFC - Model free control

MIMO - Multiple input multiple output

MIT - Massachusetts Institute of Technology

MP - Minimum phase

MRAC - Model reference adaptive control

NARMAX - Non-linear autoregressive moving average with exogenous input

NMP - Non-minimum phase

NSP - New Smith Predictor

PID - Proportional-integral-derivative

RBDO - Reliability-based design optimisation

RDO - Robust design optimisation

RLS - Recursive least squares

SISO - Single input single output

SMC - Sliding mode control

Nomenclatures:

A_c - characteristics equation/closed-loop transfer function

e - error signal

f - function

f_{pd} - denominator polynomial

f_{pn} - numerator polynomial

f_r - rational function

G_c - classic controller

G_p - plant model

G_p^{-1} - inversion of plant model

$\hat{G}_p$ - desired plant model

G_{ppf} - pole pulling filter

G_{zpf} - zero pulling filter

J - cost function

r - reference input

t - time (s)

u - input signal

U - pseudo variable

y - output signal

a - scalar multiplier

f - regression term

j - regressed matrix

Y - time varying parameters

q - associated parametric vector

Chapter 1 Introduction

Control engineering is applied in many engineering disciplines, such as mechanical, electrical and aerospace engineering, and it is used everywhere in our physical lives (Marlin, 1995). In the design of control systems, designers specify the desired system characteristics performance or behaviours first, and then configure or synthesize the controllers and filters for the plant models to best match the requirements of the desired qualities (Phillips and Habor, 2011).

A closed-loop control system, is set of mechanical or electronic devices designed to automatically achieve and maintain the desired output condition by comparing it with the actual condition (Phillips and Habor, 2011). Therefore, the system inputs can be described as a function of the system outputs, in addition, the system outputs are also a function of the system inputs.

For example, a house temperature control system is a simple closed-loop control system, which maintains the temperature of the house at a desired value automatically. To achieve the physical variable, the value of this variable should be measured. The measurement of a variable is called a sensor. In this house temperature control system, the sensor is a thermostat, which shows an appropriate temperature by open an electrical switch and low temperature by closing the switch. Activating a gas furnace will increase the temperature, so that the plant output signal is the actual temperature of the house, and the plant input is the electrical signal that activates the furnace. In the house temperature control system, the input is connected to the output to form the closed loop of the system. Furthermore, in most closed-loop control systems, it is common to connect a third, or more, system into the loop to achieve desired performance or behaviours for the total system. The additional system is usually called a compensator, a controller, or simply a filter.

1.1 Research motivation

Generally, control systems are classified as linear or nonlinear behaviour of the dynamic process. Nowadays, linear control becomes a mature subject with large amount of powerful methodologies and a long history of successful experiment application (Slotine and Li, 1991; Åström and Wittenmark, 2013). Control systems based on these linear methods are usually efficacious in the experiments as:

1) The control system maintains the whole system in a small range of operating variables;
2) Some of the systems are not highly nonlinear and can be treated as approximated linear system;
3) Most of the control theories and designs (Åström and Wittenmark, 2013) are not sensitive to reasonable model errors due to nonlinearities.

Rigorously, assume that the function operates on input $u_1(t)$, $u_2(t)$, $u_1(t)+u_2(t)$, $a\,u(t)$, where a is a scalar multiplier, there are two conditions a linear function f must satisfy (Leigh, 2012):

1) $f(u_1(t))+f(u_2(t))=f(u_1(t)+u_2(t))$
2) $f(a u_1(t))=a\,f(u_1(t))$

However, not every system satisfies all these conditions (Marlin, 1995). With the fact that most real-world systems demonstrate near linear behaviour within a limited operating range and by the significantly enhanced insights available in the linear case (Goodwin, Graebe and Salgado, 2001), the control of nonlinear process/system is then addressed in control engineering.

For example, a public address system is a nonlinear system (Phillips and Habor, 2011). When the speaker’s energy reach the microphone, this system is closed loop and it is a usual case. When the speaker is placed in front of the microphone, the loop gain increases and stability problems generated. This case may occurs such a system becomes unstable. To reduce the loop gain, observer need to cup hands over the microphone. Then the oscillations die out gradually. If a large input imports, the system becomes unstable once more. This stability is a function of the amplitude (or frequency) of the input signals. Such case cannot exist in a linear time-invariant system.

In linear time-invariant system, the system stability can be determined by applying the Routh-Hurwitz criterion, the Nyquist criterion, or other techniques considered in the control system literatures. However, not much statement can be made for nonlinear control systems.

The nonlinear systems have some general characteristics (Phillips and Habor, 2011; Leigh, 2012):

1) Limit cycle. In a nonlinear system, a periodic oscillation is called a limit cycle. Generally, limit cycle is non sinusoidal. In linear time-invariant system, a periodic oscillation is sinusoidal with the amplitude of oscillation a function of both the amplitude of the system excitation and the initial conditions. In contrast, the amplitude of oscillation is independent of the nonlinear system excitation or initial conditions.
2) Subharmonic and harmonic response under a periodic input. A periodic input may exhibit a periodic output that frequency either a subharmonic or a harmonic of the input frequency in nonlinear systems.
3) Jump phenomenon. The nonlinear system input is a sinusoid of constant amplitude supposedly. When the frequency of the input sinusoid is increased, a jump (discontinuity) arises in the amplitude of the response. When the frequency is decreased, a discontinuity happens again at a different frequency. It is also called jump resonance.
4) Multiple equilibrium states. For nonlinear stable systems, there may be an amount of different states that the system can approach as time increases for no system input. These states are called equilibrium states. The state that the system approaches is evaluated by the system initial conditions. The condition depends on the disturbance when disturbed (or perturbed) is illustrated settling to the different states.
5) Poles and zeros, transform methods, root loci, frequency response of nonlinear systems describing function methods, matrix and vector methods, and block-diagram algebra are all inapplicable.
6) System design/synthesis methods for directly control nonlinear system scarcely exist.
7) Difficult to interpret. The behaviour of a nonlinear system is structurally different in different regions of state space

$$\dot{x} = f(x,u) \tag{1.1.1}$$

$$y = g(x) \tag{1.1.2}$$

and

$$x \in X \tag{1.1.3}$$

where state space X is defined for a nonlinear system according to the equation, the n -dimensional state vector x can be visualised as being set available by a nonlinear observer with inputs u and input y and with output $\hat{x}$, where as usual the superscript indicates an estimated value.

Therefore, regarding to different operating regions, the same system may present differently, such as unstable, locally stable, heavily damped or oscillatory. Local and global behaviour are identical with a scaling factor for linear systems. However, it is generally meaningless to discuss of global behaviour for nonlinear systems (Leigh, 2012).

Available approaches to the analysis of nonlinear systems are concerned almost entirely with providing stability information:

Lyapunov's second or direct method: The only approach that does not involve approximation.

However, the designed system will return to equilibrium if perturbed as the information produced by application of the method is of limited value for routine system design. A control loop of guaranteed stability may be synthesised by applying the methods (Leigh, 2012).

Lyapunov's first method: This method depends on local linearisation. It has little or no design applicability (Leigh, 2012).

Describing function method: This method is also depends on linearisation, with sinusoidal analysis proceeds by the expedient of neglecting harmonics generated by the nonlinearities (Aracil and Gordillo, 2004).

Sector bound methods: In this kind of methods, two straight line boundaries cover the nonlinear function and the boundaries are linear functions. These methods are based on the idea of ensuring system stability in the presence of any and every function that can reside in the area that the linear functions covered. There are two famous existing conjectures show the inapplicability of these methods: Aizerman's conjecture and Kalman's conjecture. Harmonics present in the sinusoidal response of the nonlinear system have no counterpart in the linear systems that represent the bounds of the approximating sector (Bragin *et al.*, 2011).

These are the methods which can analyse of nonlinear system model. A **hypothesis** is introduced now:

It is possible to use linear methodologies to directly provide solutions for identification and control of a large class of smooth nonlinear dynamic plant models, and therefore to simplify and generalise nonlinear system design in principle of parsimony.

If the answer is positive, there is no general approach to analyse nonlinear plant properties, such as stability.

What if there are some general linear approaches to design controller to drive nonlinear plant to achieve the same target?

Here the U-model method is then introduced to overcome this issue.

U-model method: This method is a model-independent control system relying on root solver, such as Newton-Ralphson algorithm, to find the controller output (Zhu and Guo, 2002). Remain controller can be any other classic linear approach. This method make the design procedure applicable to linear/nonlinear polynomial/state space model structure and it complements most existing design approaches.

Therefore, despite of difficulties to ever-changing nonlinear system, researchers are still maintain enthusiasm to develop possible design/approach to analysis nonlinear system. Not to mention directly apply linear controller design methodologies to develop nonlinear control systems for both polynomial model expression and state space expression. Under such motivation, a geometric synthesis framework can be established to represent nonlinear dynamic plants and simplify and generalise.

1.2 Research questions

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

1) What is the U-control description and how does U-model based control systems for nonlinear dynamic plants work?
2) What is non-minimum phase and how to deal with non-minimum phase system with U-model based control systems?
3) Is there robustness of U-model based control design for non-minimum phase system?
4) How to establish a generic and efficient framework to accommodate direct model reference adaptive control of nonlinear systems?
5) How to develop U-model based direct model reference adaptive control system?
6) Is the U-model based direct model reference adaptive control system work for both MIT normalised rules and Lyapunov algorithm/function?
7) How to validate and implement these effectiveness of designed U-model control systems into computational simulation and practical application?

1.3 The aims and objectives of the research

With such insight of the U-model based design approach for nonlinear polynomial control system, aim of this PhD research was to develop and analyse U-model control system to be applicable for non-minimum phase system and for direct model reference adaptive control approach. To expand the powerfulness of U-model control system design, it is necessary to establish corresponding theorems for non-minimum phase system, and corresponding structures for direct model reference adaptive control approach.

Overall, according to the hypothesis in last section, this research provides novel concepts, structures, and algorithms in academic development, and shows the applicable for industrial experiments in modelling and control of complex modern systems.

To achieve the aims, the following major objectives have been outlined:

1) To create U-model based control systems with classic pole placement with two different plants.
2) To develop a general U-model non-adaptive control framework where the linear non-adaptive control strategies can be used directly for control of non-linear systems such as non-minimum phase (NMP) systems through pulling theorem. To demonstrate the proper robustness of U-model based design for non-minimum phase system compare to other approaches. Bench test on selected non-minimum phase system model to implement the corresponding U-model based design approach, such as a rotary mechanical system and an altitude hold model of Boeing 747.
3) To develop a general U-model adaptive control framework where the linear adaptive control strategies (methods) can be used directly for control of non-linear models: (i) Lyapunov algorithm; (ii) MIT normalised methods. Bench test on selected nonlinear Hammerstein model to implement the corresponding U-model design approach for demonstrating the performance and applicability of the proposed U-model based direct model reference adaptive control system.

1.4 Contributions

The contributions of this thesis are mainly:

1) Non-minimum phase systems are common in industry applications and many researchers have discussed numerous methods to solve this problem. These methods focus including but not limited to cancellations, which are restricted to the stability of the plant model. The study proposes the pulling theorems, that not only solve the zeros outside the unit circle but also the poles outside the unit circle. In the pulling theorems, a zero pulling filter and a poles pulling filter will be set up according to the customised desired performance/characteristics. These theorems have two advantages, the first one is, even the plant model changed in proper range, the zeros pulling filter and the poles pulling filter does not need to adjust. In other words, the control system has proper robustness. The other one is, when the zeros and/or poles of the non-minimum phase system are all inside the unit two circle, the open loop system maintain stability. These two advantages make the pulling theorems significant.

2) Within pulling filters, the controller may need large power/energy in some cases. With regards to the large power/energy, the control system cost unnecessary financial/equipment supports. The U-model based control system for non-minimum phase systems is emerged to decrease the wasted. Under U-model framework. The root inversion is the core of the design. The novel design of U-model based control system for non-minimum phase systems provides a new and concise methods to deal with non-minimum phase systems for industry applications.
3) The direct model reference adaptive control is proposed to design with U-model framework to extend the applicability range from limited to linear system/plant to nonlinear system/plant. In this structure, a desired virtual linear plant model is established to instead of the original plant model in classic model reference adaptive control and the U-model framework could satisfy of the requirement of the direct model reference adaptive control for nonlinear system/plant.
4) The U-model based direct model reference adaptive control with MIT normalised rules provides the feasibility of applying nonlinear system/plant to direct model reference adaptive control with MIT normalised rules.
5) The U-model based direct model reference adaptive control with Lyapunov algorithms provides the feasibility of applying nonlinear system/plant to direct model reference adaptive control with Lyapunov algorithms when the condition changes in the design of with MIT normalised rules.
6) This study shows case studies through the analytical process and computational experiments to prove the controller output, performance and response of U-model based control systems.
7) Most importantly, this study provides novel methodologies of interdisciplinary research programme and innovative control approaches in control theory.

1.5 Structure of the thesis

This thesis is divided into six chapters. It starts with an introduction to the research in Chapter 1, and ends with conclusions drawn from this research in chapter 6. Chapter 2 provides the research background, methodology and literature review for the research. Chapter 3 presents the design of U-model based control system for non-minimum phase model. Chapters 4 and 5 demonstrate the design of U-model based model reference adaptive control with MIT normalised rules and Lyapunov algorithm respectively.

The outline of the thesis is as follow:

Chapter 1 introduces the research motivation, project aims and objective, and highlights the contributions of the research.

In Chapter 2 the literature review covers an overview of general control systems and description of U-model structure. Also, a comparison between classic pole placement method and U-model based pole placement method is present to show the difference and pros and cons.

Chapter 3 proposes theorems called pulling theorems for non-minimum phase systems. Within the U-model based control system design, it will simply solve the non-minimum phase problems including poles and zeros outside the unit circle and it shows a relatively robustness compared to other classic control methods for non-minimum phase systems.

In Chapter 4, the U-model based model reference adaptive control with MIT normalised rules control system is established with desired virtual plant model. Under the circumstance, the unknown parameters of the nonlinear plant model could be identified with virtual plant model.

In Chapter 5, the U-model based model reference adaptive control with Lyapunov algorithm is proposed to improve the unstable closed-loop problem with the U-model based model reference adaptive control with MIT normalised rules. Also with virtual plant model, a Hammerstein model is demonstrated the simulation results when reference input changes.

Finally, in chapter 6 conclusions are drawn to summarise the study and provided the proposed future research to follow up this study.

A user manual and MATLAB programs/Simulink is shown in Appendix.

Chapter 2 Research background and literature review

2.1 Overviews

Design a control system to reasonably correspond to a requested performance or design specification is the main proposition of control engineering. A control design scheme applicable should be properly presented for justifying the feasibility of model structure, no matter of representing linear systems or non-linear systems. In the linear control systems, there are many available approaches and methods to manage with control problems. Mainly, the approaches are based on two model structures, state-space model (Brogan, 1974) and polynomial model (Åström and Wittenmark, 2013). This chapter presents comparative studies on U-model based control system, which is an enhanced pole placement controller design for linear dynamic plants based on polynomial models, comparing to the classical pole placement approach to demonstrate the difference between them and the superiority of U-model based design approach.

Industrial variability, unidentified environmental conditions and accidental degradation cause uncertainty in physical parameters (Daouk *et al.*, 2015), which obtains variability in measured natural frequencies and damping. Normal analysis techniques, for instance the finite-element method (FEM), are commonly deterministic and not explicitly deliberate the properties of uncertainty. In addition, this kind of techniques regularly adopt average structural parameters, which are estimated (Choi, Canfield and Grandhi, 2007). Consequently, there may be a high degree of inconsistency between prophesied and experimentally obtained parameters. It is desired that substitute analysis methods that measure and decrease the effects of uncertainty are used. Robust design optimisation (RDO) and reliability-based design optimisation (RBDO) have been applied in these decades to address this problem. In these two techniques, an impartial function that weights both response/performance and robustness standards is well-defined that the system accomplishes best performance with negligible compromise to its robustness or probability of failure (Chateauneuf, 2008). A substitute approach is to actively adjust the system by feedback control. By conveying closed-loop poles, the frequency and damping of a system's poles can be changed (Mottershead and Ram, 2006). This is called pole placement.

Pole placement in the area of polynomial and state space is very common and widespread. For single input single output (SISO) plants model, the corresponding output feedback control should be at least of the plant order minus one to achieve arbitrary pole placement (Wang *et al.*, 2008). Arbitrary pole placement is otherwise difficult to achieve if one must apply a low-order output feedback controller for a high-order or time-delay plant.

Based on the results in (Moore, 1975) on Eigen structure assignment in the case of state feedback several solutions to the problem of pole placement by static output feedback have been testified (Srinathkumar, 1978; Champetier and Magni, 1991) and just recently another new technique has been offered (Bachelier, Bosche and Mehdi, 2006; Bachelier and Mehdi, 2008). Most of them rely on the fundamental result (Kimura, 1975; Davison and Wang, 1975), which is also known as Kimura's condition. For the generic system, all closed loop poles can be allocated almost arbitrary if $m+p \geq n+1$ where n, m, p represent the plant order and the number of inputs and outputs, respectively. Moreover, it was shown that an essential and satisfactory condition for arbitrary pole assignment for the generic system is if complex feedback gains are allowable (Wang, 1992), the important result (Wang, 1996) that for real static output feedback is sufficient for generic pole assignability.

The pole placement is a methodology that performs pole placement, by means of active control, using measured receptance (Mottershead and Ram, 2007). The pole placement is beneficial in that experimentally resolute receptance are applied directly and there is no requirement of evaluating the mass, stiffness and damping matrices of the system. Whereas the technique was initially industrialised for single-input systems, it has since been extended to the universal case of multiple input multiple output (MIMO) systems (Ram and Mottershead, 2013) and has been implemented experimentally on a range of changed systems. The receptance method is applied to a lightweight glass-fibre beam with two smart-material sensors and actuators, and to a weighty modular assessment structure by various arrangements of accelerometers and electromechanical shakers (Tehrani, Elliott and Mottershead, 2010).

U-model, a model-independent control approach, with control-oriented expression converted from original linear or non-linear models, is a time-varying parameter polynomial set which covers all existing smooth non-linear discrete time model (Zhu, Zhao and Zhang, 2016). Furthermore, U-model demonstrates an inherent appeal and a straightforward algorithm structure to decrease computational burden in controller design for both linear and nonlinear systems. For instance, since classical pole placement design (Åström and Wittenmark, 2013) constructs the poles associated for the plant model, U-model design only needs to assign the desired poles of characteristic polynomials and steady error, and then obtains controller outputs for any given model or model changes.

By introducing basic idea and properties of pole placement controller design and of with U-model approach, this chapter provides comparison and demonstration of the two approaches in design procedures and computational experiments. As U-model approach is relatively novel and less mentioned, may this study could increase the confidence and assurance for researchers in developing the U-model framework from every aspect. To explain this chapter, some research questions are listed below, which afterward guides the study to provide proper solutions and findings.

1) How to apply pole placement controller design within U-model framework?
2) What are the differences/characteristics of U-model based control system in comparison to classical approach in pole placement controller design?
3) What is the limitation or restriction of U-model based pole placement control systems design?

The rest of the chapter is divided into three sections. Section 1 shows descriptions of the basic three control system framework and provides their pros and cons. In section 2, classical pole placement method and U-model method are introduced the descriptions of design procedures from step by step to obtain the controller, respectively. In section 3, two linear plant models are nominated to demonstrate the design procedures of the two approaches and the corresponding computational experiment simulations are offered.

2.1.1 Model based control system design

To control a linear plant G_p, the controller could be considered as follows:

$$G_c = G_p^{-1} \frac{G}{1- G} \tag{2.1.1}$$

where G_c is classic controller and G is closed loop performance function, specified in advance by designers and/or users. Moreover, G_p could be also modelled as a nonlinear dynamic equation in either the polynomial or state space expression by applying U-model control system.

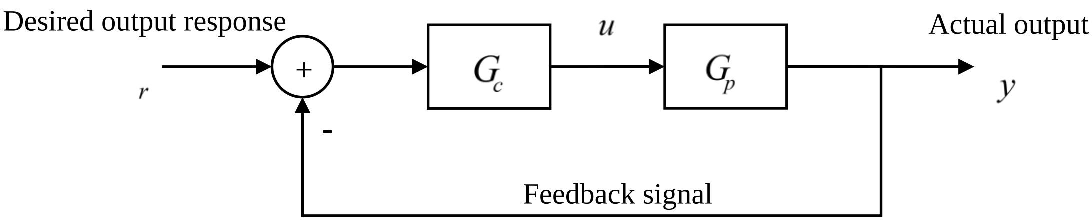


Figure 2.1 Model based closed-loop feedback control system diagram

If the plant model G_p is nonlinear, the controller is then

$$G_c = f(G_p, G) \tag{2.1.2}$$

where $f(*)$ is a mapping function that reflects the plant and closed-loop performance through a certain type of mathematical description (Zhu *et al.*, 2020b).

Remark 2.1: The model of the plant G_p is known as the linear/nonlinear polynomial and state space expressions.

Remark 2.2: A model based closed-loop feedback control system often uses a function of formulated relationship between the actual output and reference input/desired output to control the process.

Remark 2.3: The difference between the actual output and the reference input/desired output is measured and used to control the process. In general, the difference is continually reduced.

Remark 2.4: A model based feedback control system is always a negative feedback control system. Mathematically, the actual output is subtracted from the input and the difference is applied as the input signal to the controller.

Model based control system has been widely applied in academic research and industrial applications (Dorf and Bishop, 2011).

However, the framework features unnecessary repetition in design. Taking a linear plant model as an example, it unnecessarily repeats the calculation of $\frac{G}{1-G}$ if the plant model changed in (2.1.1). It is difficult to design nonlinear plant-based control systems and difficult to specify the transient responses of nonlinear control systems with this framework. The model structure affects the approach needed for the linear/nonlinear and polynomial/state space models, which is a common feature of model-based design frameworks (Zhu *et al.*, 2020b).

2.1.2 Model-free/data-driven control system design

With the developing of control theory and requirement of industry, more researchers focus on model-free/data-driven control system design (Zhu *et al.*, 2020b).

1) PID control by the Ziegler-Nichols approach

Due to high pressures and temperatures with potentially hazardous materials, industry equipment needed reliable process control many decades before digital computers became available (Marlin, 1995). Consequently, the control methods were tailored to the limited computing equipment. Analogue computation becomes the main method of automated computing. The analogue computing principle is the design of a physical system that follows the same equations as the equations desired to be solved (Korn and Theresa, 1972).

The proportional-integral-derivative (PID) algorithm has been successfully used in the process industries since the 1940s and remains the most often used algorithm today (Marlin, 1995).

The PID algorithm is used for SISO systems, which have one controlled variable y and one manipulated variable u.

The PID is often combined by three modes. The proportional mode provides fast response but does not reduce the offset to zero. The integral mode reduces the offset to zero but provides relatively slow feedback compensation. The derivative mode takes action based on the derivative of the controlled variable but has no effect on the offset. With the proper choice of tuning constants, the PID controller could achieve good control performance.

Remark 2.5: The three tuning constants in the PID algorithm interact the dynamic behaviour of the closed-loop system. They have to be adjusted simultaneously.

In 1942, John G. Ziegler and Nathaniel B. Nichols published two important PID controller gain tuning methods intended to achieve a fast closed-loop step response without excessive oscillations and excellent disturbance rejection. These approaches are categorised as Ziegler-Nichols tuning methods.

The first approach is based on closed-loop concepts requiring the computation of the ultimate gain and ultimate period. The second approach is based on open loop concepts relying on response curves.

Remark 2.6: The models do not have to be precisely known, as the Ziegler-Nichols tuning methods are relied on forms of the models of the process. This make PID control by the Ziegler-Nichols tuning approach very practical in industry process control applications.

However, the approach needs numerous experimental work to obtain plant models. Almost all engineering plants/processes and input/output measurements are possible to model in principle, although it is sometimes a difficult task.

2) Iterative learning control (ILC)

Under a repeatable control environment, perfect tracking in a finite time interval is a new class of control tasks (Xu and Tan, 2003). If a control system is able to learn from previous control trials when the task repeats consecutively, no matter succeeded or failed, it is hard to produce the same performance with improvement. Iterative learning control (ILC) is then proposed.

ILC apply the control information of the preceding trial to improve the control performance of the present trial.

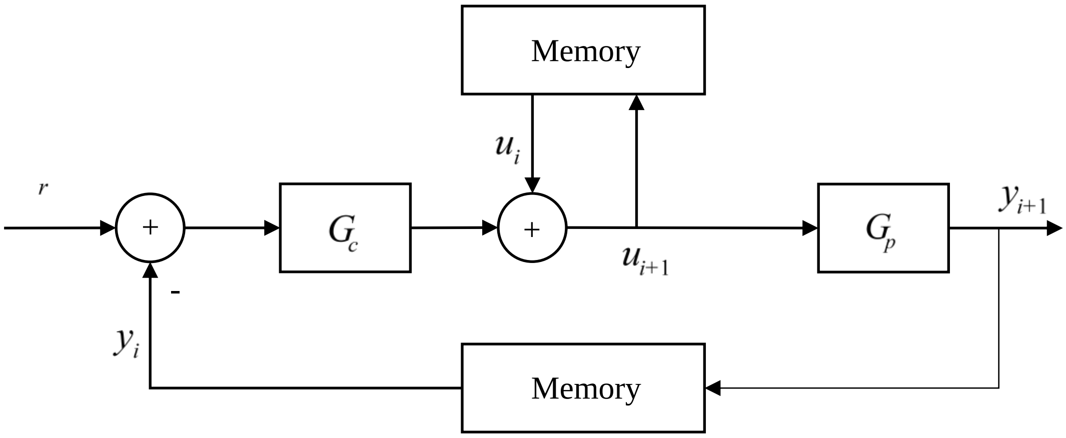


Figure 2.2 ILC control system diagram (Zhu *et al.*, 2020b)

Figure 2.2 shows the memory based ILC schematic diagram where i is the i th control trial, r is target trajectory which is repeated over a fixed time interval, u_i is control signal of the preceding trial and u_{i+1} is the present control. The plant model is deterministic with exactly the same initialization condition.

An appropriate closed-loop controller requires much of the process knowledge. The model of the plant G_p in design is unknown, which required mild conditions. The iterative learning for improving the controller G_c with repeated reference stimulation to achieve $G_c G_p = G_p^{-1} G_p = 1$.

This approach has considered every possibility for integrating past control information into the next round of control design. Same as PID approach, ILC approach requires experimental work to obtain plant models, which cost power almost in every engineering process. Moreover, this approach is only available in a repeatable control environment under strict conditions. It is challenging to control nonlinear dynamic plants with this approach.

3) Model free control (MFC)

Model free control (MFC) is an enhanced PID control (Fliess and Join, 2013). According to

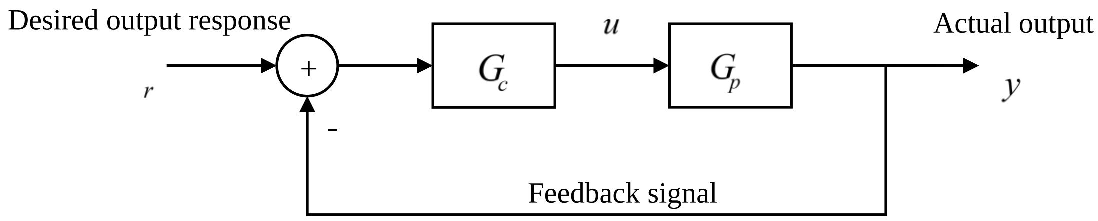


Figure 2.1**Error! Reference source not found.**, u is a single control variable and y is a single output variable. The ultra-local model, which is unknown complex mathematical model is then

$$y^{(v)} = F + au \tag{2.1.3}$$

where $y^{(v)}$ is the derivative of order $v \geq 1$ of y, $a \in \mathbb{R}$ is non-physical constant parameter and F represents the poorly known parts of the plant and the various possible disturbances. F is approximated by a piecewise constant function for estimation. The estimation has the features below:

- It requires a quite short time.
- It is described by algebraic formulae which combine low-pass filters like iterated time integrals.
- According to the new setting of noises via quick fluctuations, it has robustness with respect to quite strong noise corruption.

Then the controller could be designed from

$$u = \frac{F - y^{*} + K_P e + K_I \int e + K_D \dot{e}}{a} \tag{2.1.4}$$

where y^{*} is the reference trajectory, $e = y - y^{*}$ is the tracking error. K_P, K_I, K_D are the usual PID tuning gains. Different from classic PID control, the ultra-local model can be used to approximate complex dynamic plants and improve control performance in this approach.

2.1.3 Model-independent control system design

This framework is novel in literature which propose a parallel design controller and dynamic inversion for the design procedure applicable to linear/nonlinear polynomial/state space model structures (Zhu and Guo, 2002). The main idea of the framework is shown below.

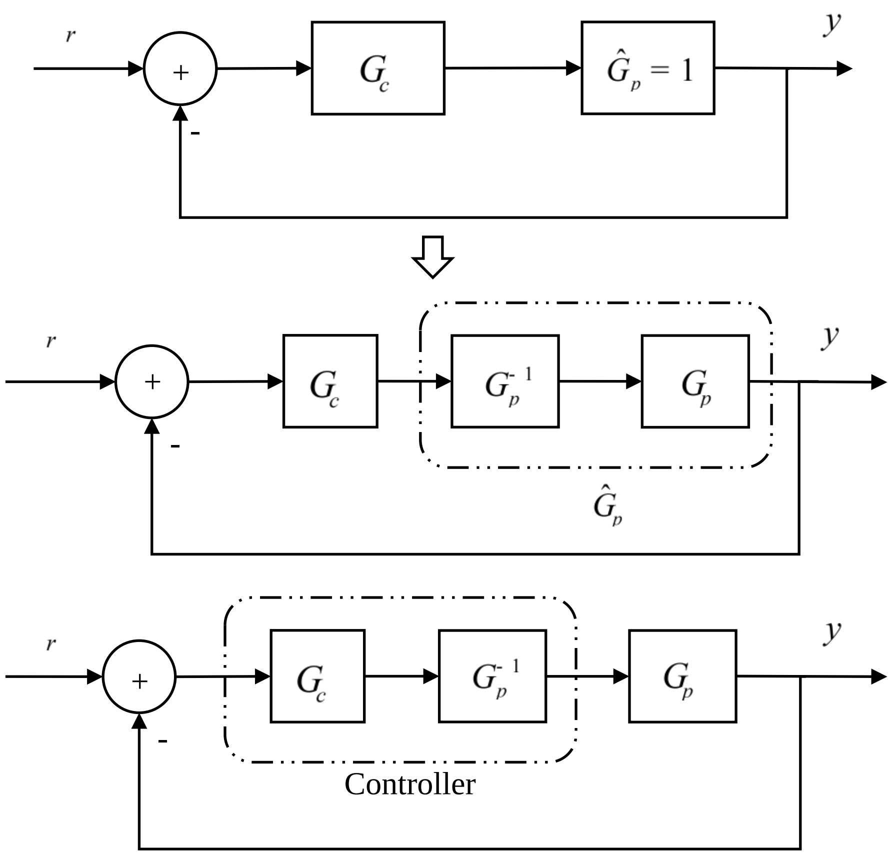


Figure 2.3 Model-independent control system

According to Figure 2.3, the controller is divided into two main parts. One part is general/classic controller G_c and one part is inversion G_p^{-1} of the plant model G_p. With the inversion G_p^{-1}, the plant model G_p is totally independent to the general/classic controller G_c. When the plant model G_p changed, it will not influent the controller G_c. It is neat in design without waste/repetition if the plant model changes. Besides, this approach complements most existing design approaches into controller G_c . U-model framework is a classical model-independent control.

Though model-independent control is sensitive to model uncertainty, the relatively robustness still can be achieve in some particular case, which will demonstrate in chapter 3 U-model structure for non-minimum phase systems.

2.2 Description of U-model structure

Classified as model-independent control, the U-model can be structured based on most of the existing control principles. The main idea of U-model is an explicit input-output relationship at time t with time-varying parameters $\Psi(t)$, that is

$$U\left(y(t), u(t-1), \Psi(t)\right) \tag{2.2.1}$$

where the input $u \in \mathbb{R}$, the output $y \in \mathbb{R}$ and $U = \{U_a \mid a \in u(t-1)\}$ is a vector of the proper dimension and $\Psi = \{\Psi_a \mid a \notin u(t-1)\}$ is a dynamic absorbing vector of the proper dimension that is associated with U (Zhu *et al.*, 2020b).

2.2.1 U-model realisation from classic polynomials model

Consider a SISO dynamic system Σ (Zhu *et al.*, 2020b)

$$\begin{aligned} y(t) &= f_p\left(\Phi(*), \Theta\right) \\ \Phi(*) &= \Phi\left(Y_{t-1}, U_{t-1}\right) \end{aligned} \tag{2.2.2}$$

At $t \in Z^+$ and $\Phi(*) = \left[f_0(*) \cdots f_L(*)\right] \in \mathbb{R}^{L+1}$ where Y_{t-1}, U_{t-1} are expanded from the output in the proper dimensions (Zhu *et al.*, 2020b).

Remark 2.7: This dynamic control system is generally applicable to almost all classic linear plant models if the inversion G_p^{-1} exists. The major idea is to let G_p be a linear model.

Remark 2.8: This dynamic control system is generally applicable to almost all smooth nonlinear plant models if the plant model inverse G_p^{-1} exists. The major idea is to let $G_{ip} = G_P^{-1} G_p = 1$.

Let

$$\varphi(\bullet) = \varphi_i\big(y(t-1),\ldots,y(t-n),u(t-1),\ldots,u(t-L)\big), \forall i = 0 \cdots L \qquad (2.2.3)$$

where L is the plant dynamic order and $\Theta = [\theta_0 \quad \cdots \quad \theta_l] \in \mathbb{R}^{L+1}$ is the associated parametric vector. The function f_p is a polynomial mapping from the input space to the output space (Zhu *et al.*, 2020b).

The vector form of its regression equation is then

$$y(t) = \Phi^T \Theta = \sum_{l=0}^{L} \varphi_l(\bullet)\theta_l \qquad (2.2.4)$$

where the regression terms $\varphi_l(\bullet)$ are the products of past inputs and outputs, such as $y^3(t-1)$, $u^2(t-1)$, $y(t-1)y(t-5)$, and the associated coefficients θ_l are real constants.

The U-model system $\sum_{\text{U-model}}$ is defined as a polynomial/rational system, where the polynomial/rational function

$$p = \{p_a \mid a \in Y, U\} \qquad (2.2.5)$$

is a mapping

$$f_U : u(t-1) \xrightarrow{Y \cup U} y(t) \in \mathbb{R} \qquad (2.2.6)$$

from the input space to the output space.

Typically, this is the general expression of a NARMAX (non-linear autoregressive moving average with exogenous input) model (Billings, 2013; Zhu and Guo, 2002).

To realise a U-model from this classic model, an absorbing rule is set up.

Absorbing rule: Let $m: R^{L+1} \rightarrow R^{M+1}$ be a map from a polynomial f_P to its U-polynomial f_U and suppose that its inverse m^{-1} exists; that is:

$$f_p\left(P(*),\Theta\right) \xrightarrow{m} f_U\left(\Psi(*),U\left(u(t-1)\right)\right) \tag{2.2.7}$$

It has the following properties

- The mapping is injective (one to one).
- The mapping is surjective (onto).
- The mapping is bijective, as it is both injective and surjective.
- The mapping is invertible.
- The mapping does not change any characteristics of both models, such as output response, stability, dynamics and statics.

The absorbing rule is a formation of $\Psi(*)$ from the polynomial f_P: first identify a control basis function $U\left(u(t-1)\right)$ is a function of past input and then absorb all the other associated functions as a coefficient that varies with time.

Therefore, using the absorbing rule, realising the f_U mapping polynomial f_P (2.2.4) gives the following:

$$y(t) = \Psi^T U = \sum_{j=0}^{M} \psi_j(t) U_j\left(u(t-1)\right) \tag{2.2.8}$$

With respect to $u(t-1)$, this function is expanded from the above nonlinear function f_P as a polynomial. Noted that, M is the number of model inputs $u(t-1)$ and the time varying parameter vector $\Psi(*) = \left[\psi_0(t) \;\; \cdots \;\; \psi_M(t)\right] \in \mathbb{R}^{M+1}$ is a function derived from absorbing the other regression terms and the coefficients.

For example, consider the polynomial model described as follow:

$$y(t)=0.2\sin\left(y(t-1)\right)+u(t-1)\exp\left(-y^2(t-1)\right)-0.8y(t-2)u(t-2)u^3(t-1) \quad (2.2.9)$$

To fit the U-model framework, take this polynomial model into U-model realisation expression (2.2.8), absorbing the terms associated with $u(t-1)$ into the vector $\Upsilon(*)$ as

$$y(t)=y_0(t)+y_1(t)U_1\left(u(t-1)\right)+y_2(t)U_2\left(u(t-1)\right) \quad (2.2.10)$$

where

$$\begin{aligned}
\Upsilon_0(t) &= 0.2*\sin\left(y(t-1)\right)\\
\Upsilon_1(t) &= \exp\left(-y^2(t-1)\right)\\
\Upsilon_2(t) &= -0.8y(t-2)u(t-2)\\
U_1\left(u(t-1)\right) &= u(t-1)\\
U_2\left(u(t-1)\right) &= u^3(t-1)
\end{aligned} \quad (2.2.11)$$

2.2.2 U-model realisation from classic rational models

Consider the SISO rational model with a ratio of two polynomials below (Zhu *et al.*, 2020b):

$$y(t)=f_r(\Phi(*),\ \Theta)=\frac{f_{pn}(\Phi_n(*),\ \Theta_n)}{f_{pd}(\Phi_d(*),\ \Theta_d)} \quad (2.2.12)$$

where f_r is a rational function, which is related to the ratio of the numerator polynomial f_{pn} and the denominator polynomial f_{pd} that mapping from the input space to the output space. In terms of parameter estimation and control input design, this rational model is totally nonlinear (Zhu *et al.*, 2015).

Then the U-rational model can be summarised with U-polynomial model conversion as:

$$y(t)=\frac{\Psi_n^T U_n}{\Psi_d^T U_d}=\frac{\sum_{j=0}^{M_n}\psi_{jn}(t)U_{jn}\left(u(t-1)\right)}{\sum_{j=0}^{M_d}\psi_{jd}(t)U_{jd}\left(u(t-1)\right)} \tag{2.2.13}$$

For more convenient to obtain the model inversion by roots solver, the U-rational model (2.2.13) can be written as

$$y(t)\left(\sum_{j=0}^{M_d}\psi_{jd}(t)U_{jd}\left(u(t-1)\right)\right)=\sum_{j=0}^{M_n}\psi_{jn}(t)U_{jn}\left(u(t-1)\right) \tag{2.2.14}$$

To help understanding, consider a rational model for example:

$$y(t)=\frac{0.1y^3(t-1)+\sin(u(t-1))+0.5u^3(t-1)}{1+y^2(t-2)+u^2(t-1)} \tag{2.2.15}$$

The U-model realisation can be written by absorbing the terms associated with $u(t-1)$ into the vector $\Upsilon_n(*)$:

$$\begin{aligned} f_{pn}(u(t-1)) &= \psi_{0n}(t)+\psi_{1n}(t)U_{1n}(u(t-1))+\psi_{2n}(t)U_{2n}(u(t-1)) \\ f_{pd}(u(t-1)) &= \psi_{0d}(t)+\psi_{1d}(t)U_{1d}(u(t-1)) \end{aligned} \tag{2.2.16}$$

where

$$\begin{aligned} &\psi_{0n}(t)=0.1y^3(t-1) \quad \psi_{1n}(t)=1 \quad \psi_{2n}=0.5 \\ &\psi_{0d}(t)=1+y^2(t-2) \quad \psi_{1d}(t)=1 \\ &U_{1n}(u(t-1))=\sin(u(t-1)) \quad U_{2n}(u(t-1))=u^3(t-1) \\ &U_{1d}(u(t-1))=u^2(t-1) \end{aligned} \tag{2.2.17}$$

2.2.3 U realisation from a classic state space mode – multi-layer U-model

Consider a general class of SISO nonlinear processes described by the discrete-time state-space model

$$\begin{aligned} X(t+1) &= F(X(t), u(t)) \\ y(t) &= h(X(t)) \end{aligned} \tag{2.2.18}$$

where $X \in \mathbb{R}^n$ is the state vector, $u \in \mathrm{R}$ is the control input, and $y \in \mathbb{R}$ is the system output at time $t \in \mathbb{Z}^+$. $F \in \mathbb{R}^n$ is a smooth vector function describing the model dynamics and $h \in \mathbb{R}$ is a smooth function relating the system states to the outputs. Assume that the system relative degree r equals to the system order n and has stable zero dynamics and that the state vector X can be obtained through measurement or observation (Zhu *et al.*, 2020b).

Convert the state-space model (2.2.18) into a multi-layer U-model expression below:

$$\begin{cases} x_1(t+1) = \sum_{j=0}^{M_1} \psi_{1j}(t) U_{1j}(x_2(t)) \\ \vdots \\ x_{n-1}(t+1) = \sum_{j=0}^{M_{n-1}} \psi_{(n-1)j}(t) U_{(n-1)j}(x_n(t)) \\ x_n(t+1) = \sum_{j=0}^{M_n} \psi_{nj}(t) U_{nj}(u(t)) \\ y(t) = h(X(t)) \end{cases} \tag{2.2.19}$$

Note that $x_{j+1}(t)$ is the next state variable and $\psi_{ij}(t) = \begin{bmatrix} \psi_{j0}(t) & \cdots & \psi_{jM_j}(t) \end{bmatrix} \in \mathrm{R}^{M_j+1}$, $i \in 1 \cdots n$ is the time-varying parameter vectors are functions absorbing the other state variables. M_j is the number of terms and M_n consists of the terms associated with the control input $u(t)$ and the time-varying vectors $\begin{bmatrix} \psi_{n0}(t) & \cdots & \psi_{nM_n}(t) \end{bmatrix} \in \mathrm{R}^{M_n+1}$ absorb all the states associated as the control vector $\begin{bmatrix} U_{n0}(t) & \cdots & U_{nM_n}(t) \end{bmatrix} \in \mathrm{R}^{M_n+1}$. This is a U-polynomial model consisting of a multi-layer U-model expression in form of state space equations.

For example, consider a nonlinear system described by the following state-space model:

$$\begin{cases} x_1(t+1) = x_2(t) + 0.1x_1(t)x_2(t) \\ x_2(t+1) = -0.1x_1(t) - 0.7x_2(t) + u(t) \\ y(t) = x_1(t) \end{cases} \tag{2.2.20}$$

According to the corresponding multi-layer U-model by using the absorbing rule, the nonlinear system can be written as

$$\begin{aligned} x_1(t+1) &= \psi_{11}(t)U_{11}(x_2(t)) \\ x_2(t+1) &= \psi_{20}(t) + \psi_{21}U_{12}(u(t)) \\ y(t) &= x_1(t) \end{aligned} \tag{2.2.21}$$

where

$$\begin{aligned} &\psi_{11}(t) = 1 + 0.1x_1(t), \psi_{20}(t) = -0.1x_1(t) - 0.7x_2(t), \\ &\psi_{21}(t) = 1, U_{11}(x_2(t)) = x_2(t), U_{12}(u(t)) = u(t) \end{aligned} \tag{2.2.22}$$

2.2.4 Inversion of U-model

In most of U-model literatures (Zhu, Zhao and Zhang, 2016; Wu *et al.*, 2011; Zhu and Guo, 2002; Zhu *et al.*, 2015), the Newton-Raphson algorithm is applied to determine the root of the U-model, which means, for a U-model root solver, the Newton-Raphson algorithm (Gerald, 2004) can be used to find the controller output $u(t-1)$. The formulation of the controller output (Zhu *et al.*, 2020b) can be estimated as a recursive computation as

$$u_{k+1}(t-1) = u_k(t-1) - \left. \frac{y(t) - \sum_{j=0}^{M} \lambda_j(t) u_k^j(t-1)}{\dfrac{d\left[\sum_{j=0}^{M} \lambda_j(t) u_j(t-1)\right]}{du(t-1)}} \right|_{u_k(t-1) = u_k^j(t-1)} \tag{2.2.23}$$

where k is the iteration index. The $(k+1)$th iteration is obtained from the k th iteration, $k > 0$. There are also root solving algorithms (Chong and Zak, 2011) could determine the inversion of U-model. Additionally, in simulation work by MATLAB, there is a MATLAB function called *roots*, can be used to solve accurate roots of the plant model.

2.2.5 U-model Controller design

In regard to Figure 2.3, a model-independent control system design with a single input $u \in \mathbb{R}^1$ and single output $y \in \mathbb{R}^1$ is structured as the triplet (Zhu *et al.*, 2020b):

$$\sum = \left(F_{lfbc} \quad G_c \quad G_{ip} \right) \tag{2.2.24}$$

This triplet can be expressed as a model-independent control system design framework. In this triplet, F_{lfbc} is the linear feedback control framework, G_c is a linear invariant controller and G_{ip} is constant unit plant, which

$$\begin{aligned} &G_c : y \to u \\ &G_{ip} = G_P^{-1} G_p = 1 : u \to y \end{aligned} \tag{2.2.25}$$

where G_p is the real plant, which can be modelled from linear/nonlinear, polynomial/state space dynamic equations, and G_p^{-1} is the plant model inversion.

Consider the SISO plant model G_p in the form of the state space equations. According to the Lipschitz continuity, without loss of generality, the plant model G_p can be described as

$$G_p(X, F, h) : \begin{aligned} X(t+1) &= F(X(t), u(t)) \\ y(t) &= h(X(t)) \end{aligned} \tag{2.2.26}$$

where $X \in \mathrm{R}^n$ is the state vector, F is the mapping function from the input space to the state space, h is the mapping function and from the state space to the output space, $u \in \mathrm{R}$ is the control input, and $y \in \mathbb{R}$ is the system output at time $t \in \mathbb{Z}^{+}$.

The inversion G_p^{-1} of the plant model G_p is implemented by u are diffeomorphisms and globally uniform Lipschitz with Lipschitz coefficients g_1, g_2 in $\mathbb{R}^n$:

$$\begin{aligned} \left\| G(x_1) - G(x_2) \right\| &\le \gamma_1 G \left\| x_1 - x_2 \right\|, \quad \forall x_1, x_2 \in \mathbb{R}^n \\ \left\| G^{-1}(x_1) - G^{-1}(x_2) \right\| &\le \gamma_1 G^{-1} \left\| x_1 - x_2 \right\|, \quad \forall x_1, x_2 \in \mathbb{R}^n \end{aligned} \tag{2.2.27}$$

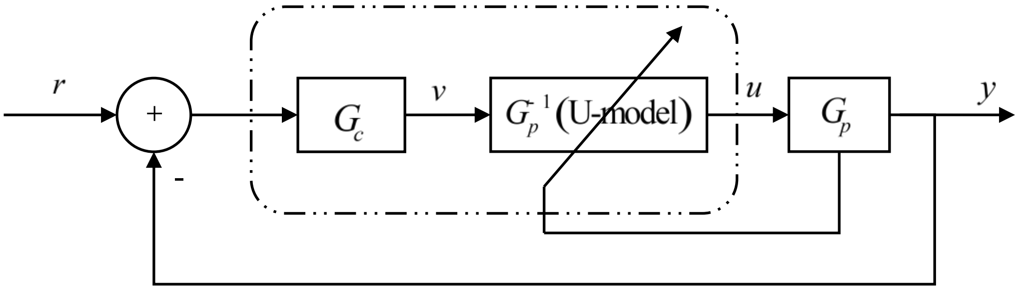


Figure 2.4 U-model control system

Remark 2.9: The U-model control system is a one-off design as the controller design contains two parts and they are independent to each other. Therefore, the design of the control system involves re-computing the inversion/plant inverse G_p^{-1} whenever the plant model G_p varies/changes. For the controller design, two separate procedures conducted in parallel:

- The linear invariant controller G_c is designed within a linear feedback structure F_{lfbc}.
- $G_{ip} = G_P^{-1} G_p = 1$. For many types of structured models, this controller design is usually feasible as long as their inverse exists.

Remark 2.10: U-control requires comparatively less determination for dynamic inversion. Noted that the dynamic inversion is involved in almost all control system designs in one way or another. For instance, in ILC, the final controller is precisely the inverse of the process model. Similarly, the iterative learning control collapses the process model into a unit constant as well as U-control. Therefore, compare to other many model based control system design approaches, U-control requires comparatively less determination for dynamic inversion. This feature could be clarified through an inverse function Ψ : for U-control it is a function of $\Psi(G_p)$, but in the other control approaches, it is at least a function of $\Psi(G_{c1}, G_p)$, which is typical in model based classic linear feedback control system design.

Remark 2.11: Based on the above proposals, this model-independent control system design provides a supplementary platform that can yield/accommodate almost all existing classic design approaches and eliminate the requirement for the model structure in controller design. Furthermore, it features the generalisation of one-off linear invariant controller design in a closed loop form of almost all types of plant models.

Remark 2.12: Generally, nonlinearity is essential to dynamical systems. The transient performance research/study of nonlinear control systems have received important cogitation and investigating their response/performance by linear control approaches is core idea (Chen *et al.*, 2003). U-control is proved that it is a promising procedure.

Remark 2.12: As U-control is basically based on the statement $G_P^{-1}G_p = 1$, it is critical to reflect the robustness to the results of the control system in the case of uncertainty, which is very common in industrial practical systems.

In general, there are two parallel routine in the U-control system design, as shown in Figure 2.4.

1) Form an appropriate linear feedback control structure. The controller, shown in the dashed line block, consists of two controller blocks/functions, an invariant controller G_c and a dynamic inverted controller G_p^{-1}. The plant model is G_p. In addition, nominate an invariant controller G_c by the characteristics/performance of the closed loop control system. By letting $G_P^{-1}G_p = 1$ and determining the desired closed loop transfer function G, $G_c = \frac{G}{1-G}$ is then obtained. Therefore, the invariant controller output $v(t)$ is the desired output as customised, while the combination of the plant model and the inversion is a unit constant. Noted that it is under a closed loop control system instead of an open loop system.
2) Determine a U-control of the plant model G_p to figure the dynamic root invertion G_p^{-1}. Allocate $y(t) = v(t)$ to determine the roots of the plant model to obtain the controller output $u(t-1)$. Assume that the plant model is a bounded-input/bounded-output (BIBO) model and it is stable. The inverse of G_p exists as well. To regulate the control

input $u(t-1)$ is to figure out the inversion by rooting the plant model by U-model expression.

2.3 Comparison of U-model and classic approach

To begin a foundation for the research, the focal concepts and algorithms of classical pole placement design and of U-model based pole placement design are described in this section.

2.3.1 Pole placement

It is assumed that the a SISO system (Åström, Hagander and Sternby, 1984) can be defined by

$$A(q)y(t) = B(q)\left(y_d\left(u(t)\right) + v(t)\right) \qquad (2.3.1)$$

where A and B are polynomials of the forward shift operator q, $y(t)$ is the plant output, $y_d(t)$ is the desired output, $u(t)$ is the control input, and $v(t)$ is a disturbance/error.

The polynomial A and B have the degrees functions $\deg A = n$ and $\deg B = \deg A - d_0$, where the parameter d_0, which is called the pole excess, representing the integer part of the ratio of time delay and sampling period. Mathematically, d_0 indicates the number of poles minus the number of zeros. In this study, express the process model in the delay operator q^{-1}. This can be overcome by presenting the reciprocal polynomial

$$A^*\left(q^{-1}\right) = q^{-n}A(q) \qquad (2.3.2)$$

where $n = \deg A$. The model can then be written as

$$A^*\left(q^{-1}\right)y(t) = B^*\left(q^{-1}\right)\left(u\left(t-d_0\right) + v\left(t-d_0\right)\right) \qquad (2.3.3)$$

where

$$A^*\left(q^{-1}\right) = 1 + a_1 q^{-1} + \ldots + a_n q^{-n}$$
$$B^*\left(q^{-1}\right) = 1 + b_1 q^{-1} + \ldots + b_n q^{-m}$$

with $m = n - d_0$. Notice that since n was defined as the degree of the system, then $n \geq m + d_0$, and trailing coefficients of A^* may thus be zero.

When the system is dealt with discrete time, the design method is purely algebraic. The continuous systems simultaneously is written as

$$Ay(t) = B\left(y_d(t) + v(t)\right) \tag{2.3.4}$$

It is assumed that A and B are relatively prime. Also, A is monic that the coefficient of the highest power in A is unity.

A general linear controller is described as

$$Ry_d(t) = Tr(t) - Sy(t) \tag{2.3.5}$$

where R, T and S are polynomials, and $r(t)$ is the reference input.

To determine the controller, controller (2.3.5) can be describe as

$$y_d(t) = \frac{T}{R} r(t) - \frac{S}{R} y(t) \tag{2.3.6}$$

Controller (2.3.6) is structured as Figure 2.5.

This control law represents a negative feedback with the transfer operator S/R and a feedforward with the transfer operator T/R. This is the general pole placement controller design where T/R and S/R are the poles should be specified.

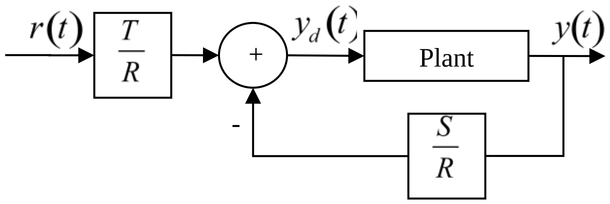


Figure 2.5 A general pole placement design controller

Taking system (2.3.4) and controller (2.3.5) to obtain the plant output $y(t)$:

$$
\begin{aligned}
y(t) &= \frac{BT}{AR+BS} r(t) + \frac{BR}{AR+BS} v(t) \\
y_d(t) &= \frac{AT}{AR+BS} r(t) + \frac{BS}{AR+BS} v(t)
\end{aligned}
\tag{2.3.7}
$$

The close-loop characteristic polynomial is thus become

$$AR + BS = A_c \tag{2.3.8}$$

Expression (2.3.8) is solved by Diophantine equation.

Only R and S can be determined by Diophantine equation. Other conditions must be introduced to also determine the polynomial T in the controller (2.3.5). The response from the command signal u_c is required to the output be described by the dynamics which is desired closed-loop system

$$A_m y_m(t) = B_m u_c(t) \tag{2.3.9}$$

It then follows from output (2.3.7) that the condition below must be held:

$$\frac{BT}{AR+BS} = \frac{BT}{A_c} = \frac{B_m}{A_m} \tag{2.3.10}$$

This model following condition indicates that the response of the close-loop system to command signals is as specified by the model (2.3.9). Whether model-following can be achieved depends on the model, the system, and the command signal. If it is possible to make the error equal to zero for all command signals, then perfect model-following is achieved.

Condition (2.3.10) implies that there are cancellations of factors of BT and A_c. Factor the B polynomial as

$$B = B^{+}B^{-} \tag{2.3.11}$$

where B^{+} is a monic polynomial whose zeros are stable and so well damped that they can be cancelled by the controller and B^{-} corresponds to unstable or poorly damped factors that cannot be cancelled. It thus follows that B^{-} must be a factor of B_m . Hence

$$B_m = B^{-}B_m^{'} \tag{2.3.12}$$

Since B^{-} is cancelled, it must be a factor of A_c. Furthermore, it follows from condition (2.3.10) that A_m must also be a factor of A_c. The close-loop characteristic polynomial thus has the from

$$A_c = A_0 A_m B^{+} \tag{2.3.13}$$

Since B^{+} is a factor of B and A_c, it follows from expression (2.3.8) that it also divides R. Hence

$$R = R^{'}B^{+} \tag{2.3.14}$$

And the Diophantine expression (2.3.8) reduces to

$$AR^{'} + B^{-}S = A_0 A_m A_c \tag{2.3.15}$$

Introducing equation (2.3.12), (2.3.13) and (2.3.14) into equation (2.3.11) gives

$$T = A_0 B_m^{'} \tag{2.3.16}$$

Consider a discrete-time plant process described by the transfer function

$$\frac{B}{A} = \frac{b_0 z + b_1}{z^2 + a_1 z + a_2} \tag{2.3.17}$$

Let the desired close-loop system be

$$A_c = \frac{B_m}{A_m} = \frac{b_{m0} z + b_{m2}}{z^2 + a_{m1} z + a_{m2}} \tag{2.3.18}$$

The controller is thus characterized by the polynomials

$$R = z + \frac{b_1}{b_0} \tag{2.3.19}$$

$$S = \frac{a_{m1} + a_1}{b_0} z + \frac{a_{m2} + a_2}{b_0} \tag{2.3.20}$$

$$T = \frac{b_{m0}}{b} z \tag{2.3.21}$$

Process above shows a simple discrete-time example how to establish a controller by pole placement. Since the design method is purely algebraic, there is no difference between discrete-time and continuous-time controller.

2.3.2 U-Model based pole placement

The U-model is a time-varying parameter polynomial which can present smooth non-linear object. Under a U mapping, the U-model output $u(t-1)$ oriented polynomial is shown below,

$$\begin{gathered} y(t) = f\left(\gamma(*), U(t-1)\right) \\ \gamma(*) = \gamma\left(y(t-1), \ldots, y(t-n), u(t-2), \ldots, u(t-n), \Theta\right) \\ U(t-1) = \left(const\ u(t-1)\, u^2(t-1) \ldots u^M(t-1)\right) \end{gathered} \tag{2.3.22}$$

where $U(t-1)$ is assumed that it is equal to $y_d(t)$.

Correspondingly, its regression equation is given as

$$y(t) = \sum_{j=0}^{M} \lambda_j(t) u^j(t-1) \qquad (2.3.23)$$

where M is the degree of model input (controller output) $u(t-1)$, the time varying parameter vector $\lambda(t) = [\lambda_0(t) \cdots \lambda_M(t)] \in R^{M+1}$ is a function of past inputs and output $(u(t-2),\ldots u(t-n), y(t-1),\ldots, y(t-n))$, and the parameters $(\theta_0 \cdots \theta_L)$.

To work out $u(t-1)$, root-solving algorithm is adopted to resolve as

$$u(t-1) = \Psi\left[y_d(t) - \sum_{j=0}^{M} \lambda_j(t) u^j(t-1) = 0\right] \qquad (2.3.24)$$

where Ψ is a root-solving algorithm, such as Newton-Raphson algorithm (Chong and Zak, 2011). A detailed analysis on the root solving issues has been presented (Zhu and Guo, 2002).

For a linear plant model,

$$u(t-1) = \frac{y_d(t) - \lambda_0(t)}{\lambda_1(t)} \qquad (2.3.25)$$

where $\lambda_1(t)$ is the coefficient associated with $u(t-1)$ (for linear time invariant models, $\lambda_1(t)$ is a constant). $\lambda_2(t)$ (nonzero) is the summation of the rest of the terms in the linear model (Zhu and Guo, 2002).

The U-model is defined as a general linearized model from the nonlinear polynomial model through the conversion to the U-model (2.3.23) and then assigned with required poles through a linear feedback control algorithm (Zhu, Zhao and Zhang, 2015).

There is an example for expediently understanding the polynomial to the U-model conversion.

The polynomial model is

$$\begin{aligned} y(t) &= 0.1y(t-1)y(t-2) - 0.5y(t-1)u^2(t-1) \\ &+ 0.8u(t-1)u(t-2) \end{aligned} \tag{2.3.26}$$

And the U-model can be expressed as,

$$y(t) = l_0(t) + l_1(t)u(t-1) + l_2(t)u^2(t-1) \tag{2.3.27}$$

where $l_0(t) = 0.1y(t-1)y(t-2)$, $l_1(t) = 0.8u(t-2)$, and $l_2(t) = -0.5y(t-1)$.

It is worthwhile to mention that for linear systems, the polynomial has only two main factors: l_0 and l_1.

Recall the general linear controller (2.3.5):

$$Ry_d(t) = Tr(t) - Sy(t) \tag{2.3.28}$$

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

$$y_d(t) = \frac{T}{R+S} r(t) = \frac{T}{A_c} r(t) \tag{2.3.29}$$

where polynomial A_c is the close-loop characteristic equation and specified in advance, that is

$$R + S = A_c \tag{2.3.30}$$

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

$$T = A_c(1) \tag{2.3.31}$$

The key idea of the design is to specify the desired close-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 desired, the controller output $u(t-1)$ can be determined by resolving one of the root of the U-model (2.3.23), which the algorithm (2.3.24) and (2.3.25) has present.

The whole framework of U-model in using linear pole placement approaches to design control systems with linear polynomial plant models is shown in Figure 2.6.

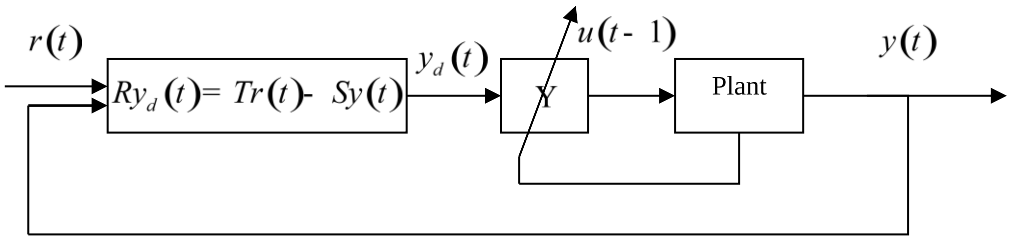


Figure 2.6 A U-model based pole placement control system

2.3.3 Case Studies: comparison of classical based pole placement and U-model based pole placement

Preparation

Consider two linear dynamic plant models for the computational experiments for two examples (Qiu *et al.*, 2016a).

Plant model 1:
$$y(t) = 0.5y(t-1) + 0.8y(t-2) + u(t-1) + 0.4u(t-2) \tag{2.3.32}$$

Plant model 2:
$$G(s) = \frac{0.5832s + 7.2610}{s^2 + 0.4463s + 3.8730} \tag{2.3.33}$$

Specify the desired close-loop characteristic equation with

$$A_c = \frac{0.1761z}{z^2 - 1.3205z + 0.4966} \tag{2.3.34}$$

The control systems of two plants will be designed with both classical approach and U-model approach. Therefore provide computational comparisons.

1) Classical pole placement control

Solution to plant model 1

The first step is to convert the linear dynamic plant (2.3.32) into the same formula as formula (2.3.17) using *z*-transform as

$$\frac{Y(z)}{U(z)} = \frac{z + 0.4}{z^2 - 0.5z - 0.8} \tag{2.3.35}$$

And then observe plant (2.3.35). From plant (2.3.35), $\deg A = 2$ and $\deg B = 1$ are easily found out. The sampled data system has a zero in -0.4 and poles in 1.1787 and 0.6787.

From formula (2.3.17) and plant (2.3.35), $b_0 = 1$, $b_1 = 0.4$, $a_1 = 0.5$ and $a_2 = -0.8$ is determined.

From formula (2.3.18) and desired characteristic equation (2.3.34), $b_{m0} = 0.1761$, $a_{m1} = -1.3205$ and $a_{m2} = 0.4966$ is determined.

As shown in formula (2.3.19), (2.3.20) and (2.3.21), R, T and S can be figured out:

$$\begin{aligned} R &= z + \frac{b_1}{b_0} = z + 0.4 \\ S &= \frac{a_{m1} + a_1}{b_0} z + \frac{a_{m2} + a_2}{b_0} = 0.8205z + 1.2966 \\ T &= \frac{b_{m0}}{b_0} z = 0.1761z \end{aligned} \tag{2.3.36}$$

Therefore the whole controller can be determined by placing T/R and S/R as shown in Figure 2.6.

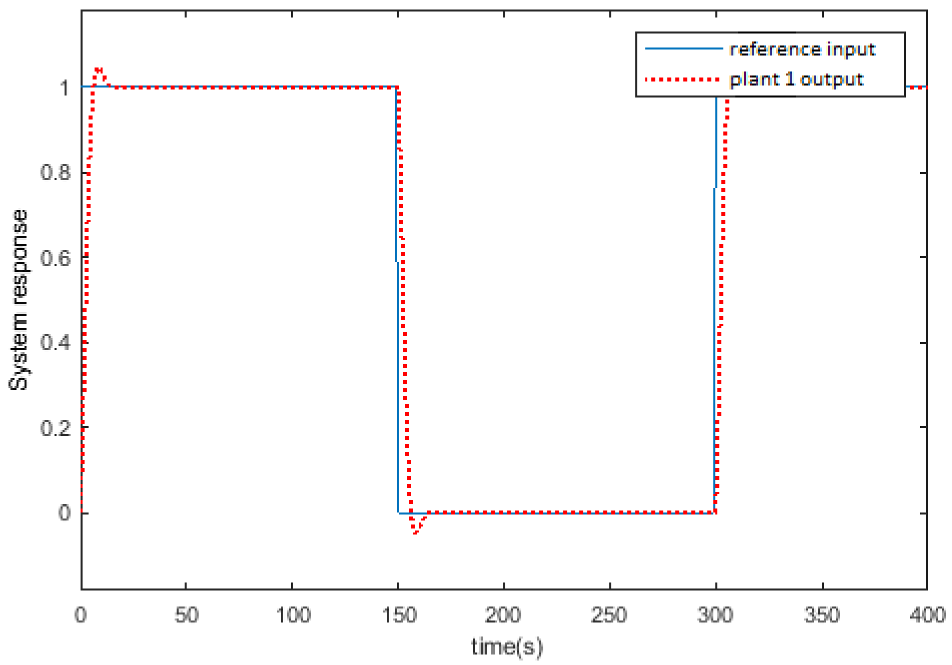


Figure 2.7 System response for plant model 1 by classical pole placement

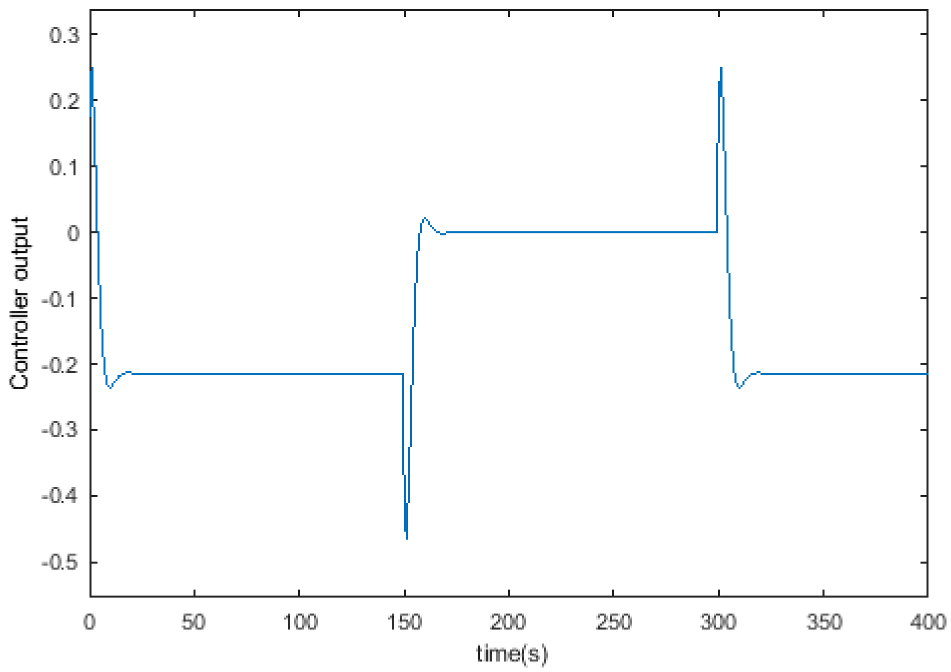


Figure 2.8 Controller output for plant 1 by classical pole placement

Solution to plant model 2

The plant (2.3.33) is a continuous-time process. This can be regarded as a normalized model for a motor. The pulse transfer operator the sampling period $h = 0.5\text{s}$ is

$$G(z) = \frac{z + 0.5}{z^2 - z + 0.8} \tag{2.3.37}$$

From plant (2.3.37), $\deg A = 2$ and $\deg B = 1$ are found out. The sampled data system has a zero in $z = -0.5$ and poles in $z = 0.5 + 1.4832j$ and $z = 0.5 - 1.4832j$.

From formula (2.3.17) and plant (2.3.37), $b_0 = 1$, $b_1 = 0.4$, $a_1 = 1$ and $a_2 = 0.8$ is determined.

From formula (2.3.18) and desired characteristic equation (2.3.34), $b_{m0} = 0.1761$, $a_{m1} = -1.3205$ and $a_{m2} = 0.4966$ is determined.

As the same step in solution to Plant 1 by classical pole placement control, R, T and S should be figured out from formula (2.3.19), (2.3.20) and (2.3.21) again:

$$\begin{aligned} R &= z + \frac{b_1}{b_0} = z + 0.5 \\ S &= \frac{a_{m1} + a_1}{b_0} z + \frac{a_{m2} + a_2}{b_0} = 0.3205z - 0.3034 \\ T &= \frac{b_{m0}}{b_0} z = 0.1761z \end{aligned} \tag{2.3.38}$$

Therefore the whole controller can be determined by placing T/R and S/R as shown in Figure 2.6

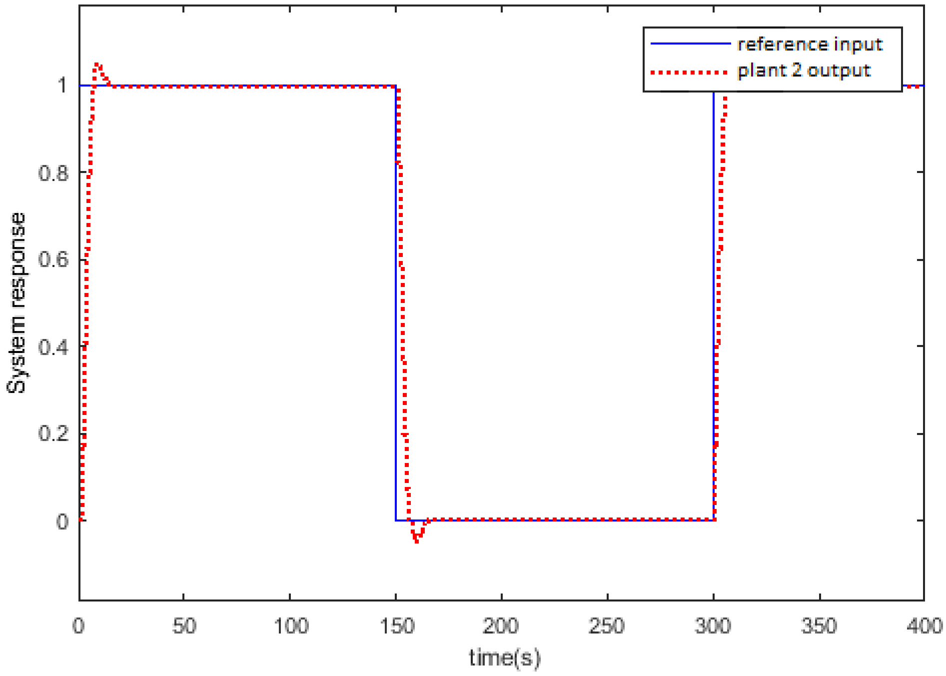


Figure 2.9 System response for plant model 2 by classical pole placement

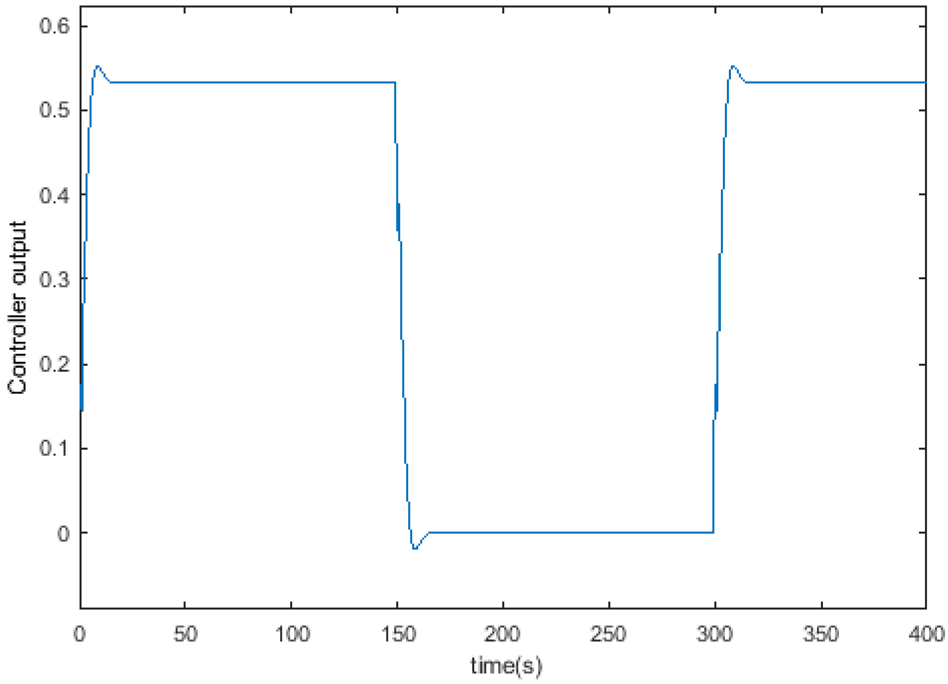


Figure 2.10 Controller output for plant model 2 by classical pole placement

2) U-model based pole placement control

Solution to plant model 1

To achieve zero steady state error, specify T by making the close-loop characteristic equation as

$$T = A_c(1) = 0.1761 \tag{2.3.39}$$

For the polynomials R and S, specify

$$R = z^2 + r_1 z + r_2$$
$$S = s_0 z + s_1 \tag{2.3.40}$$

Substituting the specifications of equation (2.3.34) and (2.3.40) into the Diophantine equation of (2.3.30), the coefficients in polynomials R and S can be expressed by

$$r_2 + r_1 = 0.4966$$
$$r_1 + s_0 = -1.3205 \qquad (2.3.41)$$

To guarantee the computation convergence of the sequence $U(t)$, i.e. to keep the difference equation with stable dynamics, let $r_1 = -0.9$ and $r_2 = 0.009$. This assignment corresponds to the characteristic equation of $U(t)$ as $(z-0.89)(z-0.01)=0$. Then the coefficients in polynomial S can be determined from the Diophantine equation of (2.3.40) as

$$s_0 = -0.4205$$
$$s_1 = 0.4876 \qquad (2.3.42)$$

Substituting the coefficients of the polynomials R and S into the controller of (2.3.5) gives rise to

$$\begin{aligned} y(t+1) = {} & 0.9y_d(t) - 0.009y_d(t-1) \\ & + 0.1761w(t-1) + 0.4205y(t) \\ & - 0.4876y(t-1) \end{aligned} \qquad (2.3.43)$$

Therefore the controller output $u(t)$ can be determined by solving the root in terms of equation

$$u_{k+1}(t-1) = u_k(t-1) - \left. \frac{\sum_{j=0}^{M} \hat{\lambda}_j(t) u^j(t-1) - U(t)}{d\left[\sum_{j=0}^{M} \hat{\lambda}_j(t) u^j(t-1)\right] \Big/ du(t-1)} \right|_{u^j(t-1)=u_k^j(t-1)} \qquad (2.3.44)$$

The corresponding control-oriented model of is obtained from formula (2.3.25):

$$y(t) = \lambda_0(t) + \lambda_1(t)u(t-1) \qquad (2.3.45)$$

where

$$\lambda_0(t) = 0.5y(t-1) + 0.8y(t-2) + 0.4u(t-2)$$

$$l_1(t) = 1 \tag{2.3.46}$$

Substituting $y(t)$ in equation (2.3.46) into (2.3.43), the output response of the designed U-model with assigned poles and steady state property is shown in Figure 2.11, and the pole placement controller output is shown in Figure 2.12 (Zhu and Guo, 2002).

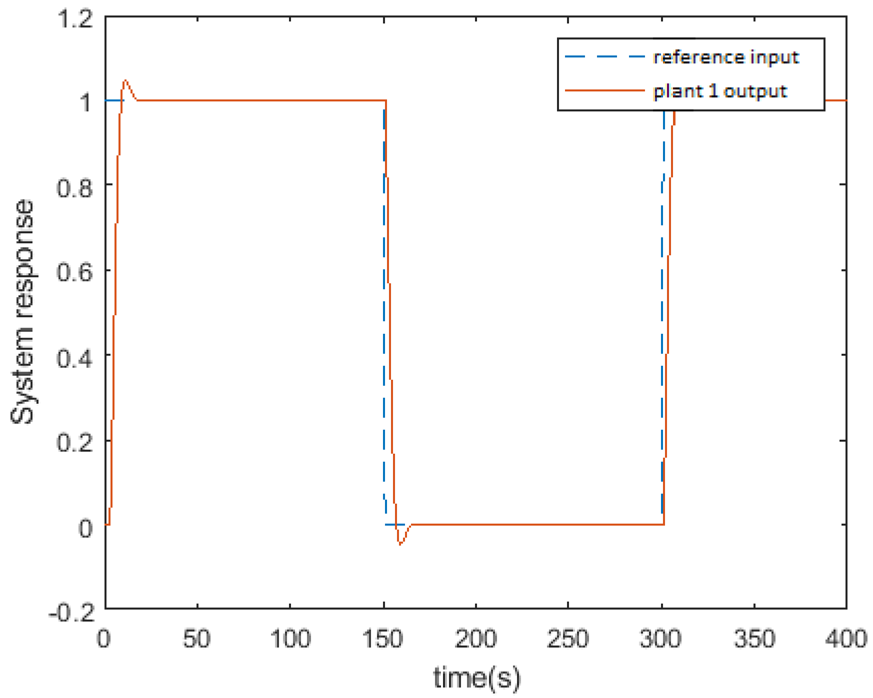


Figure 2.11 System response for plant model 1 by U-model based pole placement control

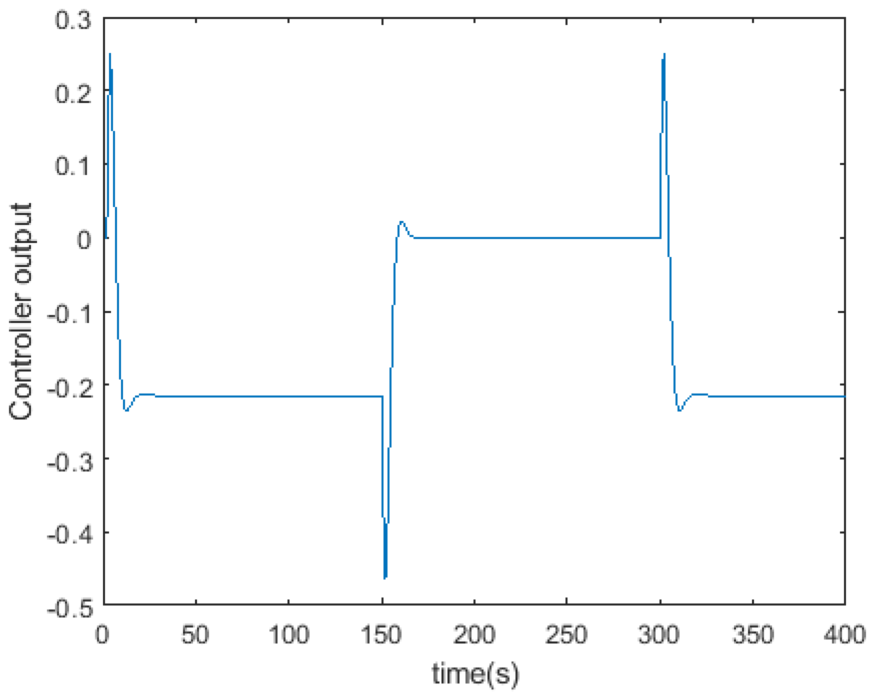


Figure 2.12 Controller output for plant model 1 by U-model based pole placement control

Solution to plant model 2

Since the desired close-loop characteristic equation is the same one as solution to Plant 1 by U-model, there is no need to calculate the controller as equations (2.3.39) to (2.3.43). Utilize the same controller parameter and just figure out corresponding plant from U-model formula (2.3.25):

$$y(t) = l_0(t) + l_1(t)u(t-1) \tag{2.3.47}$$

where

$$l_0(t) = y(t-1) - 0.8y(t-2) + 0.5u(t-2)$$
$$l_1(t) = 1 \tag{2.3.48}$$

The output response of the designed U-model with assigned poles and steady state property is shown in Figure 2.13, and the pole placement controller output is shown in Figure 2.14.

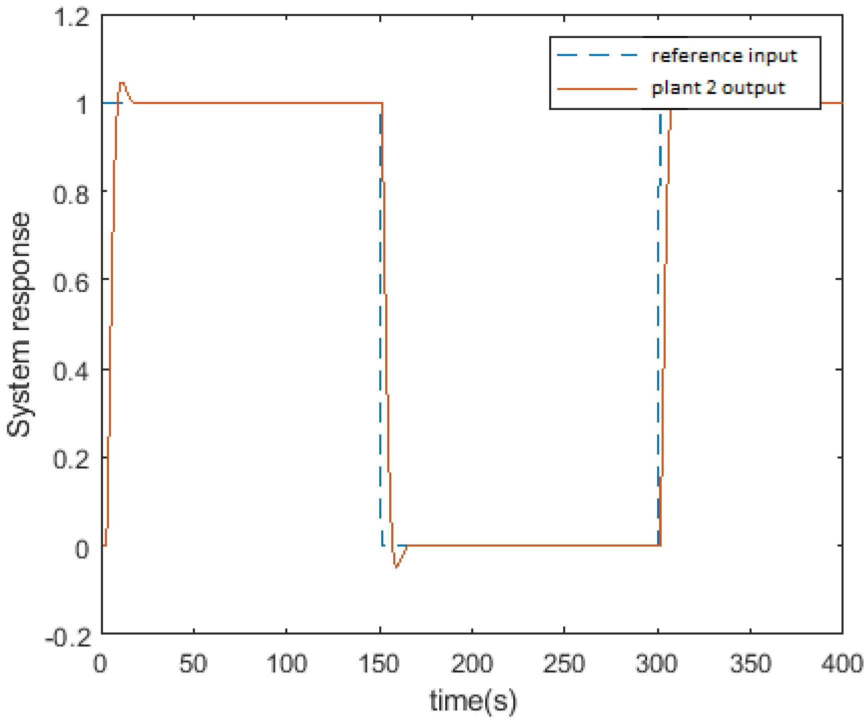


Figure 2.13 System response for plant model 2 by U-model based pole placement control

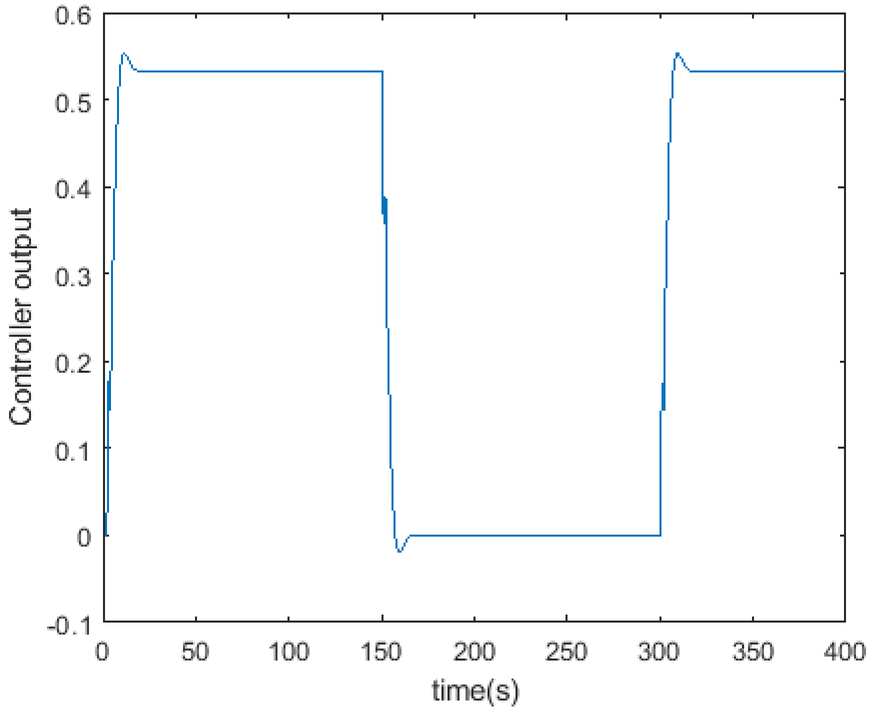


Figure 2.14 Controller output for plant model 2 by U-model based pole placement control

3) Discussions

As shown above, the U-model derived from pole placement with modularisation, obtaining a root as the controller output from a polynomial equation. The simulation results of both classical pole placement and U-model's demonstrate the same control performance achieved; however, the procedure of designing control system by U-model is much concise and generally applicable (once off design for all plant models) compared to classical pole placement (*ad hoc* design with each plant model). To explain the difference, further analysis is given below.

In U-model design, after specifying the desired close-loop characteristic polynomial A_c, polynomials R and S can be resolved through Diophantine equation (which is shown in equation (2.3.30): $R+S=A_c$). As a classical approach in pole placement (Åström and Wittenmark, 1995), the corresponding relationship is given by expression (8): $AR+BS=A_c$ where A and B are the numerator polynomial and the denominator polynomial of a plant model, respectively, which indicate the classical design depending on the plant model. Without determining poles every procedure while plant is changed, the U-model set up a law of R, T and S.

Unlike pole placement method need to calculate R, T and S every time when plant changing, U-model simplifies the routine to complete the design of control system. After the desired plant output $y_d(t)$ is designed, as solution to Plant 2 applies the same desired plant output in solution to Plant 1, the controller output $u(t-1)$ can be directly determined by resolving one of the roots of the U-model. That means, when desired close-loop characteristic equation is set up, no matter how the plant model changed, the procedure from equations (2.3.39) to (2.3.43) is constancy.

This is one of principle for U-model (Zhu and Guo, 2002): The u-model based pole placement design procedure does not depend on the plant model. Only the solution of the designed controller output involves in the plant model.

2.4 Summary

Even the proposition of U-model concept is to establish a framework which provides a generic prototype for using linear approaches to design control systems with smooth non-linear plants, U-model design still performs better in linear control system design. For linear control system design, the fundamental difference between classical approach and U-model approach lays in the design procedure. Classical approach is to design control system with plant model and controller together to find controller output, whereas U-model approach is design a general controller and then use plant models to find the controller output. Even the same control effect are obtained, U-model is superior in generality, concise, and teaching-learning. This study is the first paper to make such comparison with pole placement controller design, which should be also applicable to the other types of linear controllers.

Chapter 3 U-model enhanced controller design for non-minimum phase systems

3.1 Overview

Linear time-invariant dynamic systems are causal and stable whose inverses are causal and unstable are known as non-minimum phase (NMP) systems (Åström, Hagander and Sternby, 1984). Correspondingly the zeros of the discrete-time systems are outside the unit circle, correspondingly the zeros of the continuous-time system are on the right-hand side of the complex plane. A given non-minimum phase system has long time delay response than its minimum phase (MP) system, because of a greater phase delay than the MP system with the equivalent magnitude response (Tomizuka, 1987), a discretized model also could become a NMP delayed model (García, Albertos and Hägglund, 2006) due to the inappropriate sampling. NMP model set has deep root widely encountered in industry (García, Albertos and Hägglund, 2006; Sun *et al.*, 2016; Tsai *et al.*, 2014). In general, a stable numerator polynomial of a transfer function is called stable zero dynamics. It has been noted that a stable zero dynamic of a system is particular important for model matching control and adaptive control.

In modern industrial control process:

1) When an aircraft gaining altitude, an elevator must tilt upward to eventually point the nose of the aircraft up. But in pointing upward, it reduces the net lift on the aircraft and as a result, the aircraft initially decreases altitude before gaining altitude. As a result, elevator deflection appears to perform negatively in comparison to the targeted direction. (Qiu *et al.*, 2016b)
2) In order to keep the water in the boiler at a reference level, the inlet water flow need to be increased if the water drops below the reference. The outlet steam flow is increased, the boiler pressure reduces. More water goes to boil in boiler, creating more air bubbles inside

the water, and this increasing the water volume. Therefore, the boiler water level increases in a short period when the steam outlet flow increasing, and then it falls to the reference point. (Albertos and Mareels, 2010)

3) Not only right-half-plane but also time delay including in NMP. For example, a pipe leads water to the other side into a tank even just after closing it.
4) In mathematical, the phase angle obtained by

$$\begin{aligned}\angle H(\omega) &= \tan^{-1}\left(\frac{\Im\{H(\omega)\}}{\Re\{H(\omega)\}}\right) \\ &= \angle\text{numerator of } H(\omega) - \angle\text{denominator of } H(\omega)\end{aligned} \tag{3.1.1}$$

where $\Re\{\cdot\}$ is the real part and $\Im\{\cdot\}$ is the imaginary part of transfer function $H(\omega)$. According to $s = jw$, zeros located right hand side cause the negative phase. Correspondingly, zeros located outside of the unit circle in z-plane.

Control of NMP systems is a challenge as the unstable reverse response in time domain and the additional phase lag in frequency domain (Sun *et al.*, 2016). A number of literatures have investigated cancellation of NMP system such as Zero-Phase-Error Tracking Controller (ZPETC) (Tomizuka, 1987; Haack and Tomizuka, 1991; Gross, Tomizuka and Messner, 1994), Zero-Magnitude-Error Tracking Controller (ZMETC) (Butterworth, Pao and Abramovitch, 2008; Wen and Potsaid, 2004), and Non-minimum Phase Zeros Ignore (NPZ-Ignore) (Rigney, Pao and Lawrence, 2010; Haack and Tomizuka, 1991). Some the other methods, such as non-causal-series expansion (Rigney, Pao and Lawrence, 2010; Gross, Tomizuka and Messner, 1994), and using the exact unstable inverse and maintain stability of the system by pre-loading initial condition or using non-causal plant input. Most of existing solutions lead to complex controllers with intricate algorithms where the control methods are strictly based upon and/or limited to an explicit plant model. Actually, in modern industrial control process, it is almost impossible to have an exactly accurate filter to cancel the NMP zeros and unstable poles. Where there is any internal uncertainty appears, the cancellation could not execute successfully. Further, it is very much possible to cause problems in system stability.

Some designs focus on sliding mode control (SMC) for its high levels of robust performance in terms of dealing with NMP systems (Mirkin, Gutman and Shtessel, 2012, 2014; Do *et al.*, 2016; Patil *et al.*, 2018). These controllers need the prior knowledge such as fuzzy logic or neural network to establish an online adaptation mechanism to avoid the restrictive constraint on the knowledge of the bounds of uncertain dynamics. Some other designs (Tan, Marquez and Chen, 2003; Liu, Zhang and Gu, 2005; García, Albertos and Hägglund, 2006) proposed conventional controllers, such as PID controllers and New Smith Predictor (NSP) to enhance the capability of reference input tracking and load disturbance rejection for various unstable processes with time delay. These schemes involve more than three controllers and rise the complexity of design process.

Where there is any inaccurate disturbance appears, the cancellation could not execute successfully. Further, it may cause problems in system stability.

From the aforementioned analysis and critical review, this study proposes a pulling theorem to guide replacing zero/pole cancellation (this is a multiplication/division operation) with zero and pole pulling relocation (this is a submission/subtraction operation), which is much less sensitive in stability and NMP issues, and more generic compared with zero/pole cancellation. Further this pulling operation is systematic and concise within a well-constructed framework. After the conversion of NMP model into a stable MP model, the study brings U-model based control design procedure into the control system. Then it conducts a series of simulation experiments to demonstrate the efficiency and effectiveness of the proposed approaches.

Under U-model framework, a desired MP model forms as a generic prototype is established to be a MP reference. To achieve the same output response of the desired MP model, the differences between desired MP model and NMP plant model (in general, the differences are NMP poles and/or zeros) are removed (NB not cancelled) by feedforward controllers. U-model approach provides the linear polynomial to remove all these differences. In z-plane, this part of controller design is to pull the NMP poles and/or zeros outside the unit circle back to the unit circle as the same of poles and/or zeros inside the unit circle of desired MP model. The only task of this proposed method is to set up a desired MP model in this stage. The requirements of the desired MP model are:

1) Desired MP model has the same number of poles and zeros as of NMP plant model
2) All these poles and zeros are MP (inside the unit circle).

The requirements show that design of the desired MP model has a buffer zone, the whole range of the unit circle. Desired MP model becomes a flexible coordination-free model, no matter how the NMP plant model changed.

After removing the differences between desired MP model and NMP plant model, the rest of the design procedure goes to the second stage, the standard U-model control routines (Qiu *et al.*, 2017, 2016b; Zhu and Guo, 2002). This routine is to design an invariant closed loop system, and find the controller output by resolving an inverse dynamic in form of U-model root solver. It provides a once-off package of controller and overcome the unnecessary repetitive design.

From the aforementioned analysis and critical review, this study justifies its contribution below.

1) It proposes a pulling theorem to guide replacing zero/pole cancellation (this is a multiplication/division operation) with zero and pole pulling relocation (this is a summation/subtraction operation), which is much less sensitive in stability and NMP issues, and more generic compared with classical zero/pole cancellation. Further, this pulling operation is systematic and concise within a well-constructed framework.
2) Taking up U-model based design --- a plant model independent design procedure, it separates the control system design and conversion of NMP model into a stable MP model. The design is composed of an invariant controller with specified dynamic and static performance within a stable closed loop and a dynamic inverter of the plant model (implemented by resolving the plant U-model root).
3) It takes up a series of simulation experiments to demonstrate the efficiency and effectiveness of the proposed approaches. It is not only to demonstrate the derived analytical results with numerical tests, but also the case studies are helpful to show the application procedure for potential reader/users.

The rest of the chapter is divided into four sections. Section 2 provides relevant foundation and notations for the the pulling theorem and proves the pulling theorems for zero and pole relocation with Infinite impulse response (IIR) filters within proper structures. Section 3 presents the details of the U-model based control system design procedure for NMP systems.

and conducts simulation bench test of two examples, one from a rotary mechanical system (Wang and Su, 2015). The other is a well-studied unstable NMP model by the other researchers.

3.2 Problem formulations

Consider a linear discrete time SISO dynamic plant model in terms of z transform

$$\frac{Y(z)}{U(z)} = G_p(z) = G_p(A,B) = \frac{B(z)}{A(z)} = \frac{b_0 z^m + b_1 z^{m-1} + \ldots + b_{m-1} z + b_m}{z^n + a_1 z^{n-1} + \ldots + a_{n-1} z + a_n} \quad n > m \tag{3.2.1}$$

where $Y(z)$ is the z transform of output $y(t) \in \mathbb{R}$, $U(z)$ is the z transform of input $u(t) \in \mathbb{R}$, $t \in \mathbb{Z}^+$ is the sampling instant, $\begin{pmatrix} a_0 & \cdots & a_n \end{pmatrix} \in \mathbf{R}^n$ and $\begin{pmatrix} b_0 & \cdots & b_m \end{pmatrix} \in \mathbf{R}^m$ are the coefficient vectors of the denominator polynomial $A(z)$ and numerator polynomials $B(z)$ of the transfer function respectively, B and A are coprime, and $G(z)$ is monic and strictly proper.

The transfer function can be factorised in terms of poles and zeros below (Zhu *et al.*, 2020a).

$$\frac{Y(z)}{U(z)} = G_p(z) = \frac{B(z)}{A(z)} = \frac{\prod (z + z_1)^{m_i}}{\prod (z + p_1)^{n_i}} \quad \sum m_i = m \quad \sum n_i = n \tag{3.2.2}$$

The plant is unstable if one of the poles outside of the unit circle and it is said a non-minimum phase plant if one of the zeros outside of the unit circle. Accordingly, the plant is unstable non-minimum phase while it has both pole(s) and zero(s) outside the unit circle.

Specify a corresponding stable MP plant $\hat{G}_p(\hat{B}, \hat{A})$

$$G_p(z) = \frac{\hat{B}(z)}{\hat{A}(z)} = \frac{\hat{b}_0 z^m + \hat{b}_1 z^{m-1} + \ldots + \hat{b}_{m-1} z + \hat{b}_m}{z^n + \hat{a}_1 z^{n-1} + \ldots + \hat{a}_{n-1} z + \hat{a}_n} \tag{3.2.3}$$

Define zero differencing operator and pole differencing operator between $G_p(B, A)$ and $\hat{G}_p(\hat{B}, \hat{A})$ as feedforward operator B_{FF} and feedback operator A_{FB} respectively, that is

$$\begin{aligned} B_{FF} &= B - \hat{B} = b_{FF0} z^m + b_{FF1} z^{m-1} + \ldots + b_{FF(m-1)} z + b_{FFm} \\ A_{FB} &= A - \hat{A} = a_{FB0} z^n + a_{FB1} z^{n-1} + \ldots + a_{FB(n-1)} z + a_{FBn} \end{aligned} \tag{3.2.4}$$

where

$$\begin{aligned} b_{FFi} &= b_i - \hat{b}_i, \ \ i = 0,...,m \\ a_{FBj} &= a_j - \hat{a}_j, \ \ j = 0,...,n \end{aligned} \tag{3.2.5}$$

Accordingly, the specified stable MP plant $\hat{G}_p\left(\hat{B},\hat{A}\right)$ zeros and poles can be achieved with the differencing operators. It gives

$$\begin{aligned} \hat{B} &= B - B_{FF} \\ \hat{A} &= A - A_{FB} \end{aligned} \tag{3.2.6}$$

3.2.1 Stage one: Pulling theorems

Zeros pulling theorem (Zhu *et al.*, 2020a): Let $G_p\left(B,A\right)$ be a stable NMP transfer function, and $\hat{G}_p\left(\hat{B},A\right)$ is a correspondingly specified stable MP transfer function, G_{zpf} a zero pulling filter, and F_z is a structure mapping function. Then it has $\hat{G}_p\left(\hat{B},A\right)= F_z\left(G\left(B,A\right),G_{zpf}\right)$.

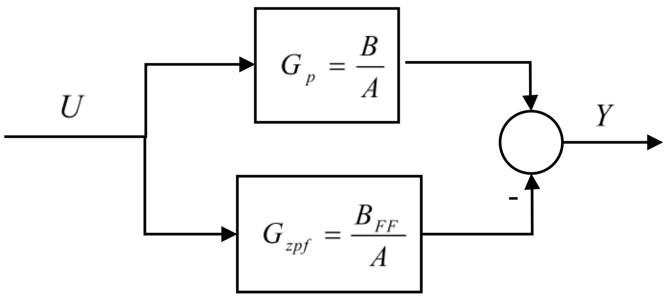


Figure 3.1 Block diagram of zero relocation by pulling theorem

Proof: Define the zero pulling filter as

$$G_{zpf} = \frac{B_{FF}}{A} \tag{3.2.7}$$

And the structure function is defined in Figure 3.1. By block diagram operational algebra, it can derive the transfer function of output against input as

$$\frac{Y}{U} = F_z\left(G(B,A),G_{zpf}\right) = G_p - G_{zpf} = \frac{B - B_{FF}}{A} = \frac{\hat{B}}{A} = \hat{G}\left(\hat{B},A\right) \qquad (3.2.8)$$

Poles pulling theorem (Zhu *et al.*, 2020a): Let $G_p(B,A)$ be an unstable NMP transfer function, and $\hat{G}_p(B,\hat{A})$ is a correspondingly specified stable MP transfer function, G_{ppf} is a pole pulling filter, and F_p is a structure mapping function. Then it has $\hat{G}_p(B,\hat{A}) = F_p(G(B,A),G_{ppf})$.

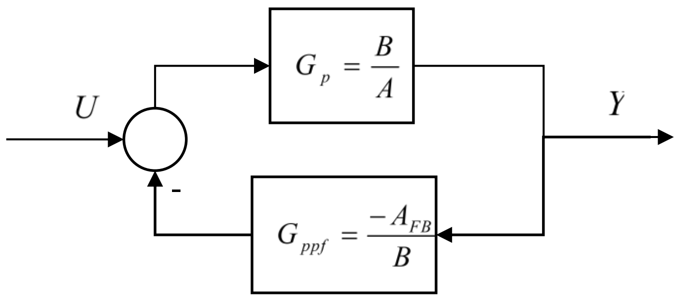


Figure 3.2 Block diagram of pole relocation by pulling theorem

Proof: Define the pole pulling filter as

$$G_{ppf} = \frac{-A_{FB}}{B} \qquad (3.2.9)$$

With the structure mapping function F_P, build up a block diagram connecting $(G(B,A),G_{ppf})$ in Figure 3.2. The whole transfer function between the output-input is derived as

$$\frac{Y}{U} = F_p\left(G(B,A),G_{ppf}\right) = \frac{G_p}{1+G_pG_{ppf}} = \frac{\frac{B}{A}}{1+\frac{B}{A}\left(\frac{-A_{FB}}{B}\right)} = \frac{B}{A - A_{FB}} = \hat{G}\left(B,\hat{A}\right) \qquad (3.2.10)$$

Zero and pole pulling theorem (Zhu *et al.*, 2020a): Let $G_p(B,A)$ be an unstable NMP transfer function, and $\hat{G}_p(\hat{B},\hat{A})$ is a correspondingly specified stable MP transfer function,

G_{zpf} is the zero pulling filter, G_{ppf} is the pole pulling filter, and F_{zp} is a structure mapping function. Then it has $\hat{G}_p\left(\hat{B},\hat{A}\right) = F_{zp}\left(G(B,A), G_{zpf}, G_{ppf}\right)$.

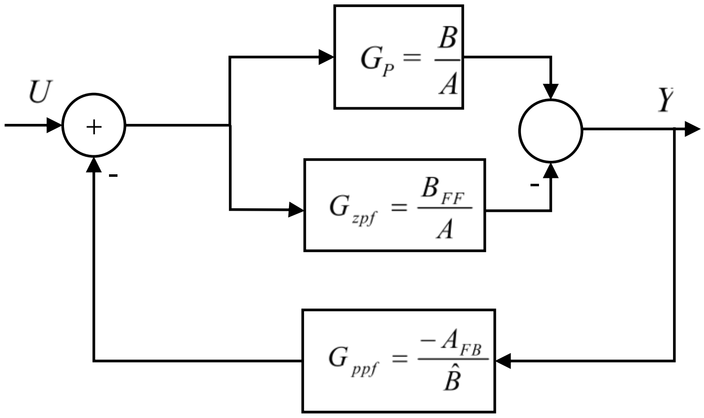


Figure 3.3 Block diagram of zero and pole relocation

Proof: With the structure mapping function F_{zp}, build up a block diagram connecting $\left(G(B,A), G_{zpf}, G_{ppf}\right)$ in Figure 3.3. The whole transfer function between the output-input is derived as

$$\frac{Y}{U} = F_{zp}\left(G(B,A), G_{zpf}, G_{ppf}\right) = \frac{B - B_{FF}}{A - A_{FB}} = \frac{\hat{B}}{\hat{A}} = \hat{G}\left(\hat{B},\hat{A}\right) \tag{3.2.11}$$

Remark 1: U-model based control system has considerable robustness.

To illustrate how potentially robust our proposed U-model based control technique for NMP systems can be, a robustness discussion seems essential and will be presented as follows.

In (3.2.3), $\left(\hat{b}_0 \quad \hat{b}_1 \quad L \quad \hat{b}_k\right)$ is not particularly coefficients. The only task of desired plant $\hat{G}_p$ is to ensure that the NMP zeros is removed and back to MP zeros.

$$\frac{\hat{b}_0 z^m + \hat{b}_1 z^{m-1} + \ldots + \hat{b}_m}{\hat{a}_0 z^n + \hat{b}_2 z^{n-1} + \ldots + \hat{a}_n z^0} = \frac{\hat{b}_0\left(z + \hat{b}_i\right)\left(z + \hat{b}_{i1}\right)\ldots\left(z + \hat{b}_{im}\right)}{\hat{a}_0 z^n + \hat{b}_2 z^{n-1} + \ldots + \hat{a}_n z^0} \tag{3.2.12}$$

where all the desired zeros are located in unit circle, which means

$$\begin{cases} -1 < \hat{b}_i < 1 \\ -1 < \hat{b}_{i1} < 1 \\ \ldots \\ -1 < \hat{b}_{in} < 1 \end{cases} \tag{3.2.13}$$

Substitute (3.2.13) into (3.2.12), due to desired plant model $\hat{G}_p$ could be selected randomly only when zeros inside the unit circle, the desired zeros have a unit circle range to locate. Correspondingly, there is a large range of buffer zone for zero pulling filter. Hence, U-model enhanced control system for NMP zeros has the ability to cope with errors during the execution and also cope with erroneous input.

Remark 2: There are two situations where the original NMP zeros may occur:

1) When the original NMP zero is inside the circle with radius 2 units, the zero pulling filter zero could be designed to be inside of unit circle. The zero pulling filter zero is a MP zero (see, e.g. Figure 3.4 (a)),
2) When the original NMP zero is outside the circle with radius 2 units, the zero pulling filter zero must be outside unit circle to haul the original NMP zero back (see, e.g. Figure 3.4 (b)).

To justify the desired zero is a NMP zero or not, the circle with radius 2 units is used as a criterion for the original NMP zero.

(a) (b)

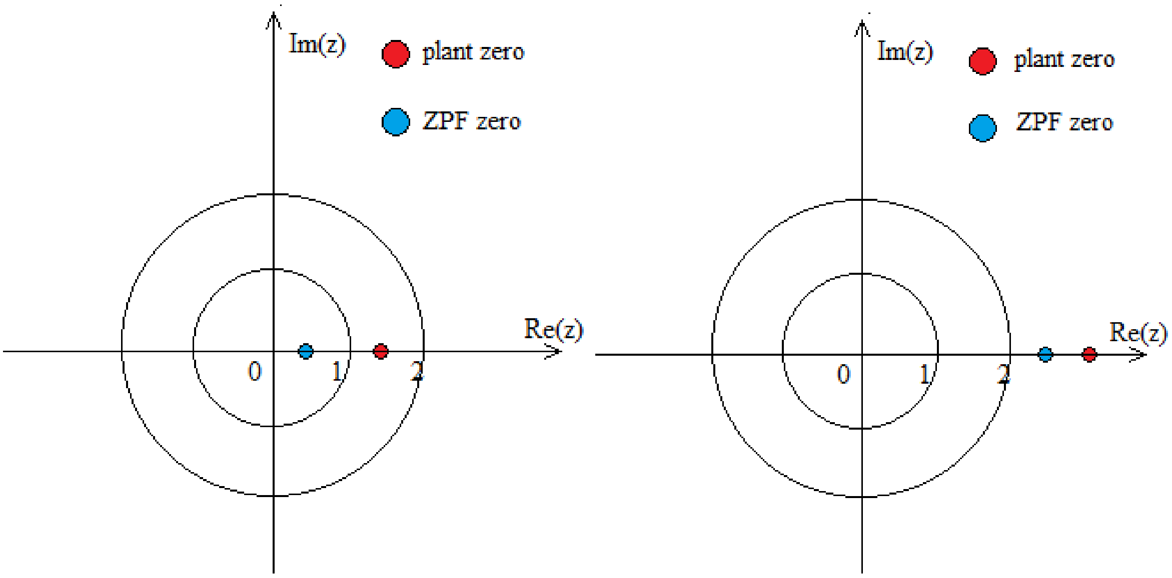


Figure 3.4 (a) Original NMP plant zero is inside the circle with radius 2 units. (b) Original NMP zero is outside the circle with radius 2 units.

3.2.2 Stage two: U-model

U-model control system design aims to free the relationship between main controller and plant. Y is a root solver such as Newton-Ralphson algorithm to make Y become the inverse model of plant model. Therefore, it is easy to design the controller, choosing any linear controller as customised. In other words, no matter how the plant model changes, we only need to find out the roots in Y . The controller could be set as pole placement, PID, etc.

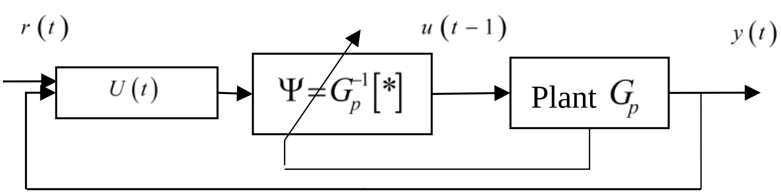


Figure 3.5 Simple block diagram of U-model system design

Combining with pulling theorem, the controller does not change, the root solver Y depends on the desired MP model.

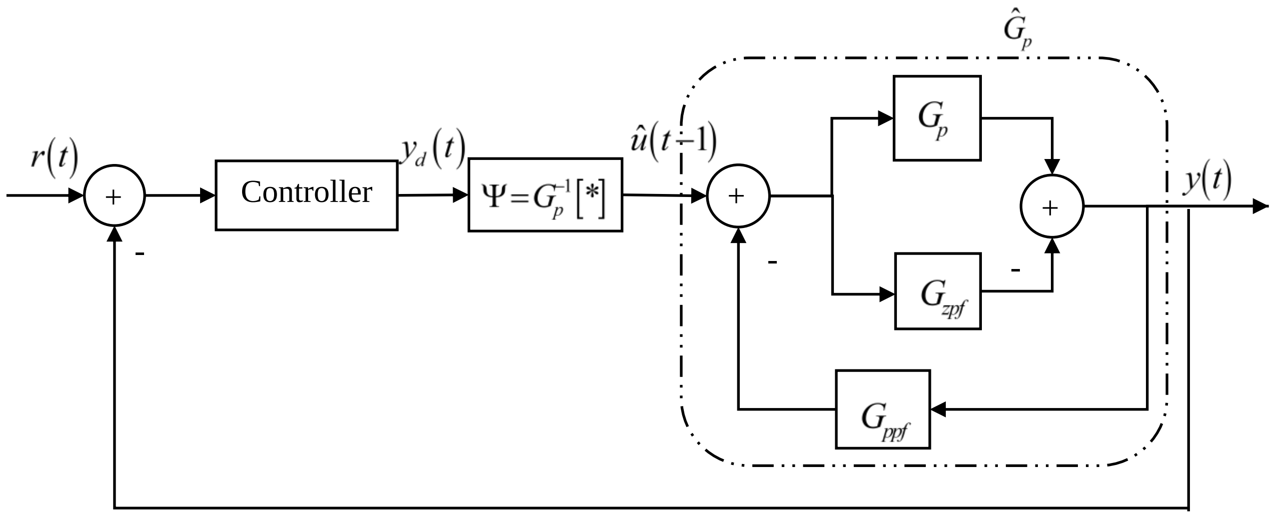


Figure 3.6 Block diagram of U-model enhanced control system for NMP

3.3 Case studies

There are three examples to explain step by step how to apply pulling theorem and U-model structure in controller design.

3.3.1 Case 1: A rotary mechanical system

Consider a rotary mechanical system (Wang and Su, 2015), the non-minimum phase characteristic was obtained by an electrical analogue which could be recognised mechanically with an inertia, a damper, a torsional spring, a timing belt, pulleys and gears (Freeman, Lewin and Rogers, 2005).

$$G_p(s)=\frac{123.853\times10^4(-s+3.5)}{(s^2+6.5s+42.25)(s+45)(s+190)} \tag{3.3.1}$$

To address the system as expression of U-model, choose sampling time $t=0.1s$, the plant model is discretized as

$$G_p(z)=\frac{-6.69z^3+7.856z^2+2.392z+0.001445}{z^4-1.233z^3+0.5356z^2-0.005799z+3.249e-11} \tag{3.3.2}$$

which zeros locate in $z=1.4252$, $z=-0.2504$ and $z=-0.0006$. Set up a desired plant model $\hat{G}_p$ which zeros all located in the unit circle casually, where $z=0.5$, $z=-0.2504$ and $z=-0.0006$ in this case:

$$\hat{G}_p(z)=\frac{-6.69z^3+1.666z^2+0.8386z+0.0005026}{z^4-1.233z^3+0.5356z^2-0.005799z+3.249e-11} \tag{3.3.3}$$

Accordingly, find out feedforward parameters, which can filter the plant model's zeros back to unit circle as the output performance of desired plant model.

$$G_{zpf}=\frac{6.19z^2+1.5534z+0.0009424}{z^4-1.233z^3+0.5356z^2-0.005799z+3.249e-11} \tag{3.3.4}$$

To figure out the U-model inversion $\hat{G}_p^{-1}$, it is important to measure the output of the zero pulling filter G_{zpf}, which is the same as desired plant model $\hat{G}_p$. Reformat the desired plant model into U-model structure

$$\begin{aligned}\hat{y}(t) = {} & 1.233y(t-1) - 0.5356y(t-2) + 0.005799y(t-3) - 3.249e-11y(t-4) - \\ & 6.69u(t-1) + 1.666u(t-2) + 0.8386u(t-3) + 0.0005026u(t-4)\end{aligned} \tag{3.3.5}$$

The controller output can be specified as

$$\hat{u}(t-1) = \frac{\hat{y}(t) - \hat{l}_0}{\hat{l}_1} \tag{3.3.6}$$

where

$$\begin{aligned}\hat{l}_0 = {} & 1.233y(t-1) - 0.5356y(t-2) + 0.005799y(t-3) - 3.249e-11y(t-4) + \\ & 1.666u(t-2) + 0.8386u(t-3) + 0.0005026u(t-4) \\ \hat{l}_1 = {} & -6.69\end{aligned} \tag{3.3.7}$$

Depend on different closed-loop transfer functions, the output responses are demonstrated accordingly. Choose closed-loop transfer function $A_{c1} = \dfrac{0.1761}{z^2 - 1.3205z + 0.4966}$, the results shown below as Figure 3.7 and Figure 3.8.

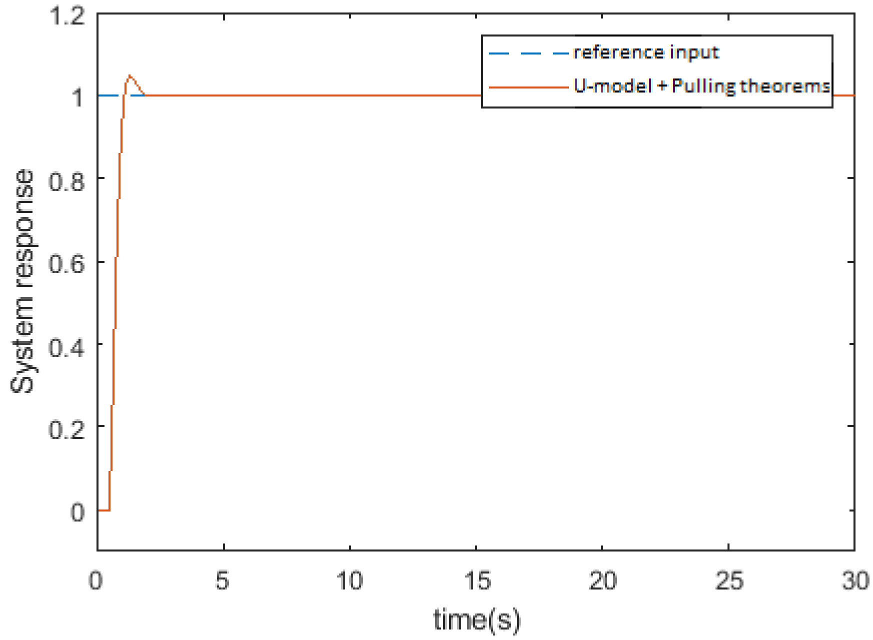


Figure 3.7 System response for case 1 with closed-loop transfer function A_{c1} with damping ratio $x < 1$

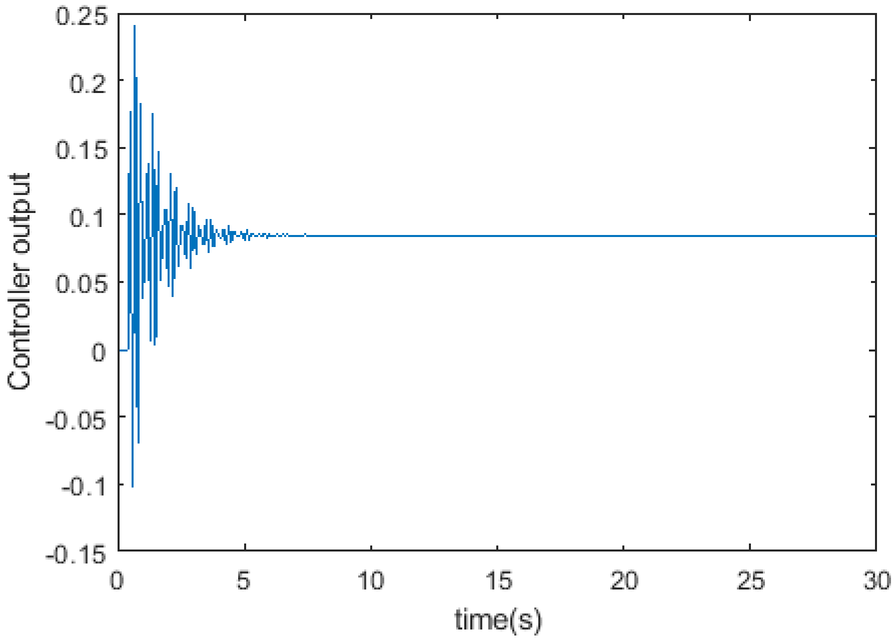


Figure 3.8 Controller output for case 1 with closed-loop transfer function A_{c1} with damping ratio $x < 1$

To achieve damping ratio $x > 1$ for some other situation, select closed-loop transfer function $A_{c2} = \dfrac{0.05848}{z^2 - 1.068z + 0.1263}$, the system response are shown in Figure 3.9 and Figure 3.10.

In Wang and Su (2015), a robust disturbance observer (DOB) based control structure is proposed to stable NMP systems with time delay. Similarly, a nominal plant is employed to compensate the uncertain plant, based on which a prefilter is implemented to acquire desired performance. The nominal plant and weighting functions are selected depends on the plant model, then the Q filter and the virtual controller can be desired. The weighting function really reflect the robust stability condition of the controller design. The DOB based control structure is proposed to suppress the input external disturbances and deal with the internal uncertainties. By U-model scheme, it is free to set up system response as desired with different closed-loop transfer function of pole placement controller for NMP problem. In controller design, it is important to set up the rise time, settling time and so on. Fortunately, the controller could be easily calculated by closed-loop transfer function in U-model structure.

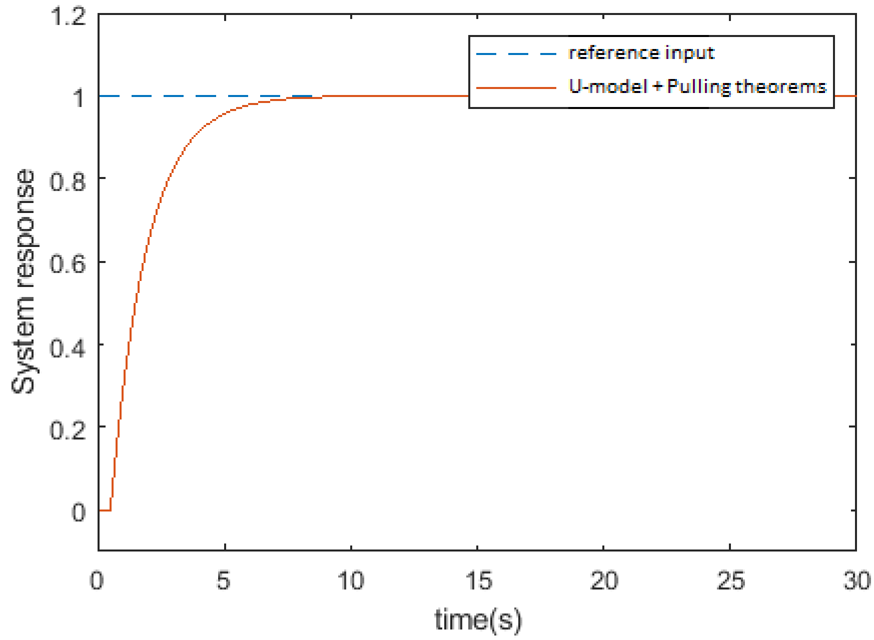


Figure 3.9 System response for case 1 with closed-loop transfer function A_{c2} with damping ratio $x > 1$

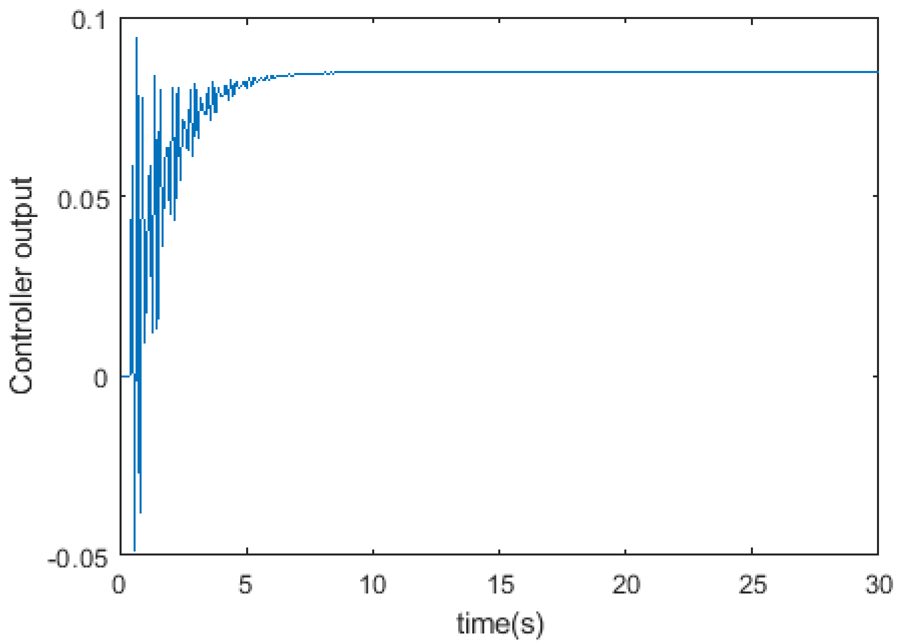


Figure 3.10 Controller output for case 1 with closed-loop transfer function A_{c2} with damping ratio $x > 1$

3.3.2 Case 2: A delayed process with two unstable poles

In this case, U-model control systems with pulling theorem will be compared with Tao's method (Liu, Zhang and Gu, 2005). Consider the following delayed process with two unstable poles, already studied in (García, Albertos and Hägglund, 2006; Liu, Zhang and Gu, 2005)

$$G_p(s) = \exp(-0.3s) \times \frac{2}{3s^2 - 4s + 1} \tag{3.3.8}$$

Choose sampling time $t = 0.005s$, the discretized plant model is

$$G_p(z) = z^{-60} \times \frac{8.352 \times 10^{-6} z + 8.37 \times 10^{-6}}{z^2 - 2.007z + 1.007} \tag{3.3.9}$$

It becomes NMP unstable plant model with zero $z = -1.0022$ and poles $z = 1.0050$, $z = 1.0017$. The corresponding desired model with inside unit circle zeros is selected randomly as

$$\hat{G}_p(z) = \frac{8.352 \times 10^{-6}(z + 0.7)}{(z - 0.5)(z - 0.6)} = \frac{8.352 \times 10^{-6} z + 5.8464 \times 10^{-6}}{z^2 - 1.1z + 0.3} \tag{3.3.10}$$

Therefore, the zero pulling filter could be figured out by plant model and desired model as

$$G_{zpf} = \frac{2.5236 \times 10^{-6}}{z^2 - 2.007z + 1.007} \tag{3.3.11}$$

The output of zero pulling filter is still unstable when the poles outside the unit circle. Therefore, the pole pulling filter is measured out as

$$G_{ppf} = \frac{0.907z - 0.707}{8.352 \times 10^{-6} z + 5.8464 \times 10^{-6}} \tag{3.3.12}$$

To gain the close-loop transfer function of

$$G_{CL} = \frac{0.1042z + 0.02709}{z^2 - 0.8796z + 0.01083} \tag{3.3.13}$$

The controller C can be determined by close-loop transfer function and desired model $\hat{G}_p$ as

$$C = \frac{G_{CL}}{\left(1 - G_{CL}\right)\hat{G}_p} = \frac{0.1042z^3 - 0.08753z^2 - 0.001461z + 0.008127}{8.352\times10^{-6}z^3 - 2.3703\times10^{-6}z^2 - 5.8875\times10^{-6}z - 0.09506\times10^{-6}} \tag{3.3.14}$$

The system response and controller output of the control system fig. 10 without U-model root solver shows below.

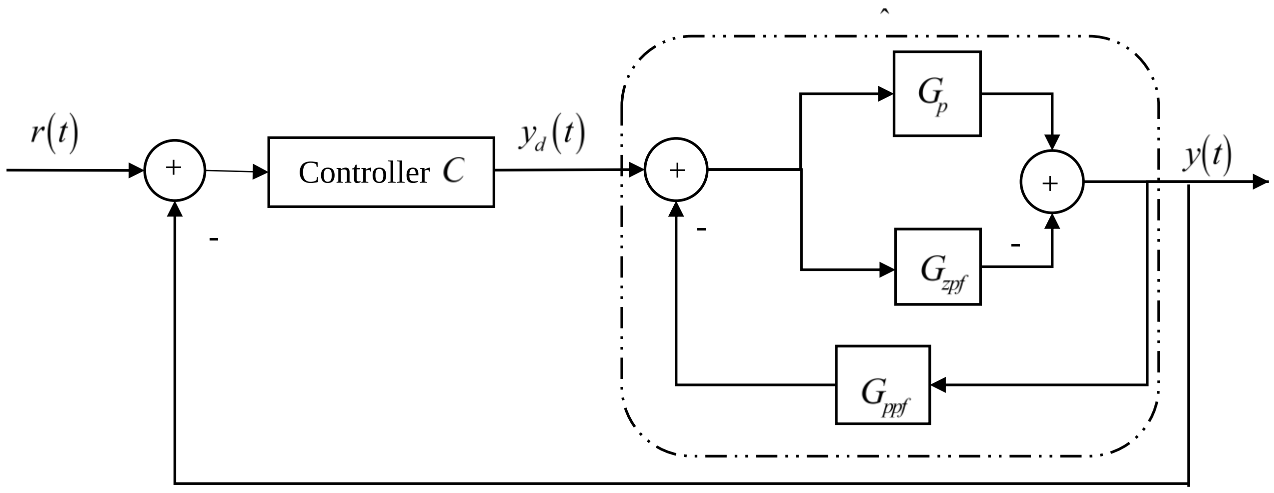


Figure 3.11 Block diagram of NMP control system with Zero and pole pulling theorem

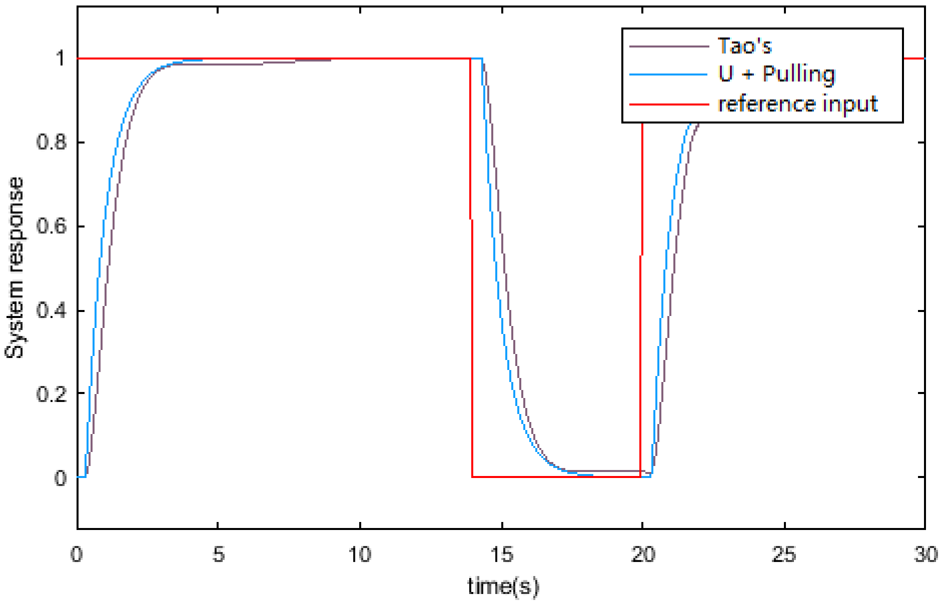


Figure 3.12 System response with pulling theorem and Tao's methods (García, Albertos and Hägglund, 2006)

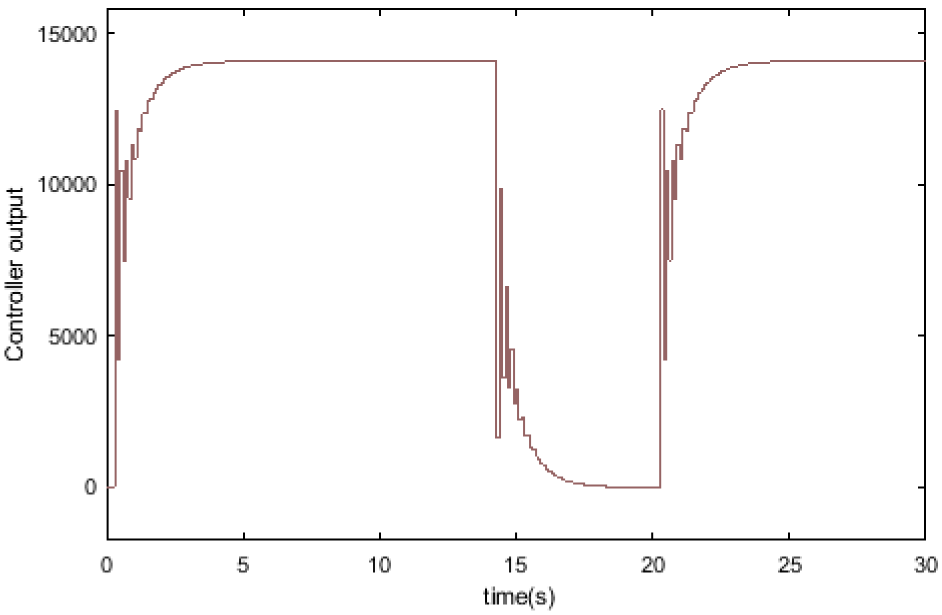


Figure 3.13 Controller output with pulling theorem

From the results, we could see that the controller required a very large power to achieve step system response. Due to the high controller output, it is necessary to divide zero pulling filter into two parts, one for eliminating the delay, and one for pulling the zero back to unit circle.

Apply U-model structure with root solver as Figure 3.14. Consider the combination of root solver and desired model as 1, and then $y_d(t) = y(t)$.

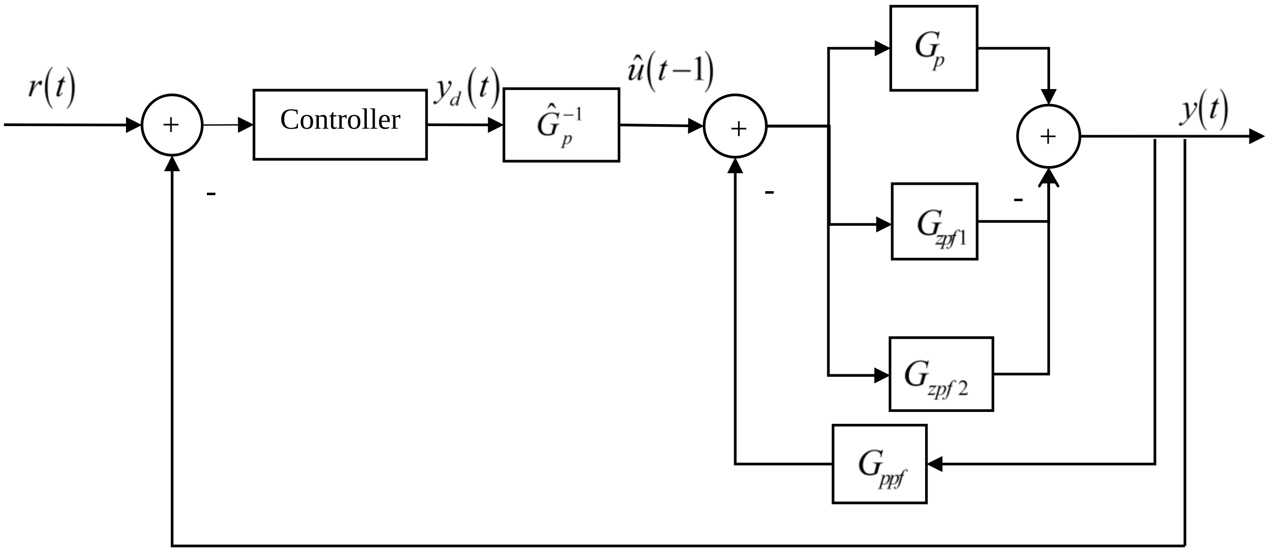


Figure 3.14 U-model Control system structure with time delay

To eliminate the time delay, G_{zpf1} should be a filter with a same transfer function as plant model. The numerator of G_{zpf2} could be any parameter in unit circle, which leads to a desired numerator of desired model. This leads the denominator of G_{ppf} correspondingly changed to the same with the numerator of G_{zpf2}.

Therefore,

$$G_{zpf1} = z^{-60} \times \frac{8.352 \times 10^{-6} z + 8.37 \times 10^{-6}}{z^2 - 2.007z + 1.007} \tag{3.3.15}$$

$$G_{zpf2} = \frac{0.8352z + 0.58464}{z^2 - 2.007z + 1.007} \tag{3.3.16}$$

$$G_{ppf} = \frac{0.907z - 0.707}{0.8352z + 0.58464} \tag{3.3.17}$$

Expressed by U-structure, the desired plant model becomes

$$\hat{y}(t) = 1.1y(t-1) - 0.3y(t-2) + 0.8352u(t-1) + 0.58464u(t-2) \tag{3.3.18}$$

For determining root solver, the output $\hat{u}(t-1)$ should be

$$\hat{u}(t-1)=\frac{\hat{y}(t)-1.1y(t-1)+0.3y(t-2)-0.58464u(t-2)}{0.8352} \tag{3.3.19}$$

The controller C only determined by close-loop transfer function

$$\begin{aligned} C &= \frac{G_{CL}}{1-G_{CL}} \\ &= \frac{0.03871z+0.01977}{z^2-1.10671z+0.10653} \end{aligned} \tag{3.3.20}$$

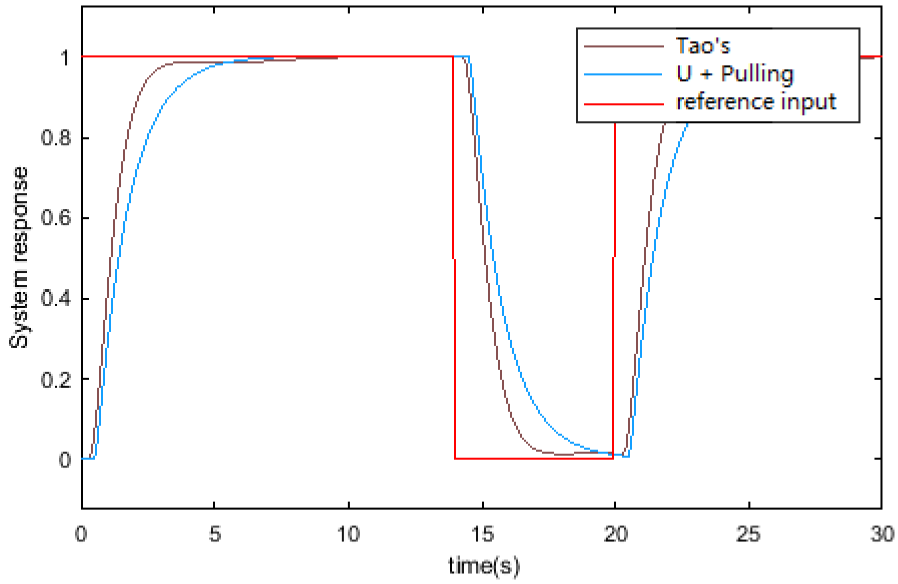


Figure 3.15 System response for case 2 with U-model structure and pulling theorem

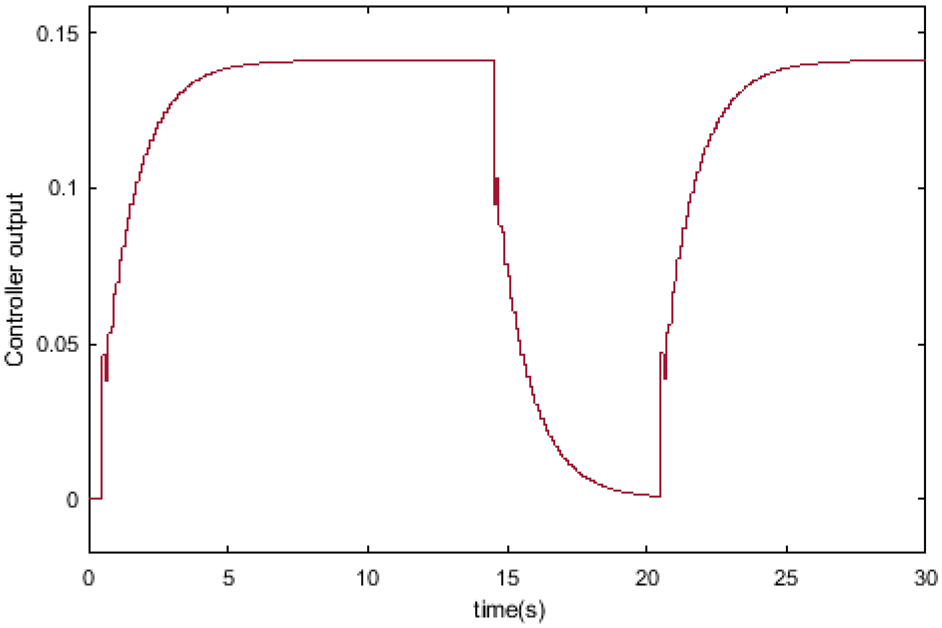


Figure 3.16 Controller output for case 2 with U-model structure and pulling theorem

Note:

If there exist modelling errors such that the real plant is (García, Albertos and Hägglund, 2006)

$$\begin{aligned} G_r(s) &= G_p(s)\left(1 + W_m(s)\right) \\ &= G_p(s)\left(1 + \frac{0.5}{0.1s+1}\right) = G_p(s)\left(\frac{0.1s+0.5}{0.1s+1}\right) \end{aligned} \tag{3.3.21}$$

The Tao's method is unstable. The robust stability condition (García, Albertos and Hägglund, 2006) is

$$\left\| H_1 \frac{KG}{1+KG} W_m \right\|_\infty = 1.4927 > 1 \tag{3.3.22}$$

Correspondingly, in U-model based methods, the real plant become

$$G_r(z) = G_p(z)\left(1 + W_m(z)\right) = G_p(z)\left(\frac{z-0.05182}{z-0.3679}\right) \tag{3.3.23}$$

The system response shown in Figure 3.17.

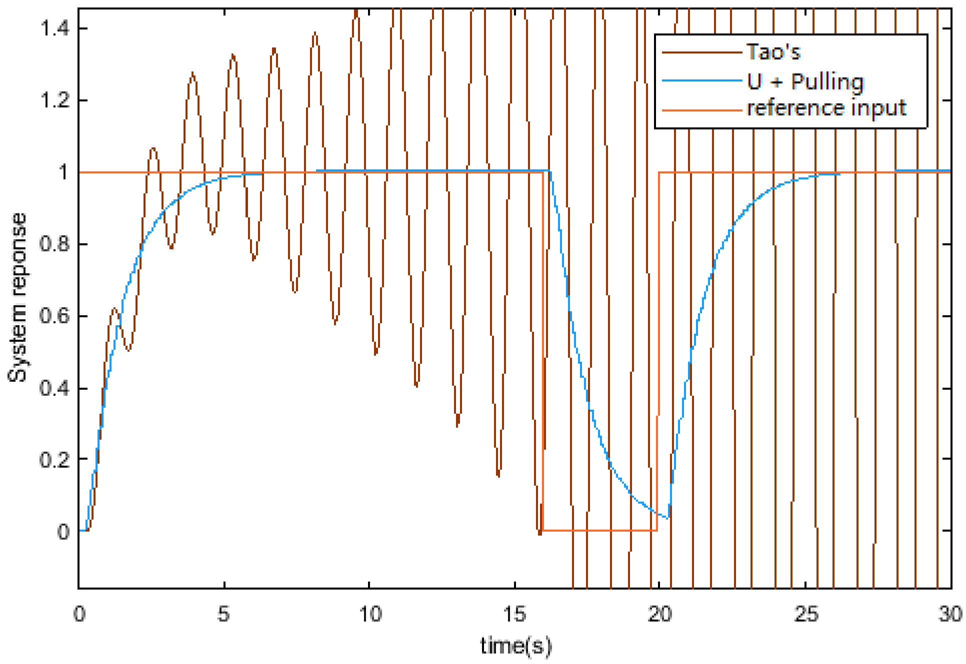


Figure 3.17 System response for case 2 with modelling errors

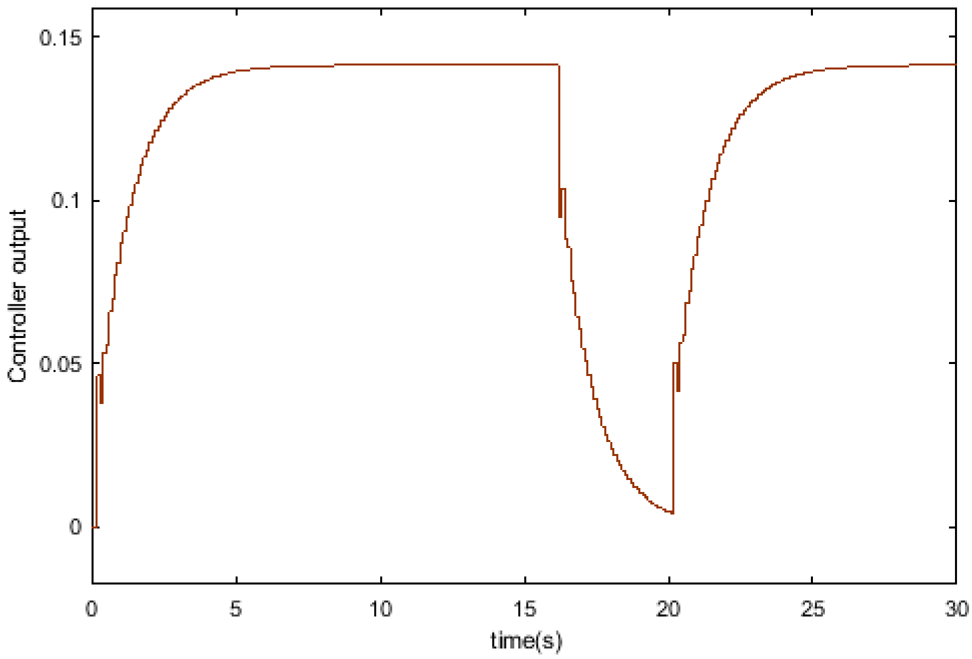


Figure 3.18 Controller output for case 2 with modelling errors

By using Tao's method, there are three controllers must be designed for tracking the reference, rejecting the disturbances and stabilizing the plant model. In U-model scheme, the key work is to design two controller to pull the zeros and poles inside the unit circle. It is not only the feasibility of zero and pole pulling theorem, but also the necessary of root solver by U-model structure. Zero pulling theorem provides the feasibility of time delay eliminate, and also the pull of NMP zero. Within the time delay, the controller output should be very large power to offset the plant model. While applying root solver, $u(t-1)$ can be determined to combined with desired model as a unit signal. At last, the controller can be easily calaculated by close-loop transfer function. The process of this controller design demonstrates the priority of U-model structure in reducing the computational and theoretical complexities. Zero and pole pulling theorem and U-model structure together make the controller design of NMP becomes simple addition and subtraction.

3.3.3 Case 3: An altitude-hold model of an autopilot of Boeing 747

For an altitude-hold model of an autopilot of Boeing 747 using a $2°$ (0.35rad) step command in pitch angle q , the transfer function of the system regarding to height h and elevator δe is (Franklin, Powell and Emami-Naeini, 2002)

$$\frac{h(s)}{\delta e(s)}=\frac{32.7(s+0.0045)(s+5.64)(s-5.61)}{s(s+2.25\pm j2.99)(s+0.0105)(s+0.0531)} \tag{3.3.24}$$

Discrete the transfer function by $t=0.1\text{s}$

$$\frac{h(z)}{\delta e(z)}=\frac{0.1358z^4-0.3341z^3+0.0618z^2+0.2531z-0.1167}{z^5-4.52z^4+8.194z^3-7.462z^2+3.421z-0.6336} \tag{3.3.25}$$

The zero pulling filter

$$G_{zpf}=\frac{-0.3341z^3+0.1175z^2+0.2531z-0.3698}{z^5-4.52z^4+8.194z^3-7.462z^2+3.421z-0.6336} \tag{3.3.26}$$

will construct the desired plant located zeros at $z=0.5,-0.5,0.4,-0.4$.

In the study shown in the book (Franklin, Powell and Emami-Naeini, 2002), select

$$K_{LQG} = \begin{bmatrix} -4 & 2.7 & -112.6 & -4899.1 & -3.2 \end{bmatrix} \tag{3.3.27}$$

and

$$L = \begin{bmatrix} -3.344 & 650.943 & 1.126 & 0.922 & 15.123 \end{bmatrix} \tag{3.3.28}$$

The results of LQG design and U-model control systems with pulling theorem will be presented below.

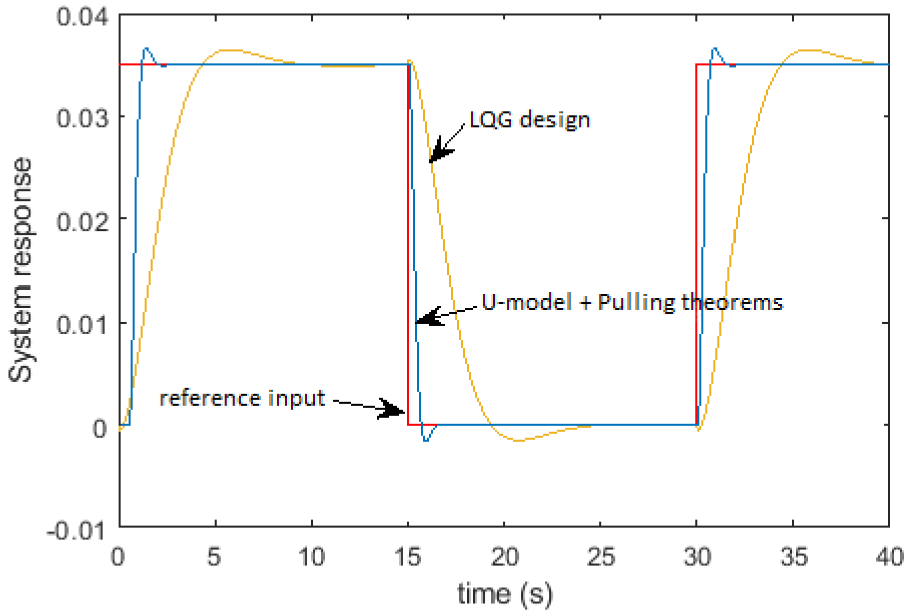


Figure 3.19 Comparison for case 3 between LQG design and U-model based design

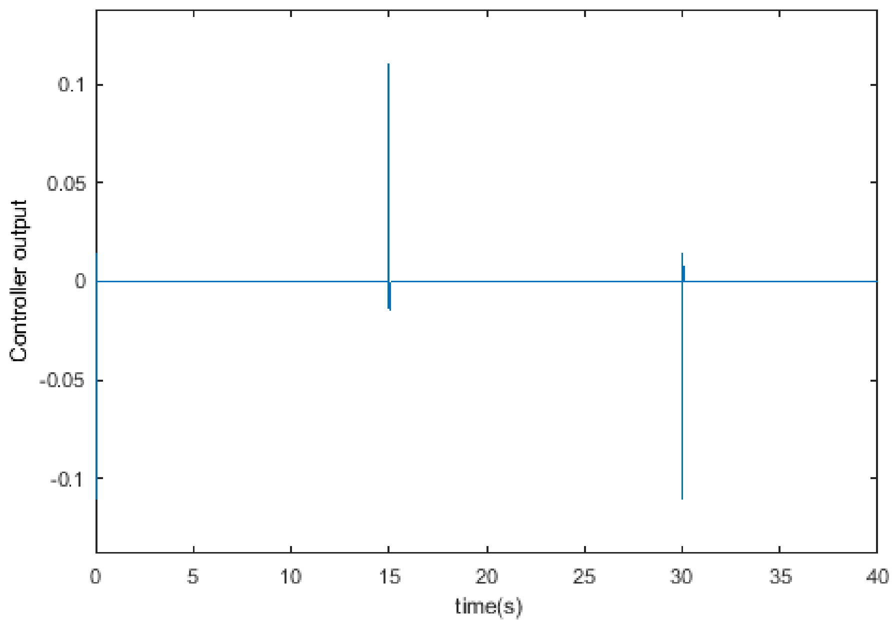


Figure 3.20 Controller output of LQG design for case 3

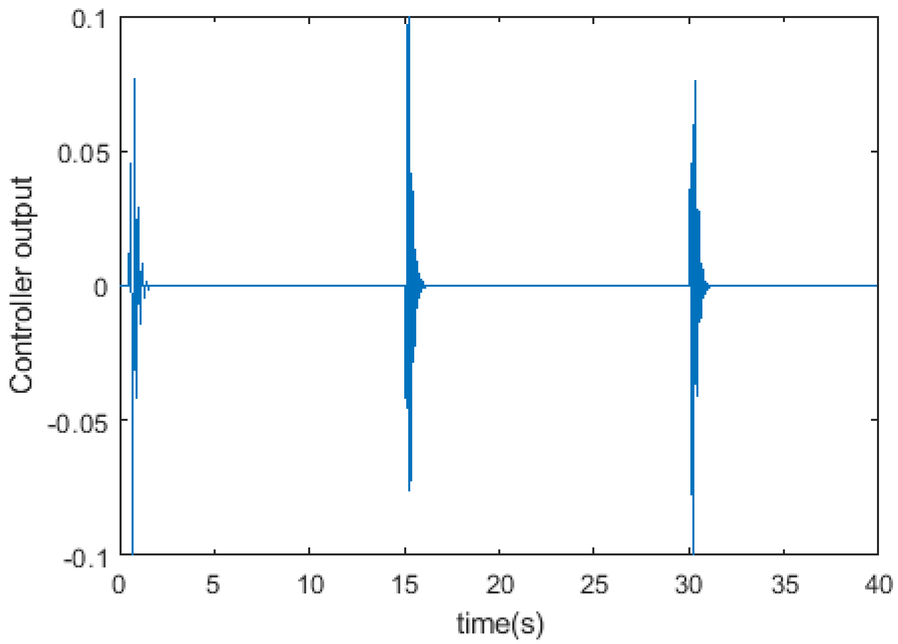


Figure 3.21 Controller output of U-model based design with pulling theorem for case 3

This example shows the difference between U-model and LQG design. LQG design could be the same response of the system with U-model, but it should be tested in many times to find out the characteristics parameter K_{LQG} and L (Qiu *et al.*, 2017). However, we just need the close-loop transfer function to find out the controller by U-model structure. Although there are some oscillation in controller output, we still get a better response of the system by U-model structure (in Figure 3.19).

3.4 Summary

Traditional methods for NMP plant model stand on similar way of cancellation. They apply complex model to cancel the NMP zeros or poles. Cancellation makes the control system becomes sensitive as the controller should be strictly correspond to the plant model. If there is some disturbance or error identification, the cancellation would fail. However, the enhanced U-model based control system design provides a new idea to 'remove' NMP phenomena by mathematical operation --- addition and subtraction. Due to addition and subtraction, it is not sensitive any more where there is a wide range zone to let the relocated plant be inside of unit circle. This good performance can be treated as a kind of considerably robustness. Results of systems output demonstrate that the enhanced U-model based control system design for NMP systems reach the same, even better effect, in response of comparing to other existing methods. The systems output is promising, and the procedure of controller design is much easier to set up and fine-tune. The only one task is to set up a desired MP model to remove the NMP phenomenon. Pulling theorem can be largely applied when any poles and/or zeros of plant model is/are outside the unit circle. Desired MP model is a reference model for finding the differences with NMP plant model. Further, U-model control system has a concise and easy way to free the model and controller. Afterwards, applying standard U-model framework, the whole controller design is convenient to fine-tune. Even when the NMP plant model is changed, the controller design does not need to redesign to adapt new NMP plant model. This proposed method is a model-free and once-off design.

To address the problem, this study proposes a developed controller design based on U-model control method. U-model method provides concise expression of time-varying parameters and a general invariant framework to find the controller output by resolve an inverse dynamic and pole placement. As utilizing U-model discrete-time form, pulling theorem is exploited in this study to pull the NMP zeros and/or poles back to the unit circle by specify a desired minimum phase (MP) model as reference model. This expanded method is a once-off design since it has only one desired MP reference model needs to be pre-set as the same order/form of the NMP plant. Some classic examples are given to show how this proposed method liberates thoughts from complexity of control procedure.

Therefore, the enhance U-model control system design for NMP can be a significant contribution for NMP problem.

Chapter 4 U-model enhanced MRAC controller design with MIT rules for nonlinear plant model

4.1 Overview and introduction

In the past few decades, nonlinear dynamic systems controller design becomes focal point of control process in modern industry. The efficiency and quality of the controllers directly affect the profits, which contributes to the requirement of maximally reduction on controllers' complexity. However, the adaptive control system, which aims to modify the behaviour in response to the variations in the dynamics of the process, is already a complex nonlinear system (Åström and Wittenmark, 2013). Therefore, designing a concise and efficient adaptive control system for nonlinear dynamic model was motivated as a fervent challenge.

The adaptive control system repeatedly and routinely compensates for system dynamics by adjusting controller characteristics so that the overall system performance remains the same or, more precisely, at an optimal level. This control system considers all reduction of plant performance with time. The adaptive control system includes elements for measuring (or estimating) process dynamics and other components for varying the characteristics of the controller accordingly. The controller maintains the overall system performance by adjusting the controller characteristics in a manner. In simple words, the essentials of the adaptive system (Gupta and Yan, 2016) are

1) Identification of system dynamics
2) Decision
3) Modification

When the system model is identified and recognised (which is the difficulty and core procedure) the decision function begins operating. In turn, the decision function activates the modification function to change the particular process parameter and to develop performance (Gupta and Yan, 2016).

The methods of assessing performance are normally two way: by model comparison (or called model reference) and by performance criteria (Gupta and Yan, 2016).

The method by the model comparison selects a model that performances similarity to the desired system characteristics. Among them, all the influences of system characteristics and the effects of interference are known. As universal cognition, the response characteristics of the control system variable parameters are subordinate to the response characteristics of the reference model (Gupta and Yan, 2016). In this procedure, the core role is the error between the model and the control system. An adaptive operation accomplishes this procedure by producing the required system gains. The adaptation path is the minimisation of the integral of the error square (Gupta and Yan, 2016).

The method by the performance criterion applies a general performance index such as the integral of the error squared as continuously computing. The system is adjusted to keep the value of the index at a minimum level. The methods include the Kalman Filter (Baker and Thennadil, 2018) the Stephanopoulos (Stebbins *et al.*, 2018) and Flintoff and Mular (Khaddage, Müller and Flintoff, 2016).

In general, there are two kinds of system model the adaptive control methods are mainly appropriate (Zhang, 2010):

- mechanical systems without significant time delays
- systems dynamics and structure are well known.

Adaptive control methods have the requirement of some kind of identification of the process dynamics. It causes some essential problems such as (Zhang, 2010)

- the requirement of the amount of offline training
- the balance between the persistent excitation of signals for accurate identification and the steady system response for control performance

- the assumption of the process structure
- the model convergence and system stability issues in real applications.

Furthermore, traditional adaptive control methods based on the knowledge of the process structure. They also have difficulties in dealing with nonlinear, structure-variant, or with large time delays processes (Zhang, 2010).

The adaptive controller designs for nonlinear dynamic systems have been proposed and discussed in different adaptive control theory. Zhao proposed smooth adaptive internal model control based on model to simplify the identification of time-varying parameters in presence of bounded external disturbances (Zhao, Wang and Zhang, 2016). Liu applied sliding-mode observer for MIMO uncertain neutral stochastic systems (Liu *et al.*, 2017) and some researchers applied sliding mode model reference adaptive control (MRAC) to deal with nonlinear dynamic systems (Mirkin, Gutman and Shtessel, 2012; Mirkin *et al.*, 2011; Mirkin, Gutman and Shtessel, 2014; Ganesan, Ezhilarasi and Jijo, 2017). MIT based MRAC and modified adaptive control (MAC) are considered for pressure regulation of hypersonic wind tunnel in (Rajani, Krishna and Nair, 2018).

MRAC is widely used in digital adaptive control dynamic systems design with online parameter estimation and adjustment. It can be applied to a nonlinear aircraft model with unknown structural damage (Guo and Tao, 2015) and a quadrotor UAV (Mohammadi and Shahri, 2013). Nonlinear Hydraulic Actuator is designed by adaptive PID and MRAC switch controller in (Zuo *et al.*, 2017). Besides, MRAC also be applied for nonlinear switched systems under asynchronous switching between subsystems and adaptive controller (Xie and Zhao, 2017) and human-robot interaction (Sharifi, Behzadipour and Vossoughi, 2014). Meanwhile, the method of neural network combined with MRAC is considered for solving the nonlinear system (Prakash and Anita, 2011; Lutfy, 2014). Fuzzy logic controller based MRAC is introduced in (Prakash and Anita, 2012). Direct and indirect MRAC is proposed in (Kersting and Buss, 2017) for multivariable piecewise affine systems. Modified MRAC is considered for inverted pendulum compared with MRAC (Pawar and Parvat, 2015).

MRAC is inherently nonlinear so the structure could be completely concise if it is analysed and designed through linear technique or plant. Up to now, the MRAC algorithms for nonlinear dynamic systems obtain numerous data identification or adjustment calculations, which cause strict requirement of microchips resources and time. Enlightened by (Zhu, Zhao and Zhang, 2015; Qiu *et al.*, 2016a; Wu *et al.*, 2011), U-model can be deployed to reduce the complexity of MRAC applying to nonlinear dynamic plant model. U-model is a plant oriental structure apply for including but not limited to most smooth nonlinear dynamic systems. With the root solving algorithms, nonlinear dynamic plant model can be directly applied to MRAC and the reference model could be designed as a linear model.

To address this issue, this study proposes a U-model root solver for the actual nonlinear dynamic plant model, and a virtual plant model to substitute plant model in MRAC with MIT normalised rules.

MIT rule was first proposed in 1960 by the researchers of Massachusetts Institute of Technology (MIT) and applied to construction the autopilot system for aircrafts (Jain and Nigam, 2013). MIT rule can be applied to design a controller with MRAC scheme for any system. The adaptive controller using MIT rule gives acceptable outcomes, but it is very sensitive to the variations of the reference input amplitude. Increasing the values of reference input, system may become unstable. To overcome this problem, normalized algorithm is applied with MIT rule to advance the control law.

- Reference Model: It is used to give an idealised response of the adaptive control system to the reference input.
- Controller: It is usually described by a set of adjustable parameters. In this chapter only one parameter θ is used to designate the control law. The value of θ is mainly reliant on adaptation gain.
- Adjustment Mechanism: This mechanism is used to adjust the parameters of the controller so that actual plant model could track the reference model. Mathematical approaches, such as MIT rule and Lyapunov theory, could be applied to develop the adjusting mechanism. In this chapter we are using MIT rule with Normalized Algorithm.
- The virtual plant model: The virtual plant model is designed as same as the plant model in the classic/standard MRAC controller system, and the virtual parameter (gain) could be time-varying estimated comparing to a reference model.

- The actual nonlinear dynamic plant model: The actual plant model is overlooked by applying the root solving algorithms of U-model.

By introducing basic idea and properties of classic model reference adaptive control with MIT normalised rules and of U-model based approach, this chapter provides comparison and demonstration of these two approaches in design procedures and computational experiments. To explain this chapter, some research questions are listed below, which afterward guides the study to provide proper solutions and findings.

1) How to apply U-model framework for direct model reference adaptive control with MIT normalised rules?
2) How to define the virtual plant model?
3) What are the differences/characteristics of U-model based direct model reference adaptive control with MIT normalised rules compared to the classic one?
4) What is the limitation or restriction of U-model based direct model reference adaptive control with MIT normalised rules control systems design?

The rest of the chapter is divided into three sections. Section 2 shows the problem statement, preliminaries and descriptions of the U-model framework for direct model reference adaptive control, the direct model reference adaptive control for discrete time model, and the MIT normalised rules. In section 3, the whole design procedures of the U-model based direct model reference adaptive control is demonstrated step by step. In section 4, a nonlinear dynamic system model is selected to test the control designs of the two approaches and the corresponding computational experiment simulations are presented.

4.2 Problem statement and preliminaries

The U-model stochastics characteristics are defined as:

$$\begin{cases} U(t) = \sum_{j}^{M} \lambda_j u^j(t-1) + e(t) \\ \lambda_j(t) = \lambda_j(\mathrm{X}, y(t-j), u(t-j-1), e(t-j)) \\ y(t) = U(t), \quad j = 1, 2, \ldots, n \end{cases} \tag{4.2.1}$$

where $U(t)$ denotes a pseudo variable, M is the degree of the control input $u(t-1), \lambda_j(t)$ denotes time-varying parameter including some model's parameter constants X, the past time input $u(t-2), \ldots, u(t-j-1)$, the past time output $y(t-1), \ldots, y(t-j)$ and error signal $e(t-1), \ldots, e(t-j)$.

The MRAC is one of the major approaches in adaptive control. In practical application, direct MRAC is most commonly used. The scheme diagram shown in Figure 4.1 (Akhtar and Bernstein, 2005).

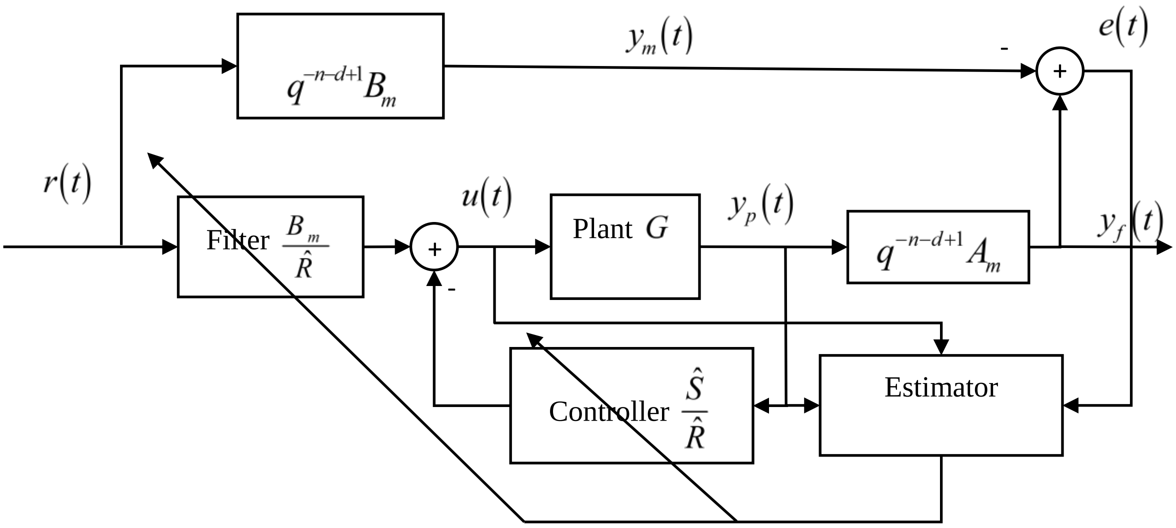


Figure 4.1 Block diagram of discrete time MRAC

The concept behind Model Reference Adaptive Control System is to form a closed loop control system with parameters that can be updated to change the response of the system (Pal *et al.*, 2015).

Apply pole placement to figure out discrete time plant model. Let the plant model written in terms of the forward shift operator $\boldsymbol{q}$ as

$$\boldsymbol{A}(\boldsymbol{q})y_p(t)=\boldsymbol{B}(\boldsymbol{q})u(t) \tag{4.2.2}$$

where y_p and u are output and input of the plant model. $\boldsymbol{A}$ and $\boldsymbol{B}$ are polynomials of degree n and m. They are coprime and $\boldsymbol{B}$ is minimum phase. Defined them as

$$\boldsymbol{A}(\boldsymbol{q})=\boldsymbol{q}^n+a_1\boldsymbol{q}^{n-1}+...+a_n \tag{4.2.3}$$

and

$$\boldsymbol{B}(\boldsymbol{q})=\boldsymbol{q}^m+b_1\boldsymbol{q}^{m-1}+...+b_m \tag{4.2.4}$$

where $m<n$ and $b_0 \neq 0$.

A general linear controller is described as

$$\boldsymbol{R}(\boldsymbol{q})u(t)=\boldsymbol{T}(\boldsymbol{q})r(t)-\boldsymbol{S}(\boldsymbol{q})y_p(t) \tag{4.2.5}$$

To determine the controller,

$$u(t)=\frac{\boldsymbol{T}(\boldsymbol{q})}{\boldsymbol{R}(\boldsymbol{q})}r(t)-\frac{\boldsymbol{S}(\boldsymbol{q})}{\boldsymbol{R}(\boldsymbol{q})}y_p(t) \tag{4.2.6}$$

where $r(t)$ is the reference input signal. Then the plant output could be written as

$$y_p(t)=\frac{\boldsymbol{B}(\boldsymbol{q})\boldsymbol{T}(\boldsymbol{q})}{\boldsymbol{A}(\boldsymbol{q})\boldsymbol{R}(\boldsymbol{q})+\boldsymbol{B}(\boldsymbol{q})\boldsymbol{S}(\boldsymbol{q})}r(t) \tag{4.2.7}$$

The output and the input of the reference model can be described by

$$y_m(t)=\frac{\boldsymbol{B_m}(\boldsymbol{q})}{\boldsymbol{A_m}(\boldsymbol{q})}r(t) \tag{4.2.8}$$

where A_m is monic and stable, $\deg \boldsymbol{A}_m(\boldsymbol{q})$- $\deg \boldsymbol{B}_m(\boldsymbol{q})$= d , $d > 0$.

To force the output of plant model the same as the output of reference model,

$$\frac{\boldsymbol{B}(\boldsymbol{q})\boldsymbol{T}(\boldsymbol{q})}{\boldsymbol{A}(\boldsymbol{q})\boldsymbol{R}(\boldsymbol{q})+\boldsymbol{B}(\boldsymbol{q})\boldsymbol{S}(\boldsymbol{q})}=\frac{\boldsymbol{B}_{\boldsymbol{m}}(\boldsymbol{q})}{\boldsymbol{A}_{\boldsymbol{m}}(\boldsymbol{q})} \tag{4.2.9}$$

which can be easily found as

$$\frac{\boldsymbol{T}(\boldsymbol{q})}{\boldsymbol{A}(\boldsymbol{q})\boldsymbol{R}(\boldsymbol{q})+\boldsymbol{B}(\boldsymbol{q})\boldsymbol{S}(\boldsymbol{q})}=\frac{\boldsymbol{B}_{\boldsymbol{m}}(\boldsymbol{q})}{\boldsymbol{A}_{\boldsymbol{m}}(\boldsymbol{q})\boldsymbol{B}(\boldsymbol{q})} \tag{4.2.10}$$

The roots of the close loop characteristic polynomial $\boldsymbol{A}_{\boldsymbol{m}}(\boldsymbol{q})\boldsymbol{B}(\boldsymbol{q})$ are stable as the roots of $\boldsymbol{A}_{\boldsymbol{m}}(\boldsymbol{q})$ and the roots of $\boldsymbol{B}(\boldsymbol{q})$ are stable. Define the close loop characteristic polynomials as

$$\boldsymbol{P}(\boldsymbol{q}) = \boldsymbol{A}_{\boldsymbol{m}}(\boldsymbol{q})\boldsymbol{B}(\boldsymbol{q}) = b_0\boldsymbol{q}^{(n_m+m)} + p_1\boldsymbol{q}^{(n_m+m)-1} + \ldots + p_{n_m+m} \tag{4.2.11}$$

where $n_m = \deg \boldsymbol{A}_m(\boldsymbol{q})$.

To meet the requirements of (4.2.10),

$$\boldsymbol{T}(\boldsymbol{q}) = \boldsymbol{B}_{\boldsymbol{m}}(\boldsymbol{q}) \tag{4.2.12}$$

$$\boldsymbol{A}(\boldsymbol{q})\boldsymbol{R}(\boldsymbol{q})+\boldsymbol{B}(\boldsymbol{q})\boldsymbol{S}(\boldsymbol{q}) = \boldsymbol{A}_{\boldsymbol{m}}(\boldsymbol{q})\boldsymbol{B}(\boldsymbol{q}) \tag{4.2.13}$$

Define $n_R = \deg \boldsymbol{R}(\boldsymbol{q})$, $n_S = \deg \boldsymbol{S}(\boldsymbol{q})$, $n_e = n + n_R + 1$, and $n_u = n_R + n_S + 2$. (4.2.13) can be written as

$$M\begin{bmatrix}\mathscr{C}(\boldsymbol{R})\\ \mathscr{C}(\boldsymbol{S})\end{bmatrix} = \mathscr{C}(\boldsymbol{P}) \tag{4.2.14}$$

where $M \in R^{n_e \times n_u}$ is the Sylvester matrix

$$M = \begin{bmatrix} 1 & 0 & & 0_{(n-1)\times 1} & 0_{d\times 1} & 0_{(d+1)\times 1} & & 0_{(n+d-1)\times 1} \\ a_1 & 1 & & 1 & b_0 & b_0 & & b_0 \\ a_2 & a_1 & & a_1 & b_1 & b_1 & & b_1 \\ \dots & \dots & & a_2 & \dots & \dots & \dots & b_2 \\ a_n & a_n & \dots & \dots & b_m & b_m & & \dots \\ 0_{(n-1)\times 1} & 0_{(n-2)\times 1} & & a_n & 0_{(n-1)\times 1} & 0_{(n-2)\times 1} & & b_m \end{bmatrix} \tag{4.2.15}$$

The coefficients of $\boldsymbol{R}(q)$, $\boldsymbol{S}(q)$ and $\boldsymbol{P}(q)$ are vectors

$$\mathscr{C}(\boldsymbol{R}) = \begin{bmatrix} r_0 \\ r_1 \\ \dots \\ r_{n_R} \end{bmatrix}, \mathscr{C}(\boldsymbol{S}) = \begin{bmatrix} s_0 \\ s_1 \\ \dots \\ s_{n_S} \end{bmatrix}, \mathscr{C}(\boldsymbol{P}) = \begin{bmatrix} p_0 \\ p_1 \\ \dots \\ p_{n_m+m} \end{bmatrix} \tag{4.2.16}$$

As proved in (Akhtar and Bernstein, 2005), if $n_R \geq n_S$, the control law (4.2.6) is causal. Assume that $\deg \boldsymbol{S} = n - 1$ and $\deg A_m = 2n - m - 1$, so $\deg \boldsymbol{P} = 2n - 1$. Assumed that $\deg \boldsymbol{R} = n - 1$ and $M \in R^{2n \times 2n}$ to gain a minimum degree causal controller. Then

$$\begin{aligned} \boldsymbol{R} &= r_0\boldsymbol{q}^{n-1} + r_1\boldsymbol{q}^{n-2} + \dots + r_{n-1}, \quad r_0 \neq 0 \\ \boldsymbol{S} &= s_0\boldsymbol{q}^{n-1} + s_1\boldsymbol{q}^{n-2} + \dots + s_{n-1}, \quad s_0 \neq 0 \end{aligned} \tag{4.2.17}$$

From (4.2.13) and (4.2.17), we can know that $r_0 = b_0$.

The filtered output signal can be defined for a linear estimation model

$$y_f(t) = \boldsymbol{q}^{-n-d+1}\boldsymbol{A_m}y(t) \tag{4.2.18}$$

From (4.2.2) it becomes

$$y_f(t) = \frac{\boldsymbol{q}^{-n-d+1}\boldsymbol{A_m}\boldsymbol{B}}{\boldsymbol{A}}u(t) \tag{4.2.19}$$

To match the condition (4.2.13), $y_f(t)$ should be

$$\begin{aligned} y_f(t+d) &= \frac{\boldsymbol{q}^{-n+1}(\boldsymbol{AR}+\boldsymbol{BS})}{\boldsymbol{A}}u(t) \\ &= \left(\boldsymbol{R}+\boldsymbol{S}\frac{\boldsymbol{B}}{\boldsymbol{A}}\right)u(t-n+1) \\ &= \boldsymbol{R}u(t-n+1)+\boldsymbol{S}y(t-n+1) \end{aligned} \tag{4.2.20}$$

Define the parameter vector q and the regressed matrix $\varphi(t)$ as

$$\theta = \begin{bmatrix} r_1 \\ \cdots \\ r_{n-1} \\ s_0 \\ \cdots \\ s_{n-1} \end{bmatrix} \in R^{2n-1} \tag{4.2.21}$$

and

$$\varphi(t) = \begin{bmatrix} u(t-1) \\ \cdots \\ u(t-n+1) \\ y(t) \\ \cdots \\ y(t-n+1) \end{bmatrix} \in R^{2n-1} \tag{4.2.22}$$

When $r_0 = b_0$, (4.2.20) can be obtained as the linear identification model

$$y_f(t+d) = b_0 u(t) + \varphi^T(t)\theta \tag{4.2.23}$$

From (4.2.20) and (4.2.23), the model matching control law (4.2.6) can be gained as

$$u(t) = -\frac{1}{b_0}\left[\varphi^T(t)\theta - \boldsymbol{q}^{-n+1}\boldsymbol{B_m}u(t)\right] \tag{4.2.24}$$

With the filtered plant model (4.2.23) and the matching control law (4.2.24), direct adaptive control can be apply with discrete time plant model.

However, when the plant model is unknown, the controller parameters $\boldsymbol{R}$ and $\boldsymbol{S}$ cannot be obtained. So the controller parameters $\boldsymbol{R}$ and $\boldsymbol{S}$ should be estimates by polynomials $\hat{\boldsymbol{R}}(t)$ and $\hat{\boldsymbol{S}}(t)$ in $\boldsymbol{q}$, which is

$$u(t)=\frac{\boldsymbol{B_m}}{\hat{\boldsymbol{R}}(t)}r(t)-\frac{\hat{\boldsymbol{S}}(t)}{\hat{\boldsymbol{R}}(t)}y(t) \tag{4.2.25}$$

The corresponding close loop system is

$$y(t)=\frac{\boldsymbol{BB_m}}{\boldsymbol{A}\hat{\boldsymbol{R}}(t)+\boldsymbol{B}\hat{\boldsymbol{S}}(t)}r(t) \tag{4.2.26}$$

Define the parameter error

$$\bar{\theta}(t)=\hat{\theta}(t)-\theta \tag{4.2.27}$$

where $\hat{\theta}(t)$ is the estimate of q at time t. The filtered output error signal is then defined as

$$e_f(t)=y_f(t)-\boldsymbol{q}^{-n-d+1}\boldsymbol{B_m}r(t) \tag{4.2.28}$$

By the parameter error, the filter output can be written as

$$\begin{aligned}
y_f(t+d)&=\boldsymbol{q}^{-n+1}\frac{\boldsymbol{A_m BB_m}}{\boldsymbol{A}\hat{\boldsymbol{R}}(t)+\boldsymbol{B}\hat{\boldsymbol{S}}(t)}r(t)=\boldsymbol{q}^{-n+1}\frac{\boldsymbol{AR}+\boldsymbol{BS}}{\boldsymbol{A}\hat{\boldsymbol{R}}(t)+\boldsymbol{B}\hat{\boldsymbol{S}}(t)}\boldsymbol{B_m}r(t)\\
&=\boldsymbol{q}^{-n+1}\frac{\boldsymbol{R}+\frac{\boldsymbol{BS}}{\boldsymbol{A}}}{\hat{\boldsymbol{R}}(t)+\frac{\boldsymbol{B}\hat{\boldsymbol{S}}(t)}{\boldsymbol{A}}}\boldsymbol{B_m}r(t)=\boldsymbol{q}^{-n+1}\frac{\boldsymbol{R}u(t)+\boldsymbol{S}y(t)}{\hat{\boldsymbol{R}}(t)u(t)+\hat{\boldsymbol{S}}(t)y(t)}\boldsymbol{B_m}r(t)\\
&=\frac{b_0u(t)+\varphi^T(t)\theta}{\boldsymbol{q}^{n-1}\left[b_0u(t)+\varphi^T(t)\hat{\theta}(t)\right]}\boldsymbol{B_m}r(t)
\end{aligned} \tag{4.2.29}$$

Compared (4.2.23) and (4.2.29),

$$b_0u(t)+\varphi^T(t)\hat{\theta}(t)=\boldsymbol{q}^{-n+1}\boldsymbol{B_m}r(t) \tag{4.2.30}$$

Therefore, the filter output error signal is than

$$
\begin{aligned}
e_f(t+d) &= y_f(t+d) - \boldsymbol{q}^{-n+1}\boldsymbol{B_m} r(t) \\
&= b_0 u(t) + \varphi^T(t)\theta - b_0 u(t) - \varphi^T(t)\hat{\theta}(t) \\
&= -\varphi^T(t)\bar{\theta}(t)
\end{aligned}
\tag{4.2.31}
$$

The model matching error dynamics can be expressed in the n th order fraction form as Figure 4.2 (Middleton and Goodwin, 1990):

$$
\begin{aligned}
&\boldsymbol{A}\xi(t-n) = u(t) \\
&y(t) = \boldsymbol{B}\xi(t-n)
\end{aligned}
\tag{4.2.32}
$$

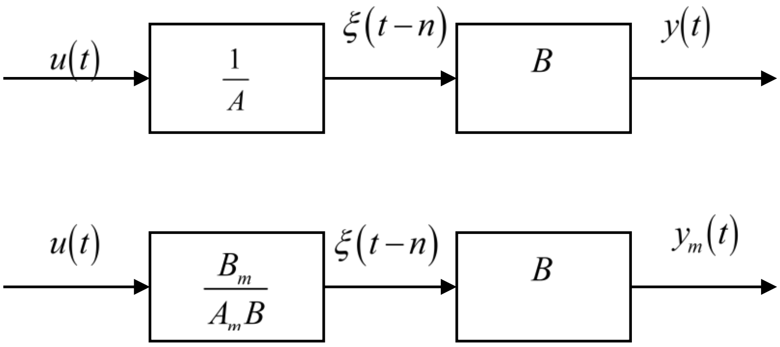


Figure 4.2 Fraction forms of the plant and the reference model

Then, the filtered output can be

$$
y_f(t+d) = \boldsymbol{q}^{-n+1}\boldsymbol{A_m} y(t) = \boldsymbol{q}^{-n+1}\boldsymbol{A_m}\boldsymbol{B}\xi(t-n) = \boldsymbol{q}^{-2n+1}\boldsymbol{P}\xi(t) \tag{4.2.33}
$$

Similarly, the reference model can be expressed in the $2n-1$th order non-minimal fraction form as

$$
\begin{aligned}
&\boldsymbol{A_m}\boldsymbol{B}\xi_m(t-n) = \boldsymbol{B_m} r(t) \\
&y_m(t) = \boldsymbol{B}\xi_m(t-n)
\end{aligned}
\tag{4.2.34}
$$

The output of the plant is compared to the desired response y_m from a reference model. The controller parameters are updated based on this error. Model Reference Adaptive Controller using MIT rule consists of four parts, plant model, reference model, controller and adjustment mechanism (Pal *et al.*, 2015).

So that it can be expressed as

$$\boldsymbol{q}^{-n+1}\boldsymbol{B_m}r(t)=\boldsymbol{q}^{-n+1}\boldsymbol{A}_m\boldsymbol{B}\xi_{\boldsymbol{m}}(t-n)=\boldsymbol{q}^{-n+1}\boldsymbol{P}\xi_{\boldsymbol{m}}(t) \tag{4.2.35}$$

Define

$$\xi_e(t)=\xi(t)-\xi_m(t) \tag{4.2.36}$$

From (4.2.33) and (4.2.35), the d -step ahead filtered output error can be expressed by

$$\begin{aligned} e_f(t+d) &= y_f(t+d)-\boldsymbol{q}^{-n+1}\boldsymbol{B_m}r(t) \\ &= \boldsymbol{q}^{-2n+1}\boldsymbol{P}\left[\xi(t)-\xi_m(t)\right] \\ &= \boldsymbol{q}^{-2n+1}\boldsymbol{P}\xi_e(t) \end{aligned} \tag{4.2.37}$$

Consider a closed loop system in which controller has one adjustable parameter q. The desired closed loop response is specified by a model whose output is y_m. Let e be the error between output y_f of closed loop system and output y_m of reference model. The variable control parameter q is adapted such a way that the cost function (Åström and Wittenmark, 2013),

$$J(q)=\frac{1}{2}e^2 \tag{4.2.38}$$

is minimized, the given cost function can be minimized if we change the parameter in the direction of negative gradient of J, generally known as gradient descent approach, in the following manner (Pawar and Parvat, 2015),

$$\frac{dq}{dt}=-g\frac{\P J}{\P q}=-ge\frac{\P e}{\P q} \tag{4.2.39}$$

This is known as MIT rule (Åström and Wittenmark, 2013). The partial derivative $\P e/\P q$ is called the sensitivity derivative of the system. It tells how the error is effected by the adjustable parameter. The derivative $\P e/\P q$ evaluates under the assumption of q is a constant as the parameter changes are slower than the other variables in the system (Åström and Wittenmark, 2013).

Define the plant, reference model, and model matching error states as

$$
\begin{aligned}
x(t) &= \begin{bmatrix} \xi(t-1) \\ \dots \\ \xi(t-2n+1) \end{bmatrix} \in R^{2n-1} \\
x_m(t) &= \begin{bmatrix} \xi_m(t-1) \\ \dots \\ \xi_m(t-2n+1) \end{bmatrix} \in R^{2n-1} \\
x_e(t) &= x(t) - x_m(t) = \begin{bmatrix} \xi_e(t-1) \\ \dots \\ \xi_e(t-2n+1) \end{bmatrix} \in R^{2n-1}
\end{aligned}
\tag{4.2.40}
$$

Moreover,

$$
\begin{aligned}
\xi_e(t) &= \boldsymbol{q}\xi_e(t-1) \\
&= -\boldsymbol{q}\left[\frac{1}{b_0}\boldsymbol{q}^{-2n+1}\boldsymbol{P} - 1\right]\xi_e(t-1) + \frac{1}{b_0}\boldsymbol{q}^{-2n+1}\boldsymbol{P}\xi_e(t) \\
&= -\boldsymbol{q}\left[\frac{1}{b_0}\boldsymbol{q}^{-2n+1}\boldsymbol{P} - 1\right]\xi_e(t-1) + \frac{1}{b_0}e_f(t+d)
\end{aligned}
\tag{4.2.41}
$$

the state equation of x_e in the controllable canonical form is now then

$$
x_e(t+1) = Ax_e(t) + \frac{1}{b_0}Be_f(t+d), \quad k \geq 0 \tag{4.2.42}
$$

where

$$
A = \begin{bmatrix} -p_0/b_0 & \dots & -p_{2n-1}/b_0 \\ & & 0 \\ I_{(2n-2)\times(2n-2)} & & \dots \\ & & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 1 \\ 0 \\ \dots \\ 0 \end{bmatrix} \tag{4.2.43}
$$

The model matching dynamics becomes

$$x_e(t+1) = Ax_e(t) - \frac{1}{b_0} B\varphi^T(t)\bar{\theta}(t), \; t \geq 0 \tag{4.2.44}$$

Lemma 4.1:

The plant state x is defined by (4.2.40) and the regressor (4.2.22) have a relationship of

$$\varphi(t) = M_0 x(t) \tag{4.2.45}$$

where the non-singular matrix

$$M_0 = \begin{bmatrix} 1 & a_1 & a_2 & \cdots & a_n & & & 0_{1\times(n-2)} & & \\ & & \ddots & & & \ddots & & \ddots & & \\ & & 0_{1\times(n-2)} & & & & 1 & a_1 & a_2 & \cdots & a_n \\ 0_{1\times(d-1)} b_0 & b_1 & \cdots & b_m & & & & 0_{1\times(n-1)} & & \\ & & \ddots & & & & & \ddots & & \\ & & 0_{1\times(n+d-2)} & & & & b_0 & b_1 & \cdots & b_m \end{bmatrix} \tag{4.2.46}$$

Proof:

From the function (4.2.32), the regressor can be expressed as

$$\varphi(t) = \begin{bmatrix} u(t-1) \\ \vdots \\ u(t-n+1) \\ y(t) \\ \vdots \\ y(t-n+1) \end{bmatrix} = \begin{bmatrix} \boldsymbol{A}(\boldsymbol{q})\xi(t-n-1) \\ \vdots \\ \boldsymbol{A}(\boldsymbol{q})\xi(t-2n+1) \\ \boldsymbol{B}(\boldsymbol{q})\xi(t-n) \\ \vdots \\ \boldsymbol{B}(\boldsymbol{q})\xi(t-2n-1) \end{bmatrix} \tag{4.2.47}$$

It can be described as

$$\begin{aligned} u(t-1) &= \boldsymbol{A}(\boldsymbol{q})\xi(t-n-1) \\ &= \left[\boldsymbol{q}^n + a_1\boldsymbol{q}^{n-1} + \cdots + a_n\right]\xi(t-n-1) \\ &= \xi(t-1) + a_1\xi(t-2) + \cdots + a_n\xi(t-n-1) \\ &= \begin{bmatrix}1 & a_1 & \cdots & a_n\end{bmatrix}\begin{bmatrix}\xi(t-1) & \cdots & \xi(t-n-1)\end{bmatrix}^T \\ &\ \vdots \end{aligned} \tag{4.2.48}$$

$$\begin{aligned} u(t-n+1) &= \boldsymbol{A}(\boldsymbol{q})\xi(t-2n-1) \\ &= \left[\boldsymbol{q}^n + a_1\boldsymbol{q}^{n-1} + \cdots + a_n\right]\xi(t-2n-1) \\ &= \xi(t-n+1) + \cdots + a_n\xi(t-2n-1) \\ &= \begin{bmatrix} 1 & a_1 & \cdots & a_n \end{bmatrix}\begin{bmatrix} \xi(t-n+1) & \cdots & \xi(t-2n-1) \end{bmatrix}^T \\ &\vdots \end{aligned} \tag{4.2.49}$$

$$\begin{aligned} y(t) &= \boldsymbol{B}(\boldsymbol{q})\xi(t-n) \\ &= \left[b_0\boldsymbol{q}^n + b_1\boldsymbol{q}^{n-1} + \cdots + b_m\right]\xi(t-n) \\ &= b_0\xi(t-d) + \cdots + b_m\xi(t-n) \\ &= \begin{bmatrix} b_0 & b_1 & \cdots & b_m \end{bmatrix}\begin{bmatrix} \xi(t-d) & \cdots & \xi(t-n) \end{bmatrix}^T \\ &\vdots \end{aligned} \tag{4.2.50}$$

$$\begin{aligned} y(t-n+1) &= \boldsymbol{B}(\boldsymbol{q})\xi(t-2n+1) \\ &= \left[b_0\boldsymbol{q}^n + b_1\boldsymbol{q}^{n-1} + \cdots + b_m\right]\xi(t-2n+1) \\ &= \xi(t-n+1) + \cdots + a_n\xi(t-2n-1) \\ &= \begin{bmatrix} b_0 & b_1 & \cdots & b_m \end{bmatrix}\begin{bmatrix} \xi(t-n-d+1) & \cdots & \xi(t-2n-1) \end{bmatrix}^T \\ &\vdots \end{aligned} \tag{4.2.51}$$

It can be seen that M_0 is the $(2n-1)\times(2n-1)$ submatrix of M^T, which omit the first row and first volume of M^T. Remark that $\det M_0 = \det M$. When A and B are relatively co-prime by supposition, then M is non-singular, M_0 is non-singular.

Consider the variable lead compensator with integral action by a low level direct controller in discrete time transfer function,

$$G_c(z) = K\frac{z-\tilde{A}}{z-\tilde{B}} - \frac{C}{z-1} \tag{4.2.52}$$

with its equivalent continuous-time transfer function (Tustin transformation) given as (Tang, De Silva and Poo, 2001)

$$G_c(s) = K_c\frac{t s+1}{a t s+1} + \frac{C}{z-1} \tag{4.2.53}$$

The digital controller characteristic parameters A , B , K and C are fully programmable during run-time (Tang, De Silva and Poo, 2001).

4.3 Controller design

The actual nonlinear plant model G apply root solver $\psi = U^{-1}$. In order to apply the linear direct MRAC controller design, it is laconic to construct a virtual plant model G_v as the combination of the root solver Υ and the actual nonlinear plant model G is considered to be a constant 1 mathematically. The whole structure shown as follow:

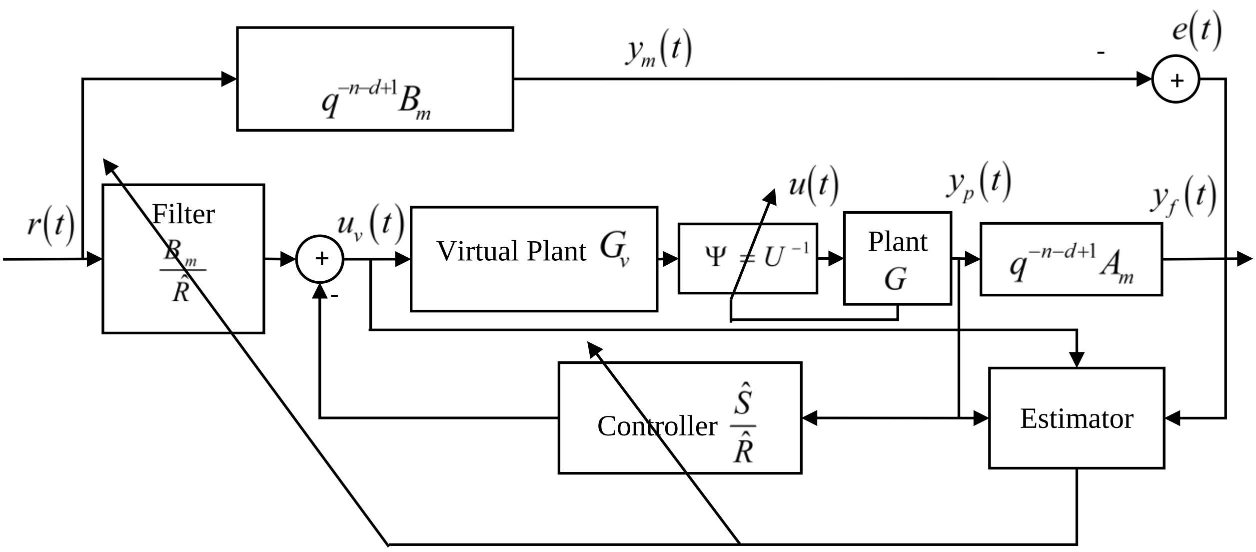


Figure 4.3 U-model based MRAC control systems

As shown in Figure 4.3, the virtual plant model G_v is $\frac{B}{A}$, which mathematically same as plant model in Figure 4.1. The actual plant model in Figure 4.3can be smooth nonlinear dynamic system.

A step-by-step procedure shows below to design the U-model based MRAC scheme:

Step 1 Choose the reference model G_m and virtual plant model G_v;

Step 2 Choose reference input signal y_r, adaptive gain γ;

Step 3 Sample the reference model output y_m and virtual output y_v;

Step 4 Apply root solver to root the actual nonlinear plant model;

Step 5 Obtain the actual plant output y_p;

Step 6 Compute the controller output u by the error between reference model output y_m and actual plant model output y_p;

Step 7 $t \to t+h$, back to Step 3, continue to loop.

This is the online estimation and control law for a class of smooth nonlinear dynamic systems with U-model based MRAC structure.

4.4 Case studies

Consider a nonlinear dynamic plant model G expressed by a Hammerstein model (Zhu and Guo, 2002) as the follow

$$\begin{aligned} y(t) &= 0.5y(t-1) + x(t-1) + 0.1x(t-2) \\ x(t) &= 1 + u(t) - u^2(t) + 0.2u^3(t) \end{aligned} \tag{4.4.1}$$

Convert the plant model into U-model expression as:

$$y(t) = \lambda_0(t) + \lambda_1(t)u(t-1) + \lambda_2(t)u^2(t-1) + \lambda_3(t)u^3(t-1) \tag{4.4.2}$$

where

$$\begin{aligned}
\lambda_0(t) &= 0.5y(t-1)+1+0.3x(t-2) \\
\lambda_1(t) &= 1 \\
\lambda_2(t) &= -1 \\
\lambda_3(t) &= 0.2
\end{aligned} \tag{4.4.3}$$

With the U-model expression, Newton-Raphson algorithm is applied to find out the root of the nonlinear system.

Assume that the virtual model is

$$G_v(z) = \frac{k_v}{z^2 - 1.3205z + 0.4966}, \quad k_v \text{ is unknown} \tag{4.4.4}$$

In MATLAB, k_v is set as 1 for initialization. For comparison, standard MRAC with MIT normalised algorithm will be present by applying the virtual model (4.4.4) as plant model G and the whole controller scheme shown in Figure 4.3.

The reference model is

$$G_m(z) = \frac{k_m}{z^2 - 1.3205z + 0.4966}, \; k_m = 0.1761 \tag{4.4.5}$$

Let the adaptive gain $\gamma = 0.1$, $\alpha = 0.01$, and $\beta = 2$. y_r is a square wave signal. The amplitude $r = 1$. The results shown in Figure 4.4 and Figure 4.5 compared the U-model based MIT normalised MRAC control system applied for Hammerstein model and standard MIT normalised MRAC control system applied for virtual model without root solver U^{-1} nor Hammerstein model.

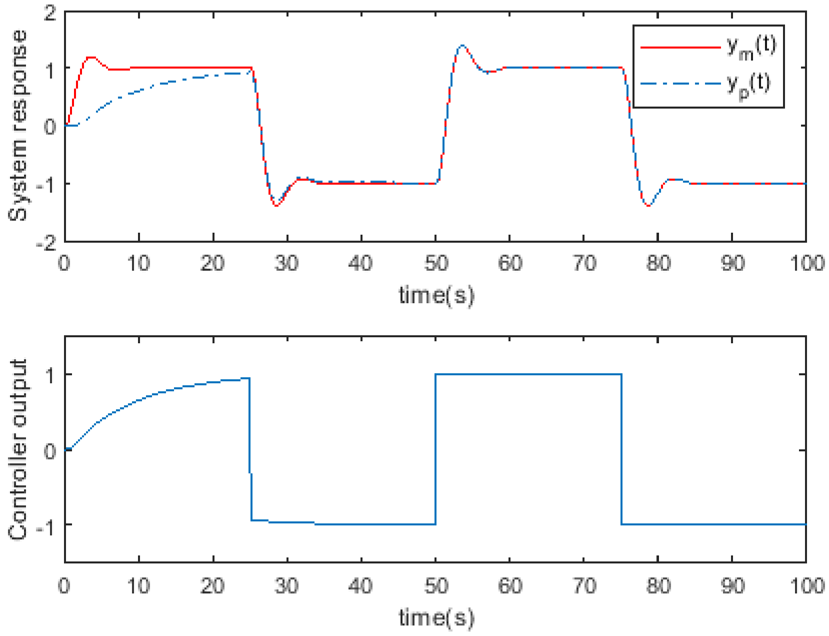


Figure 4.4 System and controller output response of standard MIT normalised MRAC control system on Hammerstein model

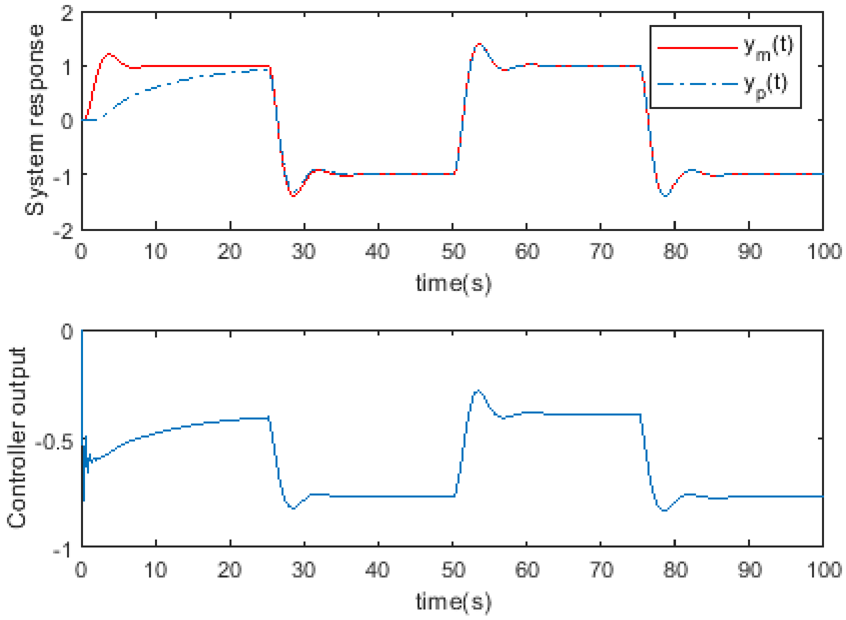


Figure 4.5 System and controller output response of U-model based MIT normalised MRAC control system on Hammerstein model

Discussions

Figure 4.4 and Figure 4.5 show that the U-model based MIT normalised MRAC control system maintain almost the same system response as the standard one. After 50s both the control systems track the reference model signal. However, the plant model of U-model based control system is nonlinear system, and the plant model of standard MRAC control system is linear system. Within the inversion and the virtual plant model, it demonstrates that nonlinear plant model also could be directly applied to MRAC control system.

4.5 Summary

The plant model of U-model based MIT normalised MRAC control system is a nonlinear dynamic system (4.4.1) and the plant model of standard MIT normalised MRAC control system is a linear model. However, both control system designs maintain the consistent system responses and the same reference outputs. Owing to the linear plant model in standard MIT normalised MRAC is actually identical with the virtual plant model in U-model based MIT normalised MRAC control system, the combination output of U-model root solver and actual nonlinear dynamic model can be verified as 1. Accordingly, the nonlinear dynamic system can be applied to obtain the same system response as a MRAC control design for linear plant model. Although it is now not a mature method for all types of nonlinear system, it provides a novel idea to directly adapt nonlinear system with adaptive control system for further research.

Chapter 5 U-model enhanced MRAC controller design with Lyapunov function for nonlinear plant

5.1 Overview

From last chapter we could understand a reference model represents the design specifications of model reference adaptive control system. The model states, the model inputs and the error between plant and model output generate the appropriate control signals. The adaptive control is applied to adjust the control law for the plant parameters are not well known (Ampsefidis, Białasiewicz and Wall, 1993). The direct Lyapunov method adjusts the control law to minimize the error between the plant and the ideal target system states.

Parks firstly used Lyapunov's method to design a stable adaptive controller for SISO (Parks, 1966). Grayson, Winsor and Roy also use the same technique for the design of multiple input multiple output model reference adaptive control systems (Grayson, 1965; Winsor and Roy, 1968). However, they did not meet the Erzberger's perfect model following (PMF) conditions. That is, only if there exists a certain structural relationship between the plant and the model, these adaptive algorithms works.

For this purpose, Landau proposed another adaptive algorithm (whose stability is ensured by the hyperstability criterion of Popov) for multiple input multiple output continuous system subject to the PMF conditions (Landau, Lozano and M'Saad, 1998). After that, some researchers reported designs of adaptive controllers which do not need to satisfy the PMF conditions (Mabius and Kaufman, 1975; Broussard and O'Brien, 1980).

To adjust the parameters in adaptive systems, Lyapunov's stability theory is applied. The differential equation for error $e = y - y_m$ is resulting afterward for adjusting the parameters (Mani, Sivaraman and Kannan, 2018). Using this equation, the error could be decrease to zero by determining the appropriate value from adaptation technique and Lyapunov function. Such that the adaption gains dv/dt is adverse semi definite. This is particularly based on applying the Lyapunov theory. Therefore, to achieve the convergence, the uniform observability for the reference signal and system should be obtained. Furthermore, it is essential to attain a determined excitation. First order MRAC based on stability theory.

From that on, the Lyapunov stability theory demonstrated the controlled structure's precise information is not necessary, which is particularly appropriate for controlling the structures with parameter uncertainties. As a class method for parameter uncertainties, Lyapunov-MRAC still be applied by many researchers nowadays (Hsu *et al.*, 2015; Farajzadeh-D, Hosseini Sani and Akbarzadeh, 2019; Csanádi, Tar and Bitó, 2020). In this method, the reference model with the same structural parameters but higher damping property is designed. Results showed that the Lyapunov-MRAC can stably control the structural response by tracking the response of the reference model.

Lyapunov-MRAC is similar to normalised MIT algorithms a method adjusting the controller gain to make the errors between plant model and reference model approaching to zero. However, when the adaptive gain g is an excessive value, or performance of y_m is poor, MIT normalised MRAC may obtain an unstable close loop system (Åström and Wittenmark, 2013). Lyapunov-MRAC is introduced to improve this phenomenon.

By introducing basic idea and properties of classic model reference adaptive control with MIT normalised rules and of U-model based approach, this chapter provides comparison and demonstration of these two approaches in design procedures and computational experiments. To explain this chapter, some research questions are listed below, which afterward guides the study to provide proper solutions and findings.

1) How to apply U-model framework for direct model reference adaptive control with MIT normalised rules?
2) How to define the virtual plant model?
3) What are the differences/characteristics of U-model based direct model reference adaptive control with MIT normalised rules compared to the classic one?

4) What is the limitation or restriction of U-model based direct model reference adaptive control with Lyapunov function control systems design?

The rest of the chapter is divided into three sections. Section 2 shows the problem statement, preliminaries and descriptions of the U-model framework for direct model reference adaptive control, the direct model reference adaptive control for discrete time model, and the Lyapunov function. In section 3, the whole design procedures of the U-model based direct model reference adaptive control is demonstrated step by step. In section 4, a nonlinear dynamic system model is selected to test the control designs of the two approaches and the corresponding computational experiment simulations are presented.

5.2 Problem statement and preliminaries

The U-model based MRAC structure is the same as chapter 4. This chapter focus on the Lyapunov methods applied in U-model based direct MRAC for nonlinear dynamic system.

Let the virtual plant model written in terms of the forward shift operator $\boldsymbol{q}$ as

$$\frac{y_v(t)}{u(t)} = \frac{\boldsymbol{B}_v(\boldsymbol{q})}{\boldsymbol{A}_v(\boldsymbol{q})} \tag{5.2.1}$$

Derive a recursive least squares update law for the parameter vectors $\hat{\theta}(t)$ of a controller that asymptotically drives $e_f(t+d)$ to zero. The retrospective cost function decided the performance of $\hat{q}$ by assessing the present value of $\hat{q}(t)$ in terms of the past behaviour of the lineaentification model over the interval $d \le i \le t$. It can be determined as

$$J\left(\hat{\theta}(t),t\right) = \sum_{i=d}^{t} E^2\left(\hat{\theta}(t),i\right),\ t \ge d \tag{5.2.2}$$

The retrospective error is then defined as

$$E\left(\hat{\theta}(t),i\right) = y_f(i) - b_0 u(i-d) - \varphi^T(i-d)\hat{\theta}(t),\ t \ge d \tag{5.2.3}$$

Define

$$E\left(\hat{\theta}(t),t\right)=\begin{bmatrix} E\left(\hat{\theta}(t),d\right) \\ \cdots \\ E\left(\hat{\theta}(t),\mathrm{t}\right) \end{bmatrix}\in R^{t-d+1}$$

$$Y(t)=\begin{bmatrix} y_f(d)-b_0u(0) \\ \cdots \\ y_f(t)-b_0u(t-d) \end{bmatrix}\in R^{t-d+1} \tag{5.2.4}$$

$$\Phi(t)=\begin{bmatrix} \varphi(0) \\ \cdots \\ \varphi(t-d) \end{bmatrix}\in R^{(t-d+1)\times(2n-1)}$$

Therefore,

$$E\left(\hat{\theta}(t),t\right)=Y(t)-\Phi(t)\hat{\theta}(t)\in R^{t-d+1} \tag{5.2.5}$$

The cost function can be written by the notations as

$$\begin{aligned} J\left(\hat{\theta}(t),t\right)&=E^T\left(\hat{\theta}(t),t\right)E\left(\hat{\theta}(t),t\right) \\ &=\left[Y(t)-\Phi(t)\hat{\theta}(t)\right]^T\left[Y(t)-\Phi(t)\hat{\theta}(t)\right] \end{aligned} \tag{5.2.6}$$

The recursive least squares estimate for $\hat{\theta}(t)$ is

$$\begin{aligned} P(t)&=P(t-1)-\frac{P(t-1)\varphi(t-d)\varphi^T(t-d)P(t-1)}{1+\varphi^T(t-d)P(t-1)\varphi(t-d)},\ P(0)>0 \\ \hat{\theta}(t)&=\hat{\theta}(t-1)+P(t)\varphi(t-d)\left[y_f(t)-b_0u(t-d)-\varphi^T(t-d)\hat{\theta}(t-1)\right] \end{aligned} \tag{5.2.7}$$

where the adaptive control law is

$$u(t)=-\frac{1}{b_0}\left[\varphi^T(t)\hat{\theta}(t)-\boldsymbol{q}^{-n+1}\boldsymbol{B_m}r(t)\right] \tag{5.2.8}$$

Define that

$$\tilde{\Theta}(k) = \begin{bmatrix} \tilde{\theta}(t) \\ \vdots \\ \tilde{\theta}(t+d-1) \end{bmatrix} \tag{5.2.9}$$

The error state vector is expressed by

$$\mathrm{X}(t) = \begin{bmatrix} x_e(t) \\ \tilde{\Theta}(t) \\ \mathrm{vec}(P(t)) \\ \vdots \\ \mathrm{vec}(P(t+d-1)) \end{bmatrix} \tag{5.2.10}$$

consisting of the model matching error states and the parameter identification error states.

The closed loop error dynamics with recursive least-squares (RLS) identification can be characterised as

$$(d+1)(2n-1) + d(2n-1)^2 \tag{5.2.11}$$

dimensional system:

$$\begin{aligned}
x_e(t+1) &= Ax_e(t) - \frac{1}{b_0} B\varphi^T \tilde{\theta}(t) \\
\tilde{\theta}(t+1) &= \tilde{\theta}(t) - P(t+1)\varphi(t-d+1)\varphi^T(t-d+1)\tilde{\theta}(t) \\
&\vdots \\
\tilde{\theta}(t+d) &= \tilde{\theta}(t+d-1) - P(t+d)\varphi(t)\varphi^T(t)\tilde{\theta}(t+d-1) \\
\mathrm{vec}\left[P(t+1)\right] &= \mathrm{vec}\left[P(t) - \frac{P(t)\varphi(t-d+1)\varphi^T(t-d+1)P(t)}{1+\varphi^T(t-d+1)P(t)\varphi(t-d+1)}\right] \\
&\vdots \\
\mathrm{vec}\left[P(t+d)\right] &= \mathrm{vec}\left[P(t+d-1) - \frac{P(t+d-1)\varphi(t)\varphi^T(t)P(t+d-1)}{1+\varphi^T(t)P(t+d-1)\varphi(t)}\right]
\end{aligned} \tag{5.2.12}$$

Note that the error system (5.2.12) is time varying since the regressor $\varphi(t)$ is the expression of the exogenous signal $u_c(t)$.

What's more, the equilibrium of the error system (5.2.12) satisfying the form $\left(0, \tilde{\theta}_q, P_q\right)$, where $P_q \geq 0$.

Remark 5.1: The future parameter error $\tilde{\theta}(t+1)$ to $\tilde{\theta}(t+d-1)$ and the future adaptation gain matrices are included in the state vector $X(k)$ to facilitate the stability analysis, not the algorithm at time t.

Remark 5.2: The equilibrium of the error system is Lyapunov stable.

State that

$$V_P(P) = \mathrm{tr} P^2 \tag{5.2.13}$$

$$\Delta V_P(P) = \mathrm{tr}\left[P^2(t+1) - P^2(t)\right] \tag{5.2.14}$$

$$V_P\left(\tilde{\theta}, P\right) = \tilde{\theta}^T P^{-1} \tilde{\theta} \tag{5.2.15}$$

and

$$\Delta V_{\tilde{\theta}}(t) = \tilde{\theta}^T(t+1) P^{-1}(t+1) \tilde{\theta}(t+1) - \tilde{\theta}^T(t) P^{-1}(t) \tilde{\theta}(t) \tag{5.2.16}$$

Accordingly, for all $t \geq 0$,

$$\begin{gathered} \Delta V_{\tilde{\theta}}(t) \leq 0 \\ \Delta V_{\tilde{\theta}}(t) = \frac{-\left[\varphi(t-d+1)\tilde{\theta}(t)\right]^2}{1+\varphi^T(t-d+1)P(t)\varphi(t-d+1)} \leq 0 \end{gathered} \tag{5.2.17}$$

In addition, $\lim_{t\to\infty} P(t)$ and $\lim_{t\to\infty} \tilde{\theta}(t)$ are existed.

Lemma 5.3: From the RLS properties, when $k > 0$

$$\frac{\left[\varphi^T(t)\tilde{\theta}(t)\right]^2}{1+\varphi^T(t)P(t)\varphi(t)} \leq \sum_{i=t}^{t+d-1} \frac{\left[\varphi^T(i-d+1)\tilde{\theta}(i)\right]^2}{1+\varphi^T(i-d+1)P(i)\varphi(i-d+1)} \tag{5.2.18}$$

Proof: As it repeats self-substitutions, it becomes

$$\begin{aligned}\tilde{\theta}(t+d-1) &= \tilde{\theta}(t) - \sum_{i=1}^{d-1} P(t+d-i)\varphi(t-i)\varphi^T(t-i)\tilde{\theta}(t+d-i-1) \\ &= \tilde{\theta}(t) - \sum_{i=t}^{t+d-2} P(i+1)\varphi(i-d+1)\varphi^T(i-d+1)\tilde{\theta}(i)\end{aligned} \tag{5.2.19}$$

Multiply $\varphi^T(t)\Big/\sqrt{1+\varphi^T(t)P(t)\varphi(t)}$ to both sides, it can be derived as

$$\begin{aligned}\frac{\varphi^T(t)\tilde{\theta}(t)}{\left[1+\varphi^T(t)P(t)\varphi(t)\right]^{\frac{1}{2}}} &= \frac{\varphi^T(t)\tilde{\theta}(t+d-1)}{\left[1+\varphi^T(t)P(t)\varphi(t)\right]^{\frac{1}{2}}} + \\ &+ \sum_{i=t}^{t+d-2}\frac{\varphi^T P(i+1)\varphi^T(i-d+1)\tilde{\theta}(i)}{\left[1+\varphi^T(t)P(t)\varphi(t)\right]^{\frac{1}{2}}}\end{aligned} \tag{5.2.20}$$

From $vec\left[P(t+d)\right]$ it can be yield that

$$\begin{aligned}\frac{\varphi^T(t)\tilde{\theta}(t)}{\left[1+\varphi^T(t)P(t)\varphi(t)\right]^{\frac{1}{2}}} &= \frac{\varphi^T(t)\tilde{\theta}(t+d-1)}{\left[1+\varphi^T(t)P(t)\varphi(t)\right]^{\frac{1}{2}}} + \\ &+ \sum_{i=1}^{t+d-2}\frac{\varphi^T P(i)\varphi(i-d+1)\varphi^T(i-d+1)\tilde{\theta}(i)}{\left[1+\varphi^T(t)P(t)\varphi(t)\right]^{\frac{1}{2}}\left[1+\varphi^T(i-d+1)P(i)\varphi(i-d+1)\right]}\end{aligned} \tag{5.2.21}$$

Apply the triangle inequality as well as $P(i) \le P(t)$ for all $i \ge t$

$$\begin{aligned}\left|\frac{\varphi^T(t)\tilde{\theta}(t)}{\left[1+\varphi^T(t)P(t)\varphi(t)\right]^{\frac{1}{2}}}\right| &\le \left|\frac{\varphi^T(t)\tilde{\theta}(t+d-1)}{\left[1+\varphi^T(t)P(t)\varphi(t)\right]^{\frac{1}{2}}}\right| + \\ &+ \sum_{i=t}^{t+d-2}\left|\frac{\varphi^T P(i)\varphi(i-d+1)\varphi^T(i-d+1)\tilde{\theta}(i)}{\left[1+\varphi^T(t)P(t)\varphi(t)\right]^{\frac{1}{2}}\left[1+\varphi^T(i-d+1)P(i)\varphi(i-d+1)\right]^{\frac{1}{2}}}\right| \\ &\times\left|\frac{\varphi^T(i-d+1)\tilde{\theta}(i)}{\left[1+\varphi^T(i-d+1)P(i)\varphi(i-d+1)\right]^{\frac{1}{2}}}\right|\end{aligned} \tag{5.2.22}$$

If apply Cauchy-Schwarz inequality, it will then become

$$\begin{aligned}\left|\frac{\varphi^T(t)\tilde{\theta}(t)}{\left[1+\varphi^T(t)P(t)\varphi(t)\right]^{\frac{1}{2}}}\right| &\leq \left|\frac{\varphi^T(t)\tilde{\theta}(t+d-1)}{\left[1+\varphi^T(t)P(t)\varphi(t)\right]^{\frac{1}{2}}}\right| + \\ &+\sum_{i=t}^{t+d-2}\left|\frac{\varphi^T(i-d+1)\tilde{\theta}(i)}{\left[1+\varphi^T(i-d+1)P(i)\varphi(i-d+1)\right]^{\frac{1}{2}}}\right| \\ &=\sum_{i=t}^{t+d-1}\left|\frac{\varphi^T(i-d+1)\tilde{\theta}(i)}{\left[1+\varphi^T(i-d+1)P(i)\varphi(i-d+1)\right]^{\frac{1}{2}}}\right|\end{aligned} \tag{5.2.23}$$

Lemma 5.4: Define

$$\begin{aligned}\prod(t)&=\operatorname{diag}\left[P^{-1}(t)\quad\cdots\quad P^{-1}(t+d-1)\right]\\ V_{\tilde{\Theta}}\left(\tilde{\Theta},\prod\right)&=\tilde{\Theta}^T\prod^{-1}\tilde{\Theta}\\ \Delta V_{\tilde{\Theta}}(t)&=\tilde{\Theta}(t+1)\prod^{-1}(t+1)\tilde{\Theta}^T(t+1)-\tilde{\Theta}^T(t)\prod^{-1}(t)\tilde{\Theta}(t)\end{aligned} \tag{5.2.24}$$

Hence,

$$\Delta V_{\tilde{\Theta}}(t)\leq\frac{-\left[\varphi(t)\tilde{\theta}(t)\right]^2}{1+\varphi^T(t)P(t)\varphi(t)} \tag{5.2.25}$$

Proof:

From

$$\begin{aligned}\tilde{\Theta}(k)&=\begin{bmatrix}\tilde{\theta}(t)\\ \vdots\\ \tilde{\theta}(t+d-1)\end{bmatrix}\\ \prod(t)&=\operatorname{diag}\left[P^{-1}(t)\quad\cdots\quad P^{-1}(t+d-1)\right]\\ \Delta V_{\tilde{\Theta}}(t)&=\tilde{\Theta}(t+1)\prod^{-1}(t+1)\tilde{\Theta}^T(t+1)-\tilde{\Theta}^T(t)\prod^{-1}(t)\tilde{\Theta}(t)\end{aligned} \tag{5.2.26}$$

The function $\Delta V_{\tilde{\Theta}}(t)$ is

$$\Delta V_{\tilde{\Theta}}(t) = \sum_{i=t+1}^{t+d} \tilde{\theta}^T(i) P^{-1}(i) \tilde{\theta}(i) - \sum_{i=k}^{t+d-1} \tilde{\theta}^T(i) P^{-1}(i) \tilde{\theta}(i) \tag{5.2.27}$$

Applying $\Delta V_{\tilde{\theta}}(t) = \frac{-\left[\varphi(t-d+1)\tilde{\theta}(t)\right]^2}{1+\varphi^T(t-d+1)P(t)\varphi(t-d+1)} \leq 0$ and $\frac{\left[\varphi^T(t)\tilde{\theta}(t)\right]^2}{1+\varphi^T(t)P(t)\varphi(t)} \leq \sum_{i=t}^{t+d-1} \frac{\left[\varphi^T(i-d+1)\tilde{\theta}(i)\right]^2}{1+\varphi^T(i-d+1)P(i)\varphi(i-d+1)}$, it becomes

$$\Delta V_{\tilde{\Theta}}(t) = -\sum_{i=t}^{t+d-1} \frac{\left[\varphi^T(i-d+1)\tilde{\theta}(i)\right]^2}{1+\varphi^T(i-d+1)P(i)\varphi(i-d+1)} \leq -\frac{\left[\varphi^T(t)\tilde{\theta}(t)\right]^2}{1+\varphi^T(t)P(t)\varphi(t)} \tag{5.2.28}$$

Lemma 5.5:

Define

$$\begin{gathered} V_{\Pi}(\Pi) = \mathrm{tr}\left[\Pi^T \Pi\right] \\ \Delta V_{\Pi}(t) = \mathrm{tr}\left[\Pi^T(t+1)\Pi(t+1)\right] - \mathrm{tr}\left[\Pi^T(t)\Pi(t)\right] \end{gathered} \tag{5.2.29}$$

Therefore,

$$\Delta V_{\Pi}(t) \leq 0, \;\; k \geq 0 \tag{5.2.30}$$

Proof:

From

$$\begin{gathered} \Delta V_{\tilde{\theta}}(t) \leq 0 \\ \Pi(t) = \mathrm{diag}\left[P^{-1}(t) \quad \cdots \quad P^{-1}(t+d-1)\right] \end{gathered} \tag{5.2.31}$$

It follows that

$$\begin{aligned} \Delta V_{\Pi}(t) &= \sum_{i=t-d+1}^{t+1} \mathrm{tr}P^2(i) - \sum_{i=t-d}^{t} \mathrm{tr}P^2(i) \\ &= -\sum_{i=t-d}^{t} \Delta V_P(t) \leq 0 \end{aligned} \tag{5.2.32}$$

Lemma 5.6:

From the previous assumption, A is asymptotically stable. Assume that $P \in \mathbb{R}^{n\times n}$ and $R \in \mathbb{R}^{n\times n}$ are positive-definite matrices, and they meet the requirement of

$$P = A^T PA + R + I \tag{5.2.33}$$

Furthermore, define

$$\sigma = \sqrt{\lambda_{\max}\left(A^T PA\right)} \tag{5.2.34}$$

Moreover, let $\mu > 0$ and assume that

$$\begin{gathered} V_{x_e}\left(x_e\right) = \ln\left(1 + \mu x_e^T P x_e\right) \\ \Delta V_{x_e}\left(t\right) = V_{x_e}\left(x_e\left(t+1\right)\right) - V_{x_e}\left(x_e\left(t\right)\right) \end{gathered} \tag{5.2.35}$$

Therefore,

$$\Delta V_{x_e}\left(t\right) \le \mu \frac{-x_e^T R x_e\left(t\right) + b_0^{-2}\left(\sigma^2 + 1\right) B^T PB\left[\varphi\left(t\right)\tilde{\theta}\left(t\right)\right]^2}{1 + \mu x_e^T\left(t\right) P x_e\left(t\right)},\ t \ge 0 \tag{5.2.36}$$

Proof:

Define that

$$F = \frac{1}{\sigma} P^{\frac{1}{2}} A, \quad G = \sigma P^{\frac{1}{2}} B, \quad \vartheta\left(x_e\right) = x_e^T P x_e \tag{5.2.37}$$

and define

$$\begin{aligned} \Delta\vartheta_{x_e}\left(t\right) &= x_e^T\left(t+1\right) P x_e\left(t+1\right) - x_e^T\left(t\right) P x_e\left(t\right) \\ &= \left[Ax_e\left(t\right) = B\frac{\varphi\left(t\right)\tilde{\theta}\left(t\right)}{b_0}\right]^T P\left[Ax_e\left(t\right) - B\frac{\varphi\left(t\right)\tilde{\theta}\left(t\right)}{b_0}\right] - x_e^T\left(t\right) P x_e\left(t\right) \end{aligned} \tag{5.2.38}$$

The explicit dependence on is omitted

$$
\begin{aligned}
\Delta\vartheta_{x_e}(t) &= x_e^T A^T P A x_e - x_e^T A^T P B b_0^{-1}\varphi^T\tilde{\theta} - b_0^{-1}\varphi^T\tilde{\theta}B^T P A x_e \\
&\quad + b_0^{-1}\varphi^T\tilde{\theta}B^T P B b_0^{-1}\varphi^T\tilde{\theta}(t) - x_e^T P x_e \\
&= x_e^T\left(A^T P A - P\right)x_e - x_e^T A^T P B b_0^{-1}\varphi^T\tilde{\theta} \\
&\quad - b_0^{-1}\varphi^T\tilde{\theta}B^T P A x_e + b_0^{-2}\left(\varphi^T\tilde{\theta}\right)B^T P B \\
&= x_e^T\left(A^T P A - P\right)x_e - x_e^T F^T G b_0^{-1}\varphi^T\tilde{\theta} \\
&\quad - b_0^{-1}\varphi^T\tilde{\theta}G^T F x_e + b_0^{-2}\left(\varphi^T\tilde{\theta}\right)B^T P B \\
&= x_e^T\left(A^T P A - P + F^T F\right)x_e - x_e^T F^T G b_0^{-1}\varphi^T\tilde{\theta} \\
&\quad - b_0^{-1}\varphi^T\tilde{\theta}G^T F x_e + b_0^{-2}\left(\varphi^T\tilde{\theta}\right)^2 B^T P B \\
&\quad - x_e^T F^T F x_e + b_0^{-2}\left(\varphi^T\tilde{\theta}\right)^2 G^T G - b_0^{-2}\left(\varphi^T\tilde{\theta}\right)^2 G^T G \\
&= x_e^T\left(A^T P A - P + F^T F\right)x_e \\
&\quad - \begin{bmatrix} x_e^T & b_0^{-1}\varphi^T\tilde{\theta} \end{bmatrix}\begin{bmatrix} F^T F & F^T G \\ G^T F & G^T G \end{bmatrix}\begin{bmatrix} x_e^T \\ b_0^{-1}\varphi^T\tilde{\theta} \end{bmatrix} \\
&\quad + \left(B^T P B + G^T G\right)b_0^{-2}\left(\varphi^T\tilde{\theta}\right)^2 \\
&\le x_e^T\left(A^T P A - P + F^T F\right)x_e + \left(B^T P B + G^T G\right)b_0^{-2}\left(\varphi^T\tilde{\theta}\right)^2
\end{aligned}
\tag{5.2.39}
$$

Note that

$$
F^T F = \frac{A^T P A}{\sigma^2} = \frac{A^T P A}{\lambda_{\max}\left(A^T P A\right)} \le \frac{\lambda_{\max}\left(A^T P A\right)I_n}{\lambda_{\max}\left(A^T P A\right)} = I_n \tag{5.2.40}
$$

From $P = A^T P A + R + I$, it is shown that

$$
A^T P A - P + F^T F \le A^T P A - P + I = -R \tag{5.2.41}
$$

As a result,

$$
x_e^T(t)\left(A^T P A - P + F^T F\right)x_e^T(t) \le -x_e^T(t) R x_e^T(t) \tag{5.2.42}
$$

which indicates that

$$
\Delta\vartheta_{x_e}(t) \le -x_e^T(t) R x_e^T(t) + \left(B^T P B + G^T G\right)b_0^{-2}\left[\varphi(t)\tilde{\theta}(t)\right]^2 \tag{5.2.43}
$$

When $G^T G = \sigma^2 B^T PB$, it becomes

$$\Delta \vartheta_{x_e}(t) \le -x_e^T(t) R x_e^T(t) + \left(\sigma^2 + 1\right) B^T PB b_0^{-2} \left[\varphi(t)\tilde{\theta}(t)\right]^2 \tag{5.2.44}$$

For $\ln x \le x - 1$, where $x > 0$,

$$\begin{aligned} \Delta \vartheta_{x_e}(t) &= \ln\left(1 + \mu \frac{\Delta \vartheta_{x_e}(t)}{1 + \mu x_e^T(t) P x_e(t)}\right) \\ &\le \mu \frac{-x_e^T(t+1) P x_e(t+1) - x_e^T P x_e(t)}{1 + \mu x_e^T(t) P x_e(t)} \\ &\le \mu \frac{-x_e^T(t) R x_e(t) - b_0^{-2}\left(\sigma^2 + 1\right) B^T PB\left[\varphi(t)\tilde{\theta}(t)\right]^2}{1 + \mu x_e^T(t) P x_e(t)} \end{aligned} \tag{5.2.45}$$

Here comes the main stability result with RLS identification.

Theorem 5.7: Suppose that the reference signal $u_c(t)$ is bounded, thus the equilibrium of the error system dynamics is Lyapunov stable, $\tilde{\theta}(t)$ and $P(t)$ converge, and $y(t) - y_m(t) \to 0$ as $k \to \infty$.

Proof:

Regard to the Lyapunov function candidate

$$V(X) = aV_{x_e}(x_e) + V_{\tilde{\Theta}}\left(\tilde{\Theta}, \Pi\right) + V_{\Pi}(\Pi) \tag{5.2.46}$$

Assume that $P \in \mathbb{R}^{n \times n}$ and $R \in \mathbb{R}^{\times n}$ are positive definite and meet the requirement of $P = A^T PA + R + I$. Let $a > 0$. Applying Lemmas 5.4, 5.5 and 5.6,

$$\begin{aligned} \Delta V(t) &= V\left(X(t+1)\right) - V\left(X(t)\right) \\ &\le a\mu \frac{-x_e^T(t) R x_e^T(t) + \left(\sigma^2 + 1\right) B^T PB\left[\varphi(t)\tilde{\theta}(t)\right]^2 \Big/ b_0^{-2}}{1 + \mu x_e^T(t) P x_e(t)} \\ &\quad - \frac{\left[\varphi(t)\tilde{\theta}(t)\right]^2}{1 + \varphi^T(t) P(t) \varphi(t)} \end{aligned} \tag{5.2.47}$$

From $\varphi(t) = M_0 x(t)$, it can be derived

$$\begin{aligned}\varphi^T(t)P(t)\varphi(t) &\le \varphi^T(t)P(0)\varphi(t)\\ &= x^T M_0^T P(0) M x\\ &= (x_e + x_m)^T M_0^T P(0) M_0 (x_e + x_m)\\ &\le x^T M_0^T P(0) M_0 x_e + x_m^T M_0^T P(0) M_0 x_m\end{aligned} \tag{5.2.48}$$

Make sure $\mu_1 > 0$ to satisfy

$$\mu_1 P > M_0^T P(0) M_0 \tag{5.2.49}$$

Recall that the command signal $u_c(t)$ is bounded and A_m is stable, it can be seen that $\beta > 0$, and

$$x_m^T(t) x_m(t) \le \beta \tag{5.2.50}$$

Substitute into (5.2.48), (5.2.49) and (5.2.50),

$$\begin{aligned}\varphi^T(t)P(t)\varphi(t) &\le \mu_1 x_e^T P x_e + \mu_1 x_m^T P x_m\\ &\le \mu_1 x_e^T P x_e + \beta\mu_1\lambda_{\max}(P)\end{aligned} \tag{5.2.51}$$

Define that

$$\begin{aligned}\mu &= \frac{\mu_1}{1+\beta\mu_1\lambda_{\max}(P)}\\ a &= \frac{b_0^2}{1+\mu_1(\sigma^2+1)\lambda_{\max}(P)}\end{aligned} \tag{5.2.52}$$

It can be shown that

$$\begin{aligned}
\Delta V &\le \frac{-\mu_1 b_0^2 x_e^T(t) R x_e(t) + \mu_1(\sigma^2+1) B^T P B\left[\varphi(t)\tilde{\theta}(t)\right]^2}{\left(1+\beta\mu_1\lambda_{\max}(P)\right)\left(1+\mu_1(\sigma^2+1)B^T P B\right)\left(1+\mu x_e^T(t) P x_e(t)\right)} \\
&\quad - \frac{\left[\varphi(t)\tilde{\theta}(t)\right]^2}{1+\varphi^T(t)P(t)\varphi(t)} \\
&\le \frac{-\mu_1 b_0^2 x_e^T(t) R x_e(t)}{\left(1+\beta\mu_1\lambda_{\max}(P)\right)\left(1+\mu_1(\sigma^2+1)B^T P B\right)\left(1+\mu x_e^T(t) P x_e(t)\right)} \\
&= -\mu a \frac{x_e^T R x_e(t)}{1+\mu x_e^T P x_e(t)}
\end{aligned} \tag{5.2.53}$$

As $V(X)$ is positive definite and radially unbounded, this function shows that the original of the error system is Lyapunov stable.

5.3 Controller design

The actual nonlinear plant model G apply root solver F = U^{-1}. In order to apply the linear direct MRAC controller design, it is laconic to construct a virtual plant model G_v as the combination of the root solver F and the actual nonlinear plant model G is considered to be a constant 1 mathematically. The whole structure shown as Figure 4.3.

A step-by-step procedure shows below to design the U-model based MRAC scheme with Lyapunov methods:

Step 1 Choose the reference model G_m and virtual plant model G_v ;

Step 2 Choose reference input signal y_r , adaptive gain g ;

Step 3 Sample the reference model output y_m and virtual output y_v ;

Step 4 Apply root solver to root the actual nonlinear plant model;

Step 5 Obtain the actual plant output y_p ;

Step 6 Compute the controller output u by the error between reference model output y_m and actual plant model output y_p;

Step 7 $t \to t+h$, back to Step 3, continue to loop.

This is the online estimation and control law for a class of smooth nonlinear dynamic systems with U-model based MRAC structure.

5.4 Case studies

Consider a nonlinear dynamic plant model G expressed by a Hammerstein model (Zhu and Guo, 2002) as the follow

$$\begin{aligned} y(t) &= 0.5y(t-1) + x(t-1) + 0.1x(t-2) \\ x(t) &= 1 + u(t) - u^2(t) + 0.2u^3(t) \end{aligned} \tag{5.4.1}$$

Convert the plant model into U-model expression as:

$$y(t) = \lambda_0(t) + \lambda_1(t)u(t-1) + \lambda_2(t)u^2(t-1) + \lambda_3(t)u^3(t-1) \tag{5.4.2}$$

where

$$\begin{aligned} \lambda_0(t) &= 0.5y(t-1) + 1 + 0.3x(t-2) \\ \lambda_1(t) &= 1 \\ \lambda_2(t) &= -1 \\ \lambda_3(t) &= 0.2 \end{aligned} \tag{5.4.3}$$

With the U-model expression, Newton-Raphson algorithm is applied to find out the root of the nonlinear system.

Assume that the virtual model is

$$G_v(z) = \frac{k_v}{z^2 - 1.3205z + 0.4966}, \quad k_v \text{ is unknown} \tag{5.4.4}$$

In MATLAB, k_v is set as 1 for initialization. For comparison, standard MRAC with MIT normalised algorithm and Lyapunov rules will be present by applying the virtual model (4.4.4) as plant model G and the whole controller scheme shown in Figure 4.3.

The reference model is

$$G_m(z) = \frac{k_m}{z^2 - 1.3205z + 0.4966}, \quad k_m = 0.1761 \tag{5.4.5}$$

Let the adaptive gain $g = 0.1$. y_r is a square wave signal. The amplitude $r = 1, 2$ and 4. The results shows in Figure 5.1 to Figure 5.6.

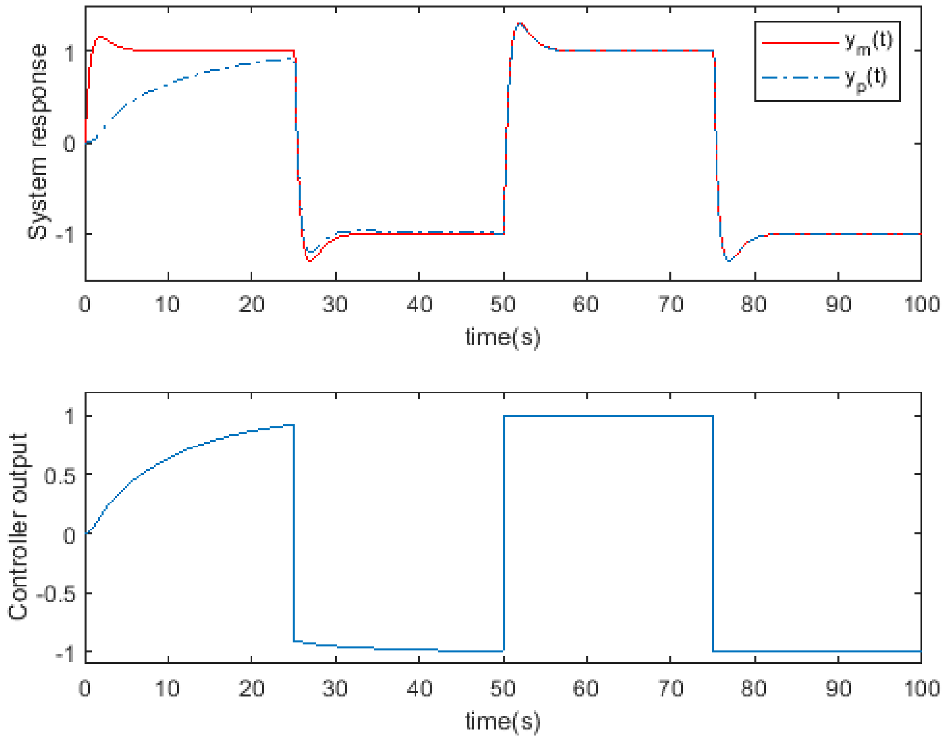


Figure 5.1 System and controller output response of standard Lyapunov-MRAC control system for virtual linear plant model when the amplitude *r=1*

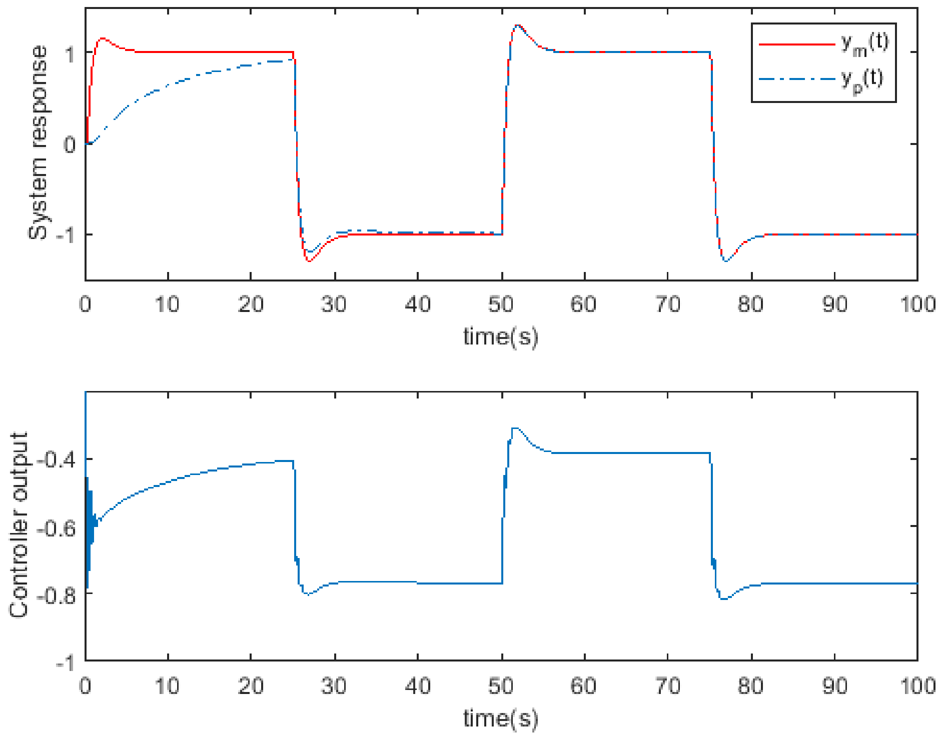


Figure 5.2 System and controller output response of U-model based Lyapunov-MRAC control system for Hammerstein model when the amplitude *r=1*

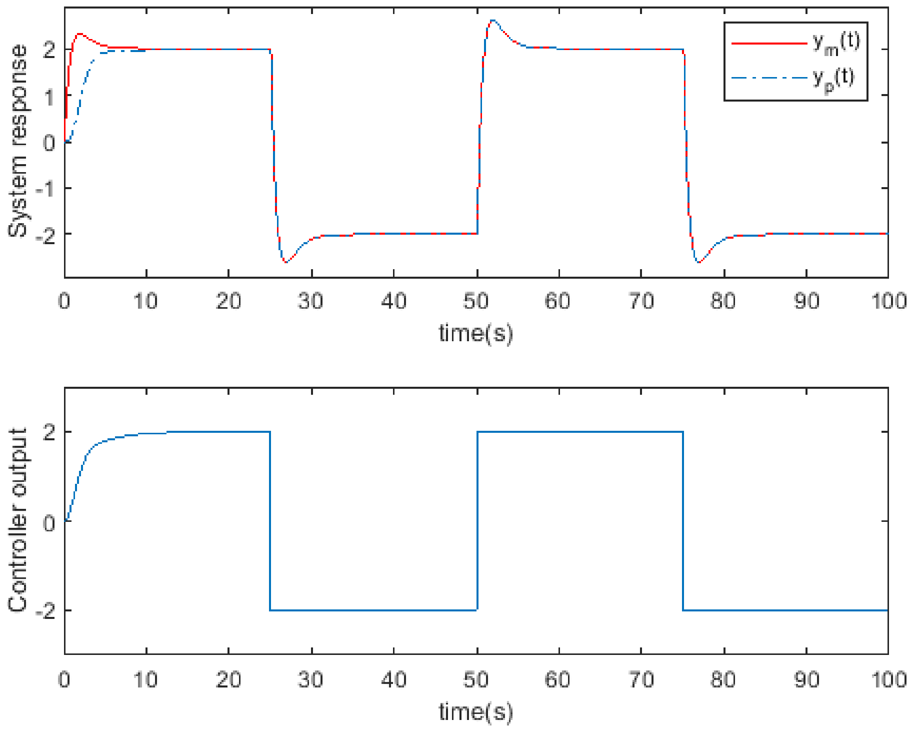


Figure 5.3 System and controller output response of standard Lyapunov-MRAC control system for virtual linear plant model when the amplitude *r=2*

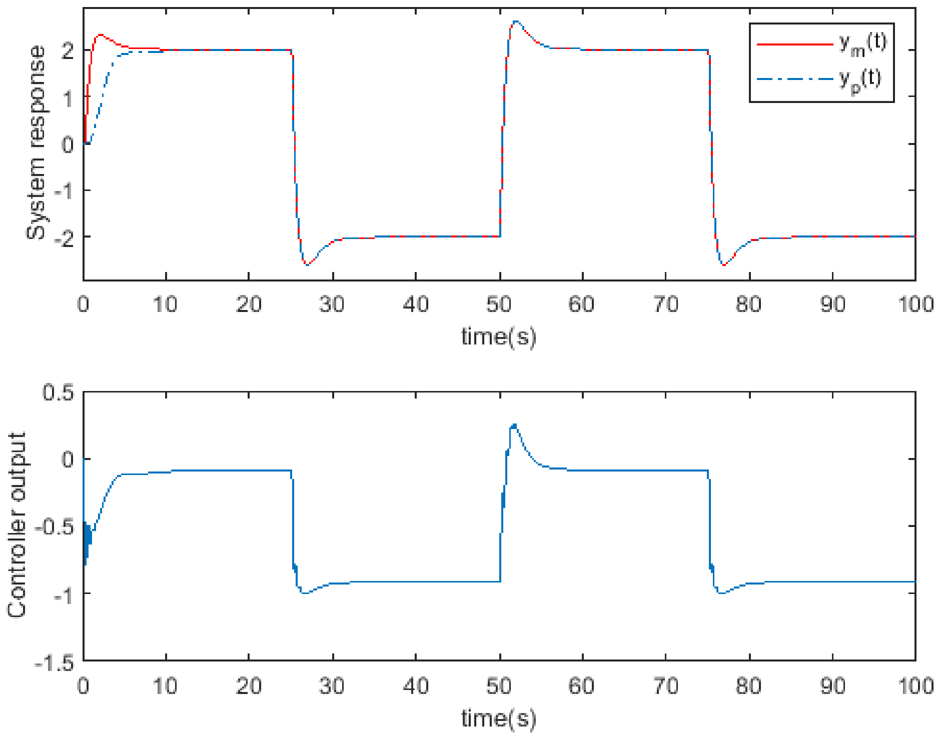


Figure 5.4 U-model based Lyapunov-MRAC control system for Hammerstein model when the amplitude *r=2*

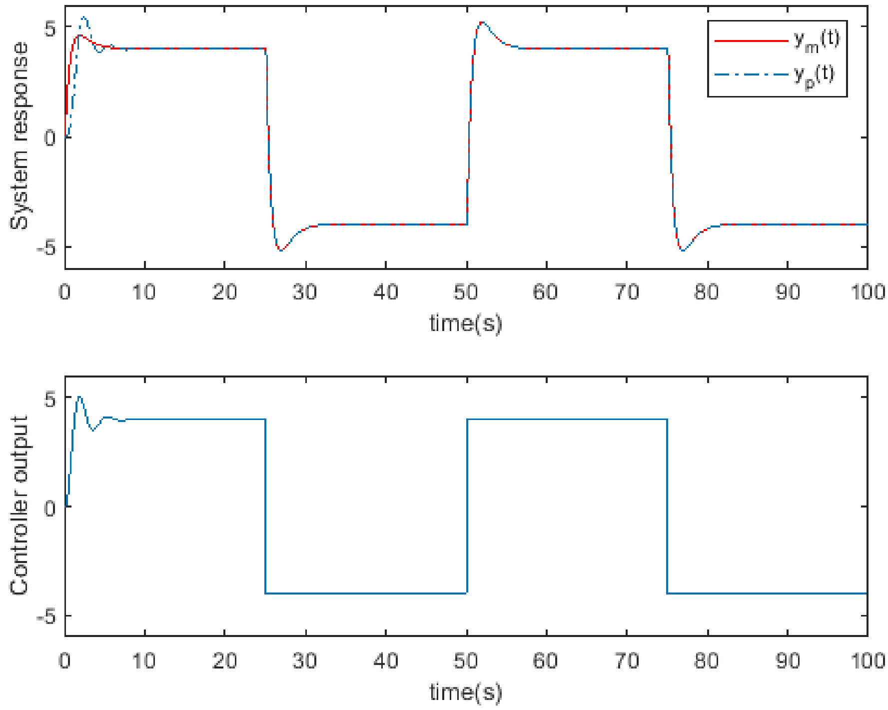


Figure 5.5 Standard Lyapunov-MRAC control system for virtual linear plant model when the amplitude *r=4*

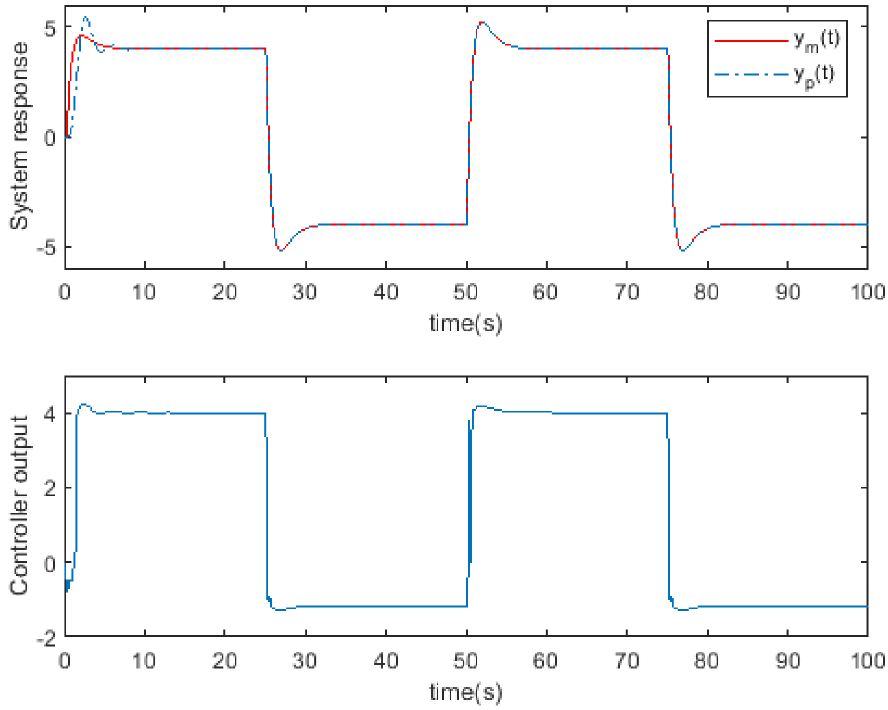


Figure 5.6 U-model based Lyapunov-MRAC control system for Hammerstein model when the amplitude *r=4*

Discussions

The amplitude of reference signals in case study are increasing from 1 to 4. If the control system doesn't apply Lyapunov algorithm, it will become unstable when the amplitude of reference signal $r = 2$ and $r = 4$. Lyapunov algorithm provides a probability of stabilised the control system for linear plant when the amplitude increased in Figure 5.1, Figure 5.3 and Figure 5.5. There is only an $< 5\%$ overshoot when the amplitude reference signal $r = 4$. Moreover, this stability could also be applied for nonlinear system in Figure 5.2, Figure 5.4 and Figure 5.6.

5.5 Summary

Similarly, both standard Lyapunov-MRAC control system and U-model based Lyapunov-MRAC control system design maintain the coherent system responses and the same reference outputs whether the amplitude changed or not. It shows that the mutative conditions will not influence the intimate relationship between standard Lyapunov-MRAC control system and the U-model based Lyapunov-MRAC control system. Moreover, it can be deduced that U-model based control system is compatible with other control system designs, not intending to improve the performance.

Chapter 6 Conclusions and further work

6.1 Conclusions

The overall aim of this PhD research was to propose the designs of U-model based control for non-minimum phase systems and adaptation of direct model reference adaptive control. Among them, pulling theorems is established for non-minimum phase features and a virtual linear system is designed to extend the U-model control system adapting direct model reference adaptive control.

In this research, a general model-independent polynomial model framework called U-model has been introduced to be the fundamental methodologies. Based on the U-model framework, the design approach for non-minimum phase system can be solved by proposed pulling theorems and the design approach for direct model reference adaptive control is simplified by using the linear virtual plant model to directly apply for the nonlinear dynamic model. Both of the designs demonstrate the generality and the compatibility of U-model framework and provide the feasibility of extending U-model framework into further classic and novel control theories/approaches.

Researchers always focus on how to demonstrate different linear control methods on nonlinear control systems for nonlinear polynomial models of the U-model based control system design. However, this study is not only showing the applicability of U-model techniques based control designs, but also the theorems independently be employed for the control problems per se. This study is full of novel ideas not limited in applications of U-model based control system designs for nonlinear plant models.

This study briefly introduces the three frameworks in control system designs, that is, model-based control system, model-free/data-driven control system and model-independent control system. In the second section, with the description of U-model structure, as one of model-independent control system, it demonstrates the superiority of model-independent control system that the design procedure is applicable to linear/nonlinear polynomial/state space model structures as it contains parallel design controllers and the dynamic inversion. It is straightforward in design without numerous miscellaneous waste/repetition if the plant model changes or proper internal disturbance. Additionally, the U-model approach complements most existing linear or nonlinear design approaches which is followed by the literature review of U-model based pole placement control system design; proposed to characterise the major essential methodologies. Also, compare to pole placement approach, which is one of the most classic control methods, U-model based pole placement control system designs are analysed to show the development of the pole placement approach with a simple linear plant model in case study.

The U-model controller enhanced design for non-minimum phase system is divided into two stages. The pulling theorems are proposed for the non-minimum phase control system characteristics and the U-model based control system is established for the whole control system with core general U-model controller---root inversion. Both zeros and poles outside the unit circle may cause the system unstable. They could be solved by the zero pulling theorems and the pole pulling theorems respectively. In the theorems, two pulling filters are established to 'pull' the system zeros and/or poles back to unit circle. They can be treated as simply addition and subtraction in mathematic expression. It is proved that the pulling theorems have proper robustness when the zeros/poles of the plant model are inside the unit two circles. The pulling theorems can be utilised independently only to solve non-minimum phase system without U-model approach. Three case studies fully explain the feasibility and simplicity of the U-model based control system with pulling theorems for unstable non-minimum phase systems.

This study also establishes a U-model based direct model reference adaptive control with MIT normalised rules for nonlinear plant model. Within the U-model platform, the controller for the nonlinear system is developed using linear direct model reference adaptive control approach. In order to implement the U-model design approach in direct model reference adaptive control system, a linear virtual plant model is established with the unknown parameters of the nonlinear system instead of the plant model in the classic direct model reference control system. With the linear virtual plant model, the direct model reference adaptive control maintains the general procedure. The plant model is the nonlinear model without unknown parameters. Through the computational experiment, it can be inspected that the system response/performance of designed U-model based direct model reference adaptive control system for nonlinear system achieves the desired requirements/targets as the same as the classic direct model reference adaptive control system for linear plant model.

The proposed U-model based direct model reference adaptive control system design approach with Lyapunov algorithm is applied to develop the controller for adaptive gain is an excessive value, or performance of is poor. The control structure is the same as the U-model based direct model reference adaptive control system with MIT normalised rules, but the estimated algorithms such as cost function is adjusted to fit the Lyapunov algorithms. From the case study, it is obviously demonstrated when the reference signal increased from 1 to 4, it shows a better performance.

Overall, this study shows that

1) The U-model based control system design within pulling theorems provides a novel method in different non-minimum phase problems. The case studies demonstrate that it has considerable robustness when the zeros/poles of the plants are in the circle with radius 2 units.
2) The U-model based direct model reference adaptive control with MIT normalised rules provides the feasibility of applying nonlinear system/plant to direct model reference adaptive control with MIT normalised rules that it has the same performance for Hammerstein model plant as the performance for linear model.
3) The U-model based direct model reference adaptive control with Lyapunov algorithms provides the feasibility of applying nonlinear system/plant to direct model reference adaptive control with Lyapunov algorithms when the condition changes in the design of with MIT normalised rules.
4) This study shows through the analytical process and computational experiments to prove the controller output, performance and response of U-model based control systems.
5) This study provides novel methodologies of interdisciplinary research programme and innovative control approaches in control theory.

6.2 Proposed further research

It has been nearly two decades from the first time the U-model based control system design been proposed (Zhu and Guo, 2002). There is a growing number of researchers pay attention and attempt to apply U-model framework.

There are many potential analytical methods of the U-model framework to be summarised in this section. The potential expansion of the present U-model framework study can be summarised as follows.

- The U-model based control system for non-minimum phase systems in this study demonstrates proper robustness. There may still be requirement of robust control algorithms and more stability analysis methods/theorems should be studied to give more powerful evidence for U-model based control system for non-minimum phase systems.
- Some of nonlinear system are also unstable and/or non-minimum phase. This study may enlighten the next stage of U-model based control approaches for nonlinear unstable non-minimum phase systems.
- Recently, some researchers proposed adaptive U-control framework (Du, Wu and Zhu, 2012; Zhu et al., 2018). Compared with the classic adaptive control scheme, U-adaptive control does not require controller design in every updating step. It only updates the plant model while the controller is fixed/changed. As the same as combination of U-model framework and adaptive control, U-adaptive control inspires the U-model based direct model reference adaptive control to improve the update rules.
- The U-model based direct model reference control system is established, the U-model based indirect model reference control system can be a break new field for further work.
- This study is almost based on computational experiments/simulations, the industrial application should be bench tested the applicability of the methodologies.
- The uncertainty of the computational experimental systems is appropriate small. It is a challenge for future to consider the large uncertainty of the systems and robustness of the controller design.

Reference list

Akhtar, S. and Bernstein, D.S. (2005) Lyapunov-stable discrete-time model reference adaptive control. *International Journal of Adaptive Control and Signal Processing*. 19 (10), pp. 745–767.

Albertos, P. and Mareels, I. (2010) Control Design. In: *Feedback and Control for Everyone*. Springer. pp. 249–274.

Ampsefidis, A.J., Białasiewicz, J.T. and Wall, E.T. (1993) Lyapunov design of a new model reference adaptive control system using partial a priori information. *Kybernetika*. 29 (4), pp. 339–350.

Aracil, J. and Gordillo, F. (2004) Describing function method for stability analysis of PD and PI fuzzy controllers. *Fuzzy Sets and Systems*. 143 (2), pp. 233–249.

Åström, K.J., Hagander, P. and Sternby, J. (1984) Zeros of sampled systems. *Automatica*. 20 (1), pp. 31–38. doi:10.1016/0005-1098(84)90062-1.

Åström, K.J. and Wittenmark, B. (2013) *Adaptive control*. Courier Corporation.

Bachelier, O., Bosche, J. and Mehdi, D. (2006) On pole placement via eigenstructure assignment approach. *IEEE Transactions on Automatic Control*. 51 (9), pp. 1554–1558.

Bachelier, O. and Mehdi, D. (2008) Non-iterative pole placement technique: A step further. *Journal of the Franklin Institute*. 345 (3), pp. 267–281.

Baker, F. and Thennadil, S.N. (2018) Investigation of performance of an adaptive kalman filter for state estimation of a gas phase reaction in a CSTR. *Chemeca 2018*. pp. 79.

Billings, S.A. (2013) *Nonlinear system identification: NARMAX methods in the time, frequency, and spatio-temporal domains*. John Wiley & Sons.

Bragin, V.O., Vagaitsev, V.I., Kuznetsov, N. V and Leonov, G.A. (2011) Algorithms for finding hidden oscillations in nonlinear systems. The Aizerman and Kalman conjectures and Chua's circuits. *Journal of Computer and Systems Sciences International*. 50 (4), pp.

511.

Brogan, W.L. (1974) *Modern control theory*. Pearson education india.

Broussard, J. and O'Brien, M. (1980) Feedforward control to track the output of a forced model. *IEEE Transactions on Automatic Control*. 25 (4), pp. 851–853.

Butterworth, J.A., Pao, L.Y. and Abramovitch, D.Y. (2008) Architectures for tracking control in atomic force microscopes. In: *IFAC Proceedings Volumes (IFAC-PapersOnline)*. 2008 doi:10.3182/20080706-5-KR-1001.0876.

Champetier, C. and Magni, J.-F. (1991) On eigenstructure assignment by gain output feedback. *SIAM Journal on Control and Optimization*. 29 (4), pp. 848–865.

Chateauneuf, A. (2008) Principles of reliability-based design optimization. *Structural Design Optimization Considering Uncertainties*. pp. 3–31.

Chen, B.M., Lee, T.H., Peng, K. and Venkataramanan, V. (2003) Composite nonlinear feedback control for linear systems with input saturation: Theory and an application. *IEEE Transactions on Automatic Control*. 48 (3), pp. 427–439.

Choi, S.-K., Canfield, R.A. and Grandhi, R. V (2007) Reliability-Based Structural Optimization. *Reliability-based Structural Design*. pp. 153–202.

Chong, E.K.P. and Zak, S.H. (2011) *An Introduction to Optimization: Third Edition*.

Csanádi, B., Tar, J.K. and Bitó, J.F. (2020) Novel Model Reference Adaptive Control Designed by a Lyapunov Function That is Kept at Low Value by Fixed Point Iteration. In: *Recent Advances in Intelligent Engineering*. Springer. pp. 129–137.

Daouk, S., Louf, F., Dorival, O., Champaney, L. and Audebert, S. (2015) Uncertainties in structural dynamics: overview and comparative analysis of methods. *Mechanics & Industry*. 16 (4), pp. 404.

Davison, E. and Wang, S. (1975) On pole assignment in linear multivariable systems using output feedback. *IEEE Transactions on Automatic Control*. 20 (4), pp. 516–518.

Do, M.T., Man, Z., Jin, J., Zhang, C., Zheng, J. and Wang, H. (2016) Sliding mode learning control of non-minimum phase nonlinear systems. *International Journal of Robust and Nonlinear Control*. 26 (11), pp. 2281–2298.

Dorf, R.C. and Bishop, R.H. (2011) *Modern control systems*. Pearson.

Du, W., Wu, X. and Zhu, Q. (2012) Direct design of a U-model-based generalized predictive controller for a class of non-linear (polynomial) dynamic plants. *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering*. 226 (1), pp. 27–42.

Farajzadeh-D, M.-G., Hosseini Sani, S.K. and Akbarzadeh, A. (2019) Performance enhancement of model reference adaptive control through normalized Lyapunov design. *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering*. pp. 0959651818822925.

Fliess, M. and Join, C. (2013) Model-free control. *International Journal of Control*. 86 (12), pp. 2228–2252.

Franklin, G.F., Powell, J.D. and Emami-Naeini, A. (2002) *Feedback Control of Dynamic Systems*.

Freeman, C.T., Lewin, P.L. and Rogers, E. (2005) Experimental evaluation of iterative learning control algorithms for non-minimum phase plants. *International Journal of Control*. 78 (11), pp. 826–846.

Ganesan, M., Ezhilarasi, D. and Jijo, B. (2017) Hybrid model reference adaptive second order sliding mode controller for automatic train operation. *IET Control Theory & Applications* [online]. 11 (8), pp. 1222–1233. Available from: http://digital-library.theiet.org/content/journals/10.1049/iet-cta.2016.1275doi:10.1049/iet-cta.2016.1275.

García, P., Albertos, P. and Hägglund, T. (2006) Control of unstable non-minimum-phase delayed systems. *Journal of Process Control*. 16 (10), pp. 1099–1111. doi:10.1016/j.jprocont.2006.06.007.

Gerald, C.F. (2004) *Applied numerical analysis*. Pearson Education India.

Goodwin, G.C., Graebe, S.F. and Salgado, M.E. (2001) *Control system design*. Prentice Hall New Jersey.

Grayson, L.P. (1965) The status of synthesis using Lyapunov's method. *Automatica*. 3 (2), pp. 91–121.

Gross, E., Tomizuka, M. and Messner, W. (1994) Cancellation of discrete time unstable zeros by feedforward control. *Journal of Dynamic Systems, Measurement, and Control*. 116 pp. 33–38. doi:10.1115/1.2900678.

Guo, J. and Tao, G. (2015) A discrete-time multivariable MRAC scheme applied to a nonlinear aircraft model with structural damage. *Automatica*. 53 pp. 43–52.

Gupta, A. and Yan, D.S. (2016) *Mineral processing design and operations: an introduction*. Elsevier.

Haack, B. and Tomizuka, M. (1991) The Effect of Adding Zeroes to Feedforward Controllers. *Journal of Dynamic Systems Measurement and Control*. 113 (March), pp. 6. doi:10.1115/1.2896362.

Hsu, L., Minhoto Teixeira, M.C., Costa, R.R. and Assunção, E. (2015) Lyapunov design of multivariable MRAC via generalized passivation. *Asian Journal of Control*. 17 (5), pp. 1484–1497.

Jain, P. and Nigam, M.J. (2013) Design of a model reference adaptive controller using modified MIT rule for a second order system. *Advance in Electronic and Electric Engineering*. 3 (4), pp. 477–484.

Kersting, S. and Buss, M. (2017) Direct and Indirect Model Reference Adaptive Control for Multivariable Piecewise Affine Systems. *IEEE Transactions on Automatic Control*. 62 (11), pp. 5634–5649. doi:10.1109/TAC.2017.2690060.

Khaddage, F., Müller, W. and Flintoff, K. (2016) Advancing mobile learning in formal and informal settings via mobile app technology: Where to from here and how? *Journal of Educational Technology & Society*. 19 (3), pp. 16–26.

Kimura, H. (1975) Pole assignment by gain output feedback. *IEEE Transactions on automatic*

Control. 20 (4), pp. 509–516.

Korn, G.A. and Theresa, M.K. (1972) *Electronic analog and hybrid computers*. New York: McGraw-Hill New York.

Landau, I.D., Lozano, R. and M'Saad, M. (1998) *Adaptive control*. Springer New York.

Leigh, J.R. (2012) *Control theory: a guided tour. 3rd ed.* London: The Institution of Engineering and Technology.

Liu, T., Zhang, W. and Gu, D. (2005) Analytical design of two-degree-of-freedom control scheme for open-loop unstable processes with time delay. *Journal of Process Control*. 15 (5), pp. 559–572.

Liu, Z., Zhao, L., Zhu, Q. and Gao, C. (2017) Design of sliding-mode observer for a class of uncertain neutral stochastic systems. *International Journal of Systems Science*. 48 (7), pp. 1380–1394. doi:10.1080/00207721.2016.1258597.

Lutfy, O.F. (2014) Wavelet neural network model reference adaptive control trained by a modified artificial immune algorithm to control nonlinear systems. *Arabian Journal for Science and Engineering*. 39 (6), pp. 4737–4751.

Mabius, L. and Kaufman, H. (1975) An implicit adaptation algorithm for a linear model reference control system. In: *1975 IEEE Conference on Decision and Control including the 14th Symposium on Adaptive Processes*. 1975 IEEE. pp. 864–865.

Mani, G., Sivaraman, N. and Kannan, R. (2018) Visual Servoing Based Model Reference Adaptive Control with Lyapunov Rule for a Ball on Plate Balancing System. In: *2018 International Conference on Intelligent and Advanced System (ICIAS)*. 2018 IEEE. pp. 1–6.

Marlin, T.E. (1995) *Process control: designing processes and control systems for dynamic performance*. McGraw-Hill New York.

Middleton, R.H. and Goodwin, G.C. (1990) *Digital Control and Estimation: A Unified Approach (Prentice Hall Information and System Sciences Series)*. Prentice Hall Englewood Cliffs, NJ.

Mirkin, B., Gutman, P.-O. and Shtessel, Y. (2014) Decentralized continuous MRAC with local asymptotic sliding modes of nonlinear delayed interconnected systems. *Journal of the Franklin Institute*. 351 (4), pp. 2076–2088.

Mirkin, B., Gutman, P.-O., Shtessel, Y. and Edwards, C. (2011) Continuous decentralized MRAC with sliding mode of nonlinear delayed dynamic systems. *IFAC World Congr., Milano, Italy*.

Mirkin, B., Gutman, P.O. and Shtessel, Y. (2012) Coordinated decentralized sliding mode MRAC with control cost optimization for a class of nonlinear systems. *Journal of the Franklin Institute*. 349 (4), pp. 1364–1379.

Mohammadi, M. and Shahri, A.M. (2013) Adaptive nonlinear stabilization control for a quadrotor UAV: Theory, simulation and experimentation. *Journal of Intelligent & Robotic Systems*. 72 (1), pp. 105–122.

Moore, B.C. (1975) On the flexibility offered by state feedback in multivariable systems beyond closed loop eigenvalue assignment. In: *1975 IEEE Conference on Decision and Control including the 14th Symposium on Adaptive Processes*. 1975 IEEE. pp. 207–214.

Mottershead, J.E. and Ram, Y.M. (2006) Inverse eigenvalue problems in vibration absorption: passive modification and active control. *Mechanical Systems and Signal Processing*. 20 (1), pp. 5–44.

Mottershead, J.E. and Ram, Y.M. (2007) Receptance method in active vibration control. *AIAA journal*. 45 (3), pp. 562–567.

Pal, M., Sarkar, G., Barai, R.K. and Roy, T. (2015) *Design of adaptive two-degree-of-freedom controller for inversion based reference input tracking of non-minimum phase system*. Michael Faraday IET International Summit 2015. Kolkata, India.

Parks, P. (1966) Liapunov redesign of model reference adaptive control systems. *IEEE Transactions on Automatic Control*. 11 (3), pp. 362–367.

Patil, M., Bandyopadhyay, B., Chalanga, A. and Arya, H. (2018) Output Tracking of Nonminimum-Phase Systems via Reduced-Order Sliding-Mode Design. *IEEE/ASME Transactions on Mechatronics*. 23 (4), pp. 1981–1992.

Pawar, R.J. and Parvat, B.J. (2015) Design and Implementation of MRAC and Modified MRAC technique for Inverted Pendulum. In: *International Conference on Pervasive Computing (ICPC)*. 2015

Phillips, C.L. and Habor, R.D. (2011) *Feedback control systems, 5th ed*. Boston. Pearson Education, Inc.

Prakash, R. and Anita, R. (2012) Modeling and simulation of fuzzy logic controller-based model reference adaptive controller. *International Journal of Innovative Computing, Information and Control*. 8 (4), pp. 2533–2550.

Prakash, R. and Anita, R. (2011) Neuro-PI controller based model reference adaptive control for nonlinear systems. *International Journal of Engineering, Science and Technology*. 3 (6), pp. 44–60.

Qiu, J., Delshad, I.S., Zhu, Q., Nibouche, M. and Yao, Y. (2017) A U-model based controller design for non-minimum phase systems: Application to Boeing 747 altitude-hold autopilot. In: *2017 9th International Conference on Modelling, Identification and Control (ICMIC)*. 2017 IEEE. pp. 122–127.

Qiu, J., Zhu, G., Zhu, Q., Nibouche, M. and Yao, Y. (2016a) Comparative Study on Pole Placement Controller Design with U-Model Approach & Classical Approach. *TCCT Control Conference (CCC), 2016 35th Chinese*. pp. 10436–10441.

Qiu, J., Zhu, Q., Mokhtar, N. and Yao, Y. (2016b) U-model Enhanced Control of Nonminimum-Phase Dynamic Systems. In: *IEEE Modelling, Identification and Control (ICMIC), 2016 8th International Conference*. 2016 pp. 648–653.

Rajani, S.H., Krishna, B.M. and Nair, U. (2018) Adaptive and Modified Adaptive Control for Pressure Regulation in a Hypersonic Wind Tunnel. *International Journal of Modelling, Identification and Control (IJMIC)*. 29 (1), pp. 78–87.

Ram, Y.M. and Mottershead, J.E. (2013) Multiple-input active vibration control by partial pole placement using the method of receptances. *Mechanical Systems and Signal Processing*. 40 (2), pp. 727–735.

Rigney, B.P., Pao, L.Y. and Lawrence, D.A. (2010) Nonminimum phase adaptive inverse control for settle performance applications. *Mechatronics*. 20 (1), pp. 35–44. doi:10.1016/j.mechatronics.2009.06.007.

Sharifi, M., Behzadipour, S. and Vossoughi, G. (2014) Nonlinear model reference adaptive impedance control for human–robot interactions. *Control Engineering Practice*. 32 pp. 9–27.

Slotine, J.-J.E. and Li, W. (1991) *Applied nonlinear control*. Prentice hall Englewood Cliffs, NJ.

Srinathkumar, S. (1978) Eigenvalue/eigenvector assignment using output feedback. *IEEE Transactions on Automatic Control*. 23 (1), pp. 79–81.

Stebbins, S., Hickman, M., Kim, J. and Vu, H.L. (2018) *Balancing Computational Requirements with Performance in Model Predictive Traffic Control*.

Sun, X., Shi, Z., Chen, L. and Yang, Z. (2016) Internal model control for a bearingless permanent magnet synchronous motor based on inverse system method. *IEEE Transactions on Energy Conversion*. 31 (4), pp. 1539–1548.

Tan, W., Marquez, H.J. and Chen, T. (2003) IMC design for unstable processes with time delays. *Journal of Process Control*. 13 (3), pp. 203–213.

Tang, P.L., De Silva, C.W. and Poo, A.-N. (2001) Development of model-referenced fuzzy adaptive control. In: *Proceedings Joint 9th IFSA World Congress and 20th NAFIPS International Conference (Cat. No. 01TH8569)*. 2001 IEEE. pp. 1856–1861.

Tehrani, M.G., Elliott, R.N.R. and Mottershead, J.E. (2010) Partial pole placement in structures by the method of receptances: theory and experiments. *Journal of Sound and Vibration*. 329 (24), pp. 5017–5035.

Tomizuka, M. (1987) Zero phase error tracking algorithm for digital control. *Journal of Dynamic Systems, Measurement, and Control*. 109 (December), pp. 65–68. doi:10.1115/1.3143822.

Tsai, H.Y., Huang, K.C., Chang, H.C., Yeh, J.L.A. and Chang, C.H. (2014) A Noncontact Skin Oxygen-Saturation Imaging System for Measuring Human Tissue Oxygen Saturation.

IEEE Transactions on Instrumentation and Measurement. 63 (11), pp. 2620–2631. doi:10.1109/TIM.2014.2312512.

Wang, L. and Su, J. (2015) Disturbance rejection control for non-minimum phase systems with optimal disturbance observer. *ISA Transactions*. 57 pp. 1–9. doi:10.1016/j.isatra.2014.08.003.

Wang, Q.-G., Zhang, Z., Astrom, K.J., Zhang, Y. and Zhang, Y. (2008) Guaranteed dominant pole placement with PID controllers. *IFAC Proceedings Volumes*. 41 (2), pp. 5842–5845.

Wang, X. (1992) Pole placement by static output feedback. *J. of Math. Systems, Estimation and Control*. 2 pp. 205–218.

Wang, X.A. (1996) Grassmannian, central projection, and output feedback pole assignment of linear systems. *IEEE Transactions on Automatic Control*. 41 (6), pp. 786–794.

Wen, J.T. and Potsaid, B. (2004) An experimental study of a high performance motion control system. In: *American Control Conference*. 2004 pp. 5158–5163.

Winsor, C. and Roy, R. (1968) Design of model reference adaptive control systems by Liapunov's second method. *IEEE Transactions on Automatic Control*. 13 (2), pp. 204.

Wu, X., Liu, L., Zhu, Q., Du, W., Wang, B. and Zhang, J. (2011) U-model-based adaptive control for a class of stochastic non-linear dynamic plants with unknown parameters. *International Journal of Modelling, Identification and Control*. 13 (3), pp. 135–143.

Xie, J. and Zhao, J. (2017) *Model reference adaptive control for nonlinear switched systems under asynchronous switching*. (January 2016), pp. 3–22. doi:10.1002/acs.2666.

Xu, J.-X. and Tan, Y. (2003) *Linear and nonlinear iterative learning control*. Springer.

Zhang, P. (2010) *Advanced industrial control technology*. William Andrew.

Zhao, L., Wang, J. and Zhang, W. (2016) Smooth Adaptive Internal Model Control Based on Model for Nonlinear Systems with Dynamic Uncertainties. *Mathematical Problems in Engineering*.

Zhu, Q. and Guo, L.Z. (2002) A pole placement controller for non-linear dynamic plants.

Proceedings of the I MECH E Part I Journal of Systems. 216 (6), pp. 467–476. doi:10.1243/095965102321042227.

Zhu, Q., Liu, L., Zhang, W. and Li, S. (2018) Control of complex nonlinear dynamic rational systems. *Complexity*. 2018 .

Zhu, Q., Qiu, J., Delshad, I.S., Nibouche, M. and Yao, Y. (2020a) U-model based control of non-minimum phase systems. *(to be submitted)*.

Zhu, Q., Wang, Y., Zhao, D., Li, S. and Billings, S.A. (2015) Review of rational (total) nonlinear dynamic system modelling, identification, and control. *International journal of systems science*. 46 (12), pp. 2122–2133.

Zhu, Q., Zhang, W., Mobayen, S., Yan, H., Qiu, J. and Narayan, P. (2020b) U-Control: A model-independent control system design framework. *(to be submitted)*.

Zhu, Q.M., Zhao, D.Y. and Zhang, J. (2016) A general U-block model-based design procedure for nonlinear polynomial control systems. *International Journal of Systems Science*. 47 (14), pp. 3465–3475.

Zuo, X., Liu, J., Wang, X. and Liang, H. (2017) Adaptive PID and Model Reference Adaptive Control Switch Controller for Nonlinear Hydraulic Actuator. *Mathematical Problems in Engineering*. 2017 . Volume 2017.

Appendix A User Manual

Introduction

This program aims to demonstrate and simulate the study of the designed U-Model based control systems for non-minimum phase and adaptive control system. MATLAB simulation program/Simulink can test the performance response of the control system. The simulation program procedure includes all the simulation procedures in this thesis.

Guide

Several steps should be done to run this program and to discover the performance of the U-model based control system design.

For .m file:

- Run the MATLAB software;
- Change the direction point to the related folder path and add to the MATLAB path;
- Run the *.m file and the simulation results will appear automatically.

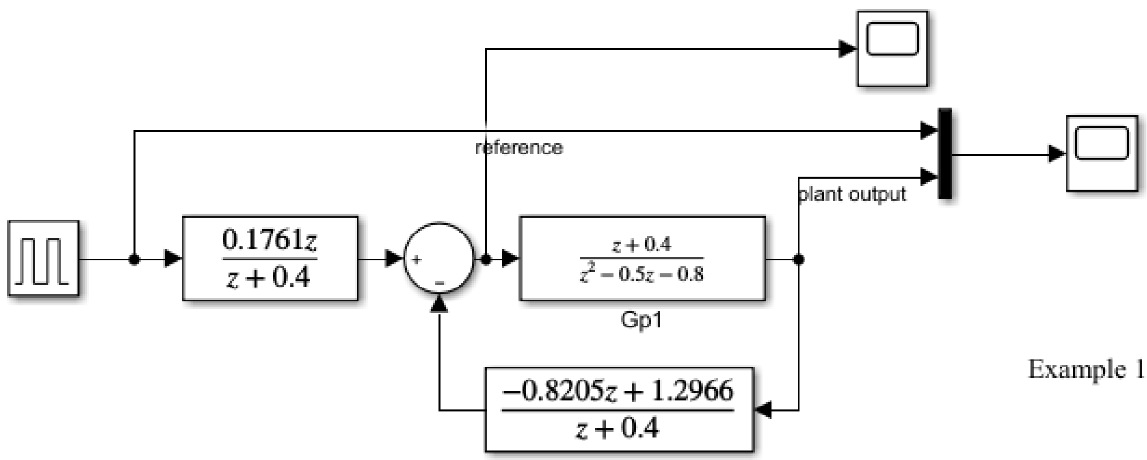
For .slx file:

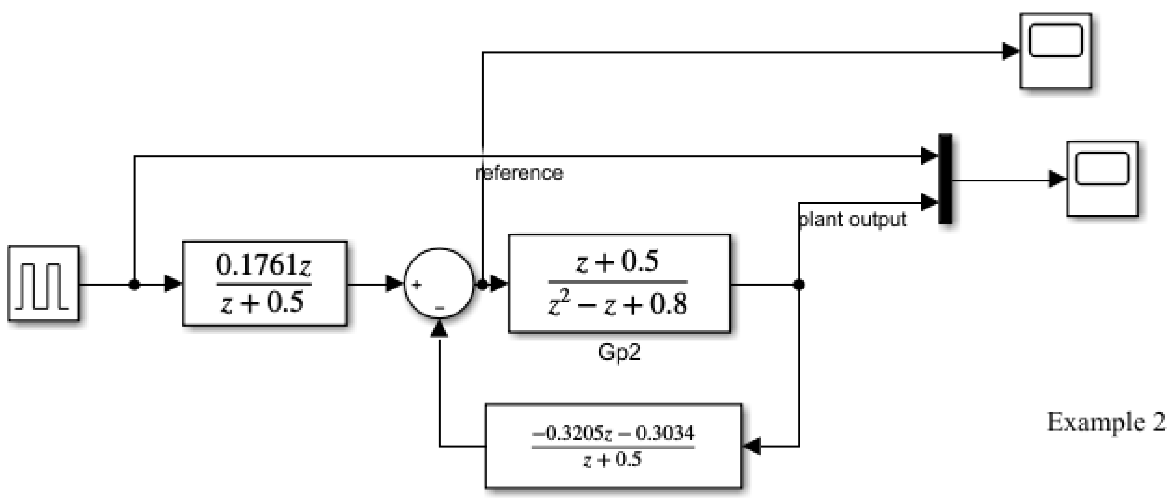
- Run the MATLAB software;
- Create a new .slx file from click the SIMULINK button in Home tool box.
- Select Blank Model.
- Click on Library Browser and drag the corresponding block to the model.
- Use Run button and double click the scope. The simulation results will appear.

Appendix B MATLAB Programs/Simulink

A. MATLAB for section 2.3.3

The .slx file for pole placement approach is shown below.





The .m file for example 1 by U-model:

```
clc, clear
% Step 1: Initialisation
ns=400; %lenght of sample
ite=1*ones(1,75);itd=1*ite(1:75);
w(1:75)=itd;%setpoint sequence

temp(1:75)=1*ite;temp(76:150)=0*ite;
temp(151:225)=0*ite;temp(226:300)=1*ite;
```

```
while length(w)<ns
      w=[w,temp];
end
w=w(1:ns);
%u(1,2)=0.01.*(2.*rand-1);
u=zeros(1,2);
y=zeros(1,3); x=y; U=y;


% Step 2: Specify desired polynomials Ac and R, compute T

deg_Ac=2; %degree of desired polynomial
Ac=[1, -1.3205, 0.4966];
R=[1, -0.9, 0.009];
T=sum(Ac);
deg_R=deg_Ac;

% Step 3: Determine polynomial S by solving Diophantine equation

deg_S=1; %degree of desired polynomial S=
S=Ac(deg_S+1:deg_Ac+1)-R(deg_S+1:deg_Ac+1);

% Step 4: Generate alpha and plant output,
 for t=3:ns
    lamda_0=0.5*y(t-1)+0.8*y(t-2)+0.4*u(t-2);
    alpha=[lamda_0, 1];
    u_temp = [1, u(t-1)];
    y(t)=alpha*u_temp';


% Step 5: Controller design for U(t-1)
    U_temp = [U(t-1), U(t-2)];
    y_temp = [y(t), y(t-1)];
    w_temp = w(t);
    R_temp = R(2:length(R));

    U(t) = -R_temp*U_temp'+T*w_temp-S*y_temp';
lamda_0_new=0.5*y(t)+0.8*y(t-1)+0.4*u(t-1);
    u(t) = U(t)-lamda_0_new;
end

% Step 6: Display simulation results
t=1:ns;
figure(1)
plot(t,w, '--', t, y, '-')
xlabel('time(s)')
ylabel('System response')
% text(0.69,0.9, '- - - reference', 'sc')
% text(0.69,0.8, '----- plant output', 'sc')

figure(2)
plot(t,u),xlabel('time(s)'),ylabel('Controller output')
```

The .mfile for example 2 by U-model:

```
clc, clear
% Step 1: Initialisation
ns=400; %lenght of sample
ite=1*ones(1,75);itd=1*ite(1:75);
w(1:75)=itd;%setpoint sequence

temp(1:75)=1*ite;temp(76:150)=0*ite;
temp(151:225)=0*ite;temp(226:300)=1*ite;
% ns=400; %lenght of sample
%
% ite=2.*ones(1,20);itd=ite(1:10);
%
% w(1:10)=itd; %setpoint sequence
% temp(1:20)=ite;temp(21:40)=3*ite;%1.*ite;
% temp(41:60)=ite;temp(61:80)=0.*ite;
while length(w)<ns
      w=[w,temp];
end
w=w(1:ns);
%u(1,2)=0.01.*(2.*rand-1);
u=zeros(1,2);
y=zeros(1,3); x=y; U=y;


% Step 2: Specify desired polynomials Ac and R, compute T

deg_Ac=2; %degree of desired polynomial
Ac=[1, -1.3205, 0.4966];
R=[1, -0.9, 0.009];
T=sum(Ac);
deg_R=deg_Ac;

% Step 3: Determine polynomial S by solving Diophantine equation

deg_S=1; %degree of desired polynomial S=
S=Ac(deg_S+1:deg_Ac+1)-R(deg_S+1:deg_Ac+1);

% Step 4: Generate alpha and plant output,
 for t=3:ns
    lamda_0=y(t-1)-0.8*y(t-2)+0.5*u(t-2);
    alpha=[lamda_0, 1];
    u_temp = [1, u(t-1)];
    y(t)=alpha*u_temp';


% Step 5: Controller design for U(t-1)
    U_temp = [U(t-1), U(t-2)];
    y_temp = [y(t), y(t-1)];
    w_temp = w(t);
    R_temp = R(2:length(R));
```

```
    U(t) = -R_temp*U_temp'+T*w_temp-S*y_temp';
lamda_0_new=y(t)-0.8*y(t-1)+0.5*u(t-1);
    u(t) = U(t)-lamda_0_new;
end

% Step 6: Display simulation results
t=1:ns;
figure(1)
plot(t,w, '--', t, y, '-')
xlabel('time(s)')
ylabel('System response')
% text(0.69,0.9, '- - - reference', 'sc')
% text(0.69,0.8, '----- plant output', 'sc')

figure(2)
plot(t,u),xlabel('time(s)'),ylabel('Controller output')
```

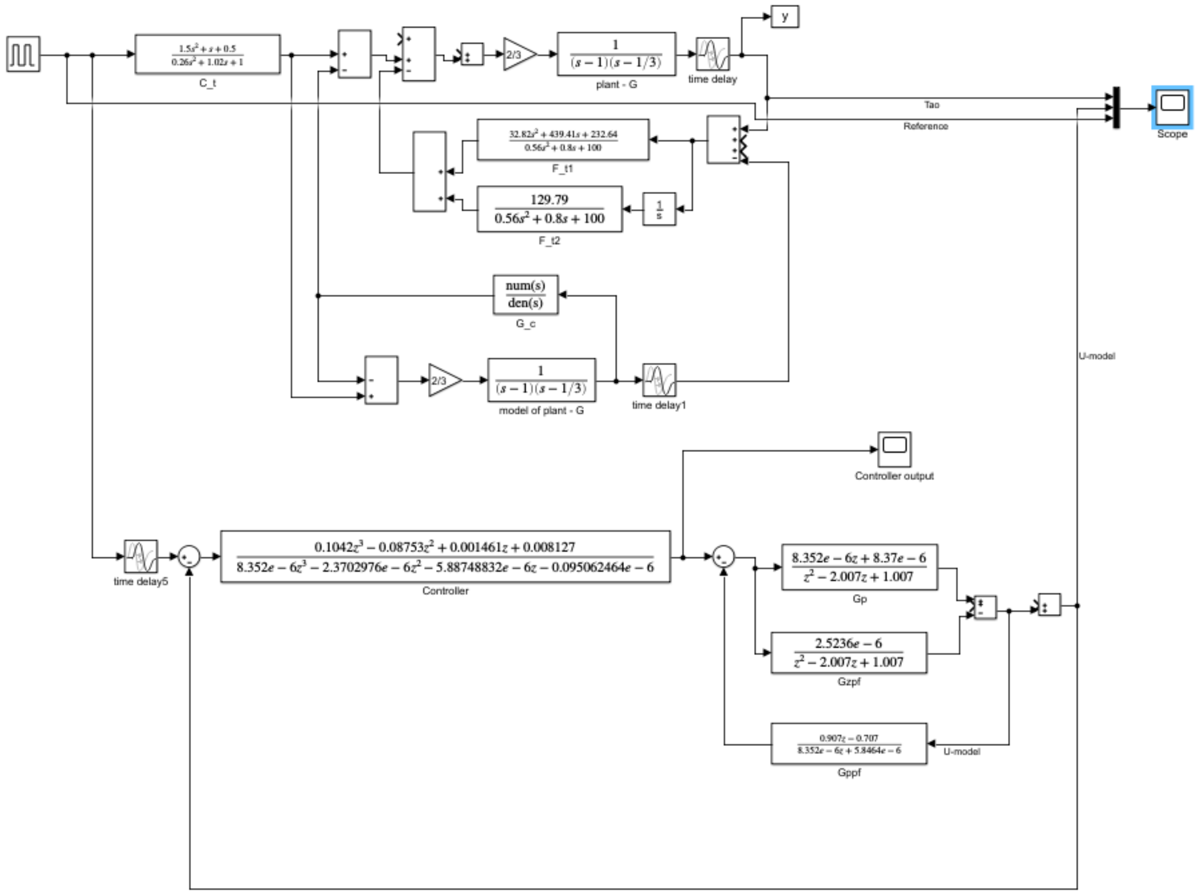
B. MATLAB code for section 3.3.1

```
clc, clear;
% Start date:06/05/2017
% Update date:14/11/2017

% Example from Control of unstable non-minimum-phase rotary mechanical system
systems
%
%
%
% Sample time: 0.005 seconds
% Discrete-time transfer function.
%
% Step 1: Initialisation
%
% Normal input:
% ------------------------------------------
% ns=300; %lenght of sample
% ite=1*ones(1,75);itd=1*ite(1:75);
% w(1:75)=itd;%setpoint sequence
%
% temp(1:75)=1*ite;temp(76:150)=1*ite;
% temp(151:225)=1*ite;temp(226:300)=1*ite;
% ------------------------------------------
%
% Impulse input:
% ------------------------------------------
ns=300; %lenght of sample
temp(1:150)=1*ones(1,150);
temp(151:152)=1*ones(1,2);
temp(153:300)=1*ones(1,148);
w(1:300)=temp;
% ------------------------------------------
%
%

while length(w)<ns
      w=[w,temp];
end
w=w(1:ns);
%u(1,2)=0.01.*(2.*rand-1);
u=zeros(1,4);
uu=u;
y=zeros(1,5); x=y; U=y;


% Step 2: Specify desired polynomials Ac and R, compute T
% Change Ac as characteristics equation

deg_Ac=2; %degree of desired polynomial
Ac=[1 -1.068 0.1263];
%Ac=[1, -1.3205, 0.4966];
R=[1, -0.9, 0.009];
%T=1.4;
T=sum(Ac);
```

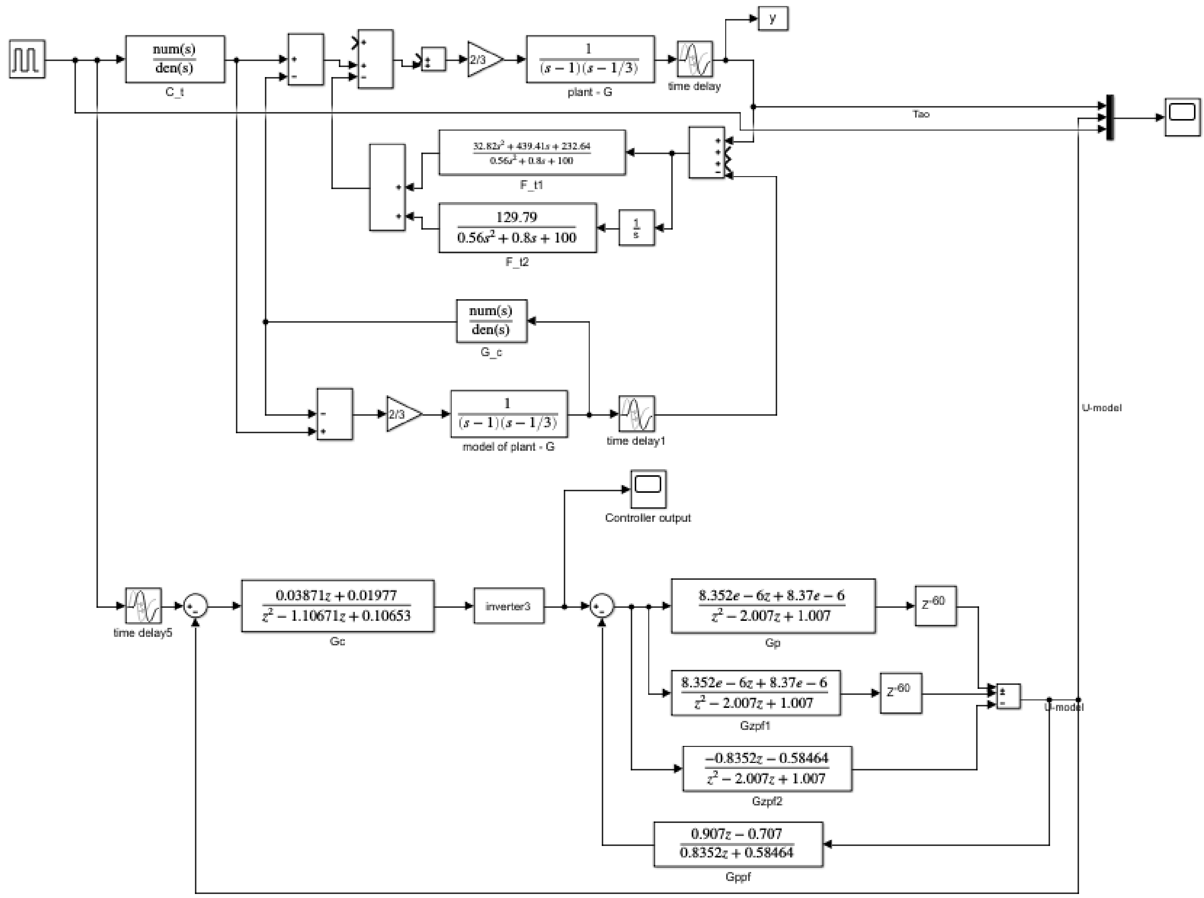
```
deg_R=deg_Ac;

% Step 3: Determine polynomial S by solving Diophantine equation

deg_S=1; %degree of desired polynomial S=
S=Ac(deg_S+1:deg_Ac+1)-R(deg_S+1:deg_Ac+1);

% Step 4: Generate alpha and plant output,
 for t=5:ns
    lamda_0=1.233*y(t-1)-0.5356*y(t-2)+0.005799*y(t-3)-(3.249e-11)*y(t-
4)+7.856*u(t-2)+2.392*u(t-3)+0.001445*u(t-4);
    %lamda_0=2.007*y(t-1)-1.007*y(t-2)+8.37e-6*u(t-2)+0.005*wgn(1,1,0.000001);
    %wgn is White Gaussian Noice
    alpha=[lamda_0, -6.69];
    u_temp = [1, uu(t-1)];
    y(t)=alpha*u_temp';


% Step 5: Controller design for U(t-1)
    U_temp = [U(t-1), U(t-2)];
    y_temp = [y(t), y(t-1)];
    w_temp = w(t);
    R_temp = R(2:length(R));

    U(t) = -R_temp*U_temp'+T*w_temp-S*y_temp';

    lamda_0_new=1.233*y(t)-0.5356*y(t-1)+0.005799*y(t-2)-3.249e-11*y(t-3)+(7.856-
6.19)*u(t-1)+(2.392-1.5534)*u(t-2)+(0.001445-0.0009424)*u(t-3);
%     u(t) = U(t)-lamda_0_new;
    u(t) = (U(t)-lamda_0_new);
    uu(t)=(u(t)-6.19*u(t-1)-1.5534*u(t-2)-0.00094424*u(t-3))/(-6.69);

end

% Step 6: Display simulation results
t=1:ns;
figure(1)
plot(0.1*t,w, '--', 0.1*t, y, '-')
axis([0 30 -0.1 1.2]);
%plot(0.1*t,y,'-')
xlabel('time(s)')
ylabel('amplitude')
%text(0.69,0.9, '----- reference', 'sc')
%text(0.69,0.8, '- - - plant output', 'sc')

figure(2)
plot(0.1*t,u),xlabel('time(s)'),ylabel('controller output')
```

C. MATLAB code for section 3.3.2

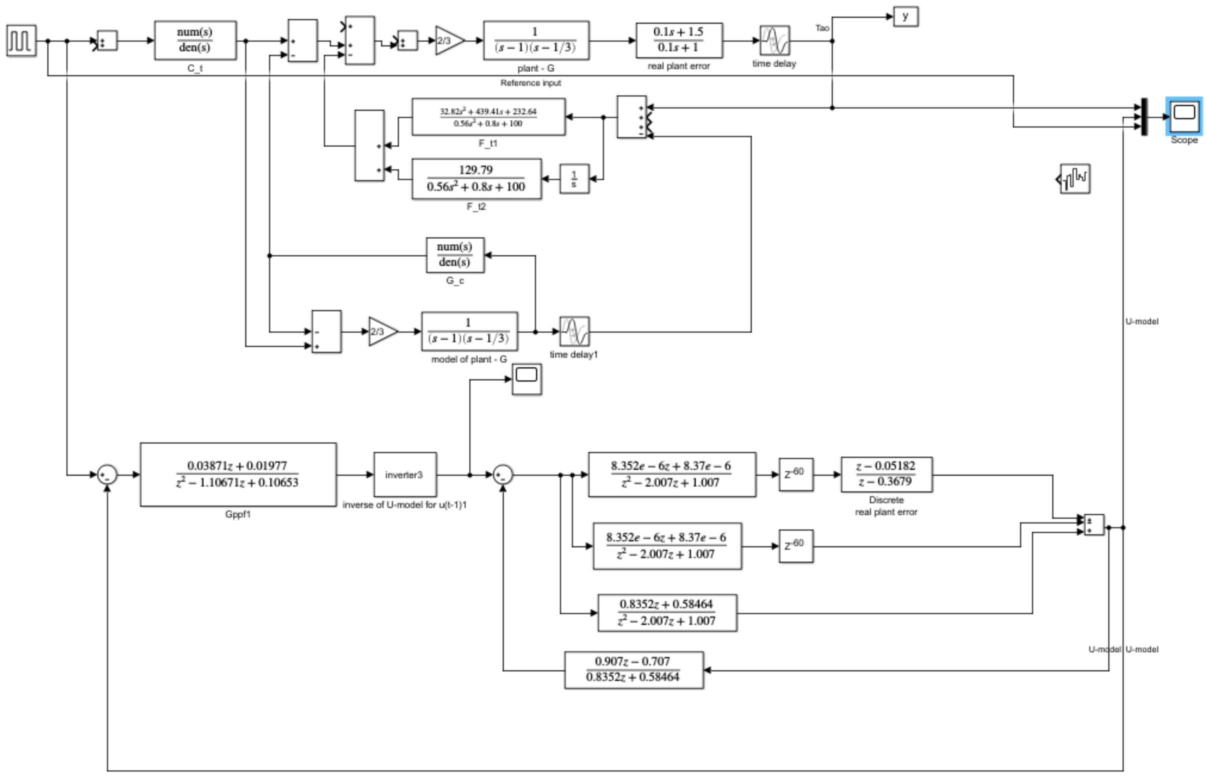
The .slx file described Tao method and U-model based controller case one is shown below.



The .slx file described Tao method and U-model based controller case two is shown below.



The .slx file described Tao method and U-model based controller case three is shown below



The s-function name inverter3 for case two and case three.

```
function [sys,x0,str,ts,simStateCompliance] = sfuntmpl(t,x,u,flag)
%SFUNTMPL General MATLAB S-Function Template
%   With MATLAB S-functions, you can define you own ordinary differential
%   equations (ODEs), discrete system equations, and/or just about
%   any type of algorithm to be used within a Simulink block diagram.
%
%   The general form of an MATLAB S-function syntax is:
%       [SYS,X0,STR,TS,SIMSTATECOMPLIANCE] = SFUNC(T,X,U,FLAG,P1,...,Pn)
%
%   What is returned by SFUNC at a given point in time, T, depends on the
%   value of the FLAG, the current state vector, X, and the current
%   input vector, U.
%
%   FLAG   RESULT             DESCRIPTION
%   -----  ------             --------------------------------------------
%   0      [SIZES,X0,STR,TS]  Initialization, return system sizes in SYS,
%                             initial state in X0, state ordering strings
%                             in STR, and sample times in TS.
%   1      DX                 Return continuous state derivatives in SYS.
%   2      DS                 Update discrete states SYS = X(n+1)
%   3      Y                  Return outputs in SYS.
%   4      TNEXT              Return next time hit for variable step sample
%                             time in SYS.
%   5                         Reserved for future (root finding).
%   9      []                 Termination, perform any cleanup SYS=[].
```

```
%
%
%   The state vectors, X and X0 consists of continuous states followed
%   by discrete states.
%
%   Optional parameters, P1,...,Pn can be provided to the S-function and
%   used during any FLAG operation.
%
%   When SFUNC is called with FLAG = 0, the following information
%   should be returned:
%
%      SYS(1) = Number of continuous states.
%      SYS(2) = Number of discrete states.
%      SYS(3) = Number of outputs.
%      SYS(4) = Number of inputs.
%               Any of the first four elements in SYS can be specified
%               as -1 indicating that they are dynamically sized. The
%               actual length for all other flags will be equal to the
%               length of the input, U.
%      SYS(5) = Reserved for root finding. Must be zero.
%      SYS(6) = Direct feedthrough flag (1=yes, 0=no). The s-function
%               has direct feedthrough if U is used during the FLAG=3
%               call. Setting this to 0 is akin to making a promise that
%               U will not be used during FLAG=3. If you break the promise
%               then unpredictable results will occur.
%      SYS(7) = Number of sample times. This is the number of rows in TS.
%
%
%      X0     = Initial state conditions or [] if no states.
%
%      STR    = State ordering strings which is generally specified as [].
%
%      TS     = An m-by-2 matrix containing the sample time
%                (period, offset) information. Where m = number of sample
%               times. The ordering of the sample times must be:
%
%               TS = [0      0,      : Continuous sample time.
%                     0      1,      : Continuous, but fixed in minor step
%                                      sample time.
%                     PERIOD OFFSET, : Discrete sample time where
%                                      PERIOD > 0 & OFFSET < PERIOD.
%                     -2     0];     : Variable step discrete sample time
%                                      where FLAG=4 is used to get time of
%                                      next hit.
%
%               There can be more than one sample time providing
%               they are ordered such that they are monotonically
%               increasing. Only the needed sample times should be
%               specified in TS. When specifying more than one
%               sample time, you must check for sample hits explicitly by
%               seeing if
%                  abs(round((T-OFFSET)/PERIOD) - (T-OFFSET)/PERIOD)
%               is within a specified tolerance, generally 1e-8. This
%               tolerance is dependent upon your model's sampling times
%               and simulation time.
%
%               You can also specify that the sample time of the S-function
```

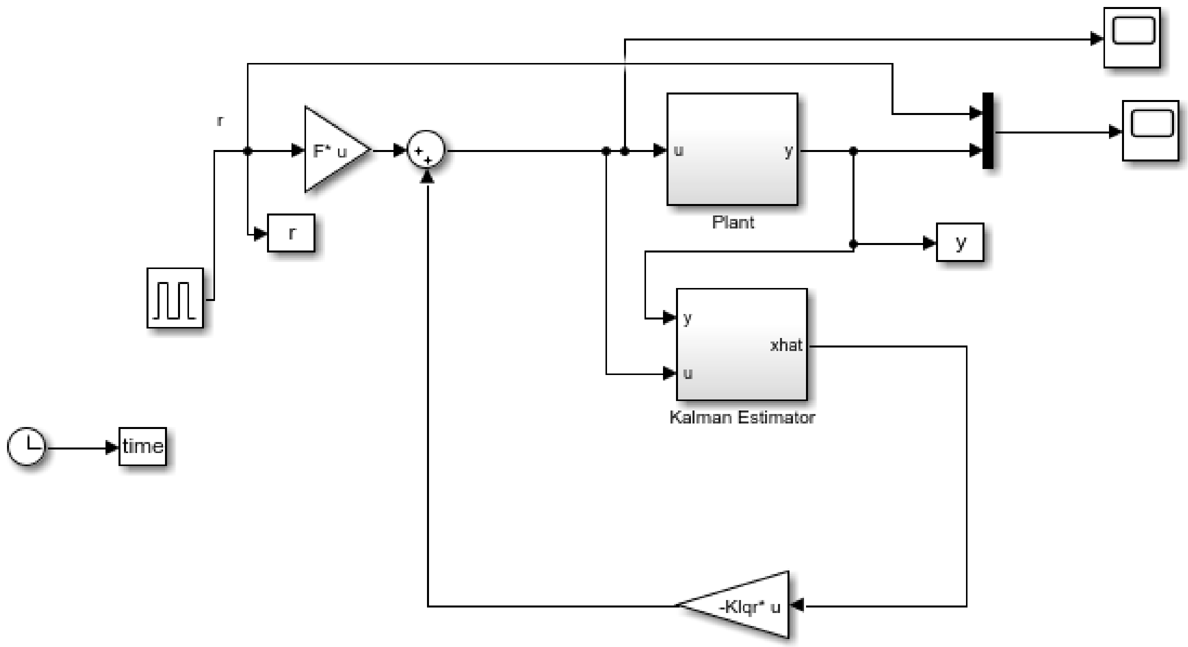
```
%                is inherited from the driving block. For functions which
%                change during minor steps, this is done by
%                specifying SYS(7) = 1 and TS = [-1 0]. For functions which
%                are held during minor steps, this is done by specifying
%                SYS(7) = 1 and TS = [-1 1].
%
%      SIMSTATECOMPLIANCE = Specifices how to handle this block when saving and
%                           restoring the complete simulation state of the
%                           model. The allowed values are: 'DefaultSimState',
%                           'HasNoSimState' or 'DisallowSimState'. If this value
%                           is not speficified, then the block's compliance with
%                           simState feature is set to 'UknownSimState'.


%   Copyright 1990-2010 The MathWorks, Inc.
%   $Revision: 1.18.2.5 $

%
% The following outlines the general structure of an S-function.
%
switch flag

  %%%%%%%%%%%%%%%%%%
  % Initialization %
  %%%%%%%%%%%%%%%%%%
  case 0
    [sys,x0,str,ts,simStateCompliance]=mdlInitializeSizes;

  %%%%%%%%%%
  % Update %
  %%%%%%%%%%
  case 2
    sys=mdlUpdate(t,x,u);

  %%%%%%%%%%%
  % Outputs %
  %%%%%%%%%%%
    case 3
        sys=mdlOutputs(t,x,u);

  %%%%%%%%%%%%%
  % Terminate %
  %%%%%%%%%%%%%
  case 9
    sys=[];

  %%%%%%%%%%%%%%%%%%%%
  % Unexpected flags %
  %%%%%%%%%%%%%%%%%%%%
  otherwise
    DAStudio.error('Simulink:blocks:unhandledFlag', num2str(flag));

end

% end sfuntmpl
```

```
%
%=============================================================================
% mdlInitializeSizes
% Return the sizes, initial conditions, and sample times for the S-function.
%=============================================================================
%
function [sys,x0,str,ts,simStateCompliance]=mdlInitializeSizes

%
% call simsizes for a sizes structure, fill it in and convert it to a
% sizes array.
%
% Note that in this example, the values are hard coded.  This is not a
% recommended practice as the characteristics of the block are typically
% defined by the S-function parameters.
%
sizes = simsizes;

sizes.NumContStates  = 0;
sizes.NumDiscStates  = 4;
sizes.NumOutputs     = 1;
sizes.NumInputs      = 1;
sizes.DirFeedthrough = 0;
sizes.NumSampleTimes = 1;   % at least one sample time is needed

sys = simsizes(sizes);


x0  = zeros(4,1);

str = [];
ts  = [-1 0];%inherited sample time

% Specify the block simStateCompliance. The allowed values are:
%    'UnknownSimState', < The default setting; warn and assume DefaultSimState
%    'DefaultSimState', < Same sim state as a built-in block
%    'HasNoSimState',   < No sim state
%    'DisallowSimState' < Error out when saving or restoring the model sim state
simStateCompliance = 'DefaultSimState';

% end mdlInitializeSizes
%
%=============================================================================
% mdlUpdate
% Handle discrete state updates, sample time hits, and major time step
% requirements.
%=============================================================================
%
function sys=mdlUpdate(t,x,u)
%the script started 16/1/2018
%updated 16/1/2018
%Reference: Zhu, Q.M.& Guo L.Z. (2002) A pole placement contorller for nonlinear
%                               dynamic plants
%inverting U-model to find controller outpust u1(t-1 and u2(t-1)
```

```
%
%
%s-function has 4 inputs: u1(t)=y_1d(t), u2(t)=y_2d(t)
%                          u3(t)=y_1(t), u4=y_2(t)
%s-function has 8 outputs: x(1,1)=y_1(t-1), x(2,1)=y_1(t-2)
%                           x(3,1)=y_2(t-1), x(4,1)=y_2(t-2)
%                           x(5,1)=u_1(t-2), x(6,1)=u_2(t-2)
%s-function has 2 outputs: x(7,1)=u_1(t-1), x(8,1)=u_2(t-1)

% 2 controller outputs

x(3,1)=(u(1)-1.1*x(1,1)+0.3*x(2,1)-5.8464e-6*x(4,1))/8.352e-6;
% x(3,1)=(u(1)-1.1*x(1,1)+0.3*x(2,1)-0.58464*x(4,1))/0.8352;

% update
x(2,1)=x(1,1);
x(1,1)=u(1);
%
x(4,1)=x(3,1);%

sys=x;

% end mdlUpdate

%
%=============================================================================
% mdlOutputs
% Return the block outputs.
%=============================================================================
%
function sys=mdlOutputs(t,x,u)
y=x(3,1);
sys =y;

% end mdlOutputs
```

D. MATLAB code for section 3.3.3

The .slx file of LQG design is shown below.



The .m file for LQG design.

```
%
clc; clear all; clf;
% -----------------------------------------------
A = [-0.00643        0.0263        0          -32.2         0;
        -0.0941       -0.624        820          0            0;
        -0.000222     -0.00153     -0.668        0            0;
        0                0            1            0            0;
        0               -1            0            830      0   ];

B = [0 ;
    -32.7;
    -2.08;
     0;
     0];

C = [0   0   0   0   1];
% -----------------------------------------------
sys = ss(A,B,C,[]);

%-------------------------------
%---------LQR design----------%
%-------------------------------

 % || x ||^2_Q + || u ||^2_R
Q = 10*eye(5);
R = eye(1);

[Klqr,S,e] = lqr(A,B,Q,R,[]); % Controller Gain u = -K x
```

```
F = -(C *(A - B*Klqr)^(-1) * B)^(-1);    % Feedforward for Tracking


%-----------------------------------
%---------Kalman design----------%
%-----------------------------------

% E(w w')=Q,?E(v v')=R,?E(w v)=N
Qn = 10e-3;
Rn = 10e-3;
[kest,L,P] = kalman(sys,Qn,Rn,[]);  % L is estimator gain

% --------------------------------------------------

tf = 40; % Simulation Time
sim('Iman_ji_LQG')

t = (time)';
r = (r)';
y = (y)';



% PLOT
figure(1)
plot(t,r,'--');hold on
plot(t,y);
ylabel('system response')
xlabel('time (s)')
% plot(t,r,'r--','LineWidth',2);hold on
% plot(t,y,'LineWidth',2);grid on
% title('Time Response','Interpreter','latex','FontSize',16,'FontWeight','bold')
% ylabel('setpoint \&
output','Interpreter','latex','FontSize',16,'FontWeight','bold')
% xlabel('Time (sec)','Interpreter','latex','FontSize',16,'FontWeight','bold')
```

The .m file for U-model based controller.

```
clc, clear;
% Start date:19/12/2016
% Update date:24/01/2017

% Example from Feedback Control of Dynamic Systems Page:492
%
% A=tf([32.7 1.128 -1035 -4.656],[1 4.564 14.29 0.8931 0.007807 0]);
% A =
%
%            32.7 s^3 + 1.128 s^2 - 1035 s - 4.656
%   ---------------------------------------------------
%   s^5 + 4.564 s^4 + 14.29 s^3 + 0.8931 s^2 + 0.007807 s
%
% p =
%
%    0.0000 + 0.0000i
```

```
%   -2.2500 + 2.9900i
%   -2.2500 - 2.9900i
%   -0.0531 + 0.0000i
%   -0.0105 + 0.0000i
% z =
%
%    -5.6400
%     5.6100
%    -0.0045
%
% B=c2d(A,0.1)
%
%B =
%
%   0.1358 z^4 - 0.3341 z^3 + 0.0618 z^2 + 0.2531 z - 0.1167
%   ----------------------------------------------------------
%   z^5 - 4.52 z^4 + 8.194 z^3 - 7.462 z^2 + 3.421 z - 0.6336
%
% Sample time: 0.1 seconds
% Discrete-time transfer function.
% z =
%
%    -0.8620
%     1.7530
%     0.9996
%     0.5689
% P =
%
%    1.0000 + 0.0000i
%    0.9990 + 0.0000i
%    0.9947 + 0.0000i
%    0.7631 + 0.2352i
%    0.7631 - 0.2352i
%
% So
% y(t)=4.52*y(t-1)-8.194*y(t-2)+7.462*y(t-3)-3.421*y(t-4)+0.6336*
%             y(t-5)+0.1358*u(t-1)-0.3341*u(t-2)+0.0618*u(t-3)+0.2531*u(t-4)
%             -0.1167*u(t-5)
%
%The desired plant is
%  C =
%
%              0.1358 z^4 - 0.05568 z^2 + 0.005432
%   ----------------------------------------------------------
%   z^5 - 4.52 z^4 + 8.194 z^3 - 7.462 z^2 + 3.421 z - 0.6336
% with zeros located in
%  z =
%
%     0.5000
%    -0.5000
%     0.4000
%    -0.4000
%
% ydesired(t)=4.52*y(t-1)-8.194*y(t-2)+7.462*y(t-3)-3.421*y(t-4)+0.6336*
%             y(t-5)+0.1358*u(t-1)-0.05568*u(t-3)+0.005432*u(t-5)
%
% Step 1: Initialisation
```

```
ns=400; %lenght of sample
ite=0.035*ones(1,75);itd=1*ite(1:75);
w(1:75)=itd;%setpoint sequence

temp(1:75)=1*ite;temp(76:150)=0*ite;
temp(151:225)=0*ite;temp(226:300)=1*ite;
% ite=2.*ones(1,20);itd=ite(1:10);
%
% w(1:10)=itd; %setpoint sequence
% temp(1:20)=ite;temp(21:40)=3*ite;%1.*ite;
% temp(41:60)=ite;temp(61:80)=0.*ite;
while length(w)<ns
      w=[w,temp];
end
w=w(1:ns);
%u(1,2)=0.01.*(2.*rand-1);
u=zeros(1,5);
uu=u;
y=zeros(1,6); x=y; U=y;
% g=randn(1,400);
% g=g/std(g);
% g=g-mean(g);
% p=0;
% q=sqrt(0.0001);
% g=p+q*g

% Step 2: Specify desired polynomials Ac and R, compute T

deg_Ac=2; %degree of desired polynomial
Ac=[1, -1.3205, 0.4966];
R=[1, -0.9, 0.009];
T=sum(Ac);
deg_R=deg_Ac;

% Step 3: Determine polynomial S by solving Diophantine equation

deg_S=1; %degree of desired polynomial S=
S=Ac(deg_S+1:deg_Ac+1)-R(deg_S+1:deg_Ac+1);

% Step 4: Generate alpha and plant output,
 for t=6:ns
    lamda_0=4.51980*y(t-1)-8.1937*y(t-2)+7.4615*y(t-3)-3.4212*y(t-4)+0.6336*y(t-5)-
0.3341*u(t-2)+0.06180*u(t-3)+0.2531*u(t-4)-0.1167*u(t-5);%+g(t);%+0.4*u(t-2)
    alpha=[lamda_0, 0.1358];

   u_temp = [1, uu(t-1)];
    y(t)=alpha*u_temp';


% Step 5: Controller design for U(t-1)
    U_temp = [U(t-1),U(t-2)];
    y_temp = [y(t), y(t-1)];
%     U_temp = [U(t-1),U(t-2), U(t-3), U(t-4),U(t-5)];
%    y_temp = [y(t), y(t-1),y(t-2),y(t-3),y(t-4)];
    w_temp = w(t);
```

```
    R_temp = R(2:length(R));

    U(t) = -R_temp*U_temp'+T*w_temp-S*y_temp';

    lamda_0_new=4.5198*y(t)-8.1937*y(t-1)+7.4615*y(t-2)-3.4212*y(t-3)+0.6336*y(t-
4)+(-0.3341+0.3341)*u(t-1)+(0.0618-0.1175)*u(t-2)+(0.2531-0.2531)*u(t-3)+(-
0.1167+0.1221)*u(t-4);%+g(t);%+0.1358*u(t)-0.05568*u(t-2)+0.005432*u(t-
4);%equivalent to assing zero z-0.5
    u(t) = (U(t)-lamda_0_new)/(0.1358);
    uu(t)=(0.1358*u(t)+0.3341*u(t-1)-0.1175*u(t-2)-0.2531*u(t-3)+0.1221*u(t-
4))/(0.1358);

end

% Step 6: Display simulation results
t=1:ns;
figure(1)
plot(t*0.1,y,'-')
%plot(t,w,'--',t, y, '-','LineWidth',2)
%xlabel('Time(sec)')
%ylabel('amplitude')
%text(0.69,0.9, '----- reference', 'sc')
%text(0.69,0.8, '- - - plant output', 'sc')

figure(2)
plot(t*0.1,u),xlabel('time(s)'),ylabel('controller output')
```

E. MATLAB code for section 4.4

The .m file for standard MIT rule MRAC

```
clear all;close all;
h=0.1;L=100/h;
num=[1];
den=[1 1 1];
n=length(den)-1;
kp=1;[Ap,Bp,Cp,Dp]=tf2ss(kp*num,den);
km=1;[Am,Bm,Cm,Dm]=tf2ss(km*num,den);

gamma=0.1;
alpha=0.01;beta=2;
yr0=0;u0=0;e0=0;ym0=0;
xp0=zeros(n,1);xm0=zeros(n,1);
kc0=0;
r=1;
yr=r*[ones(1,L/4) -ones(1,L/4) ones(1,L/4) -ones(1,L/4)];

for k=1:L
    time(k)=k*h;
    xp(:,k)=xp0+h*(Ap*xp0+Bp*u0);
    yp(k)=Cp*xp(:,k)+Dp*u0;

    xm(:,k)=xm0+h*(Am*xm0+Bm*yr0);
    ym(k)=Cm*xm(:,k)+Dm*yr0;

    e(k)=ym(k)-yp(k);
    DD=e0*ym0/km/(alpha+(ym0/km)^2);
    if DD<-beta
         DD=-beta;
    end
    if DD>beta
         DD=beta;
    end
    kc=kc0+h*gamma*DD;
    u(k)=kc*yr(k);

    yr0=yr(k);u0=u(k);e0=e(k);ym0=ym(k);
    xp0=xp(:,k);xm0=xm(:,k);
    kc0=kc;

end


subplot(2,1,1);set(gcf,'color','white')
ylim([-2 2])
plot(time,ym,'r',time,yp,'-.');
xlabel('t(s)');ylabel('y_m(t), y_p(t)');
legend('y_m(t)', 'y_p(t)');

subplot(2,1,2);
plot(time,u);
ylim([-1.5 1.5])
```

```
xlabel('t(s)'); ylabel('u(t)');
```

The .m file for U-model based MIT

```
clear all;close all;
h=0.1;L=100/h;
num=[1];
den=[1 1 1];
n=length(den)-1;
kp=1;[Ap,Bp,Cp,Dp]=tf2ss(kp*num,den);
km=1;[Am,Bm,Cm,Dm]=tf2ss(km*num,den);

gamma=0.1;
alpha=0.01;beta=2;

yr0=0;u0=0;e0=0;ym0=0;x=zeros(1,2); yp1=x; v=x;
xp0=zeros(n,1);xm0=zeros(n,1);
kc0=0;
r=1;
yr=r*[ones(1,L/4) -ones(1,L/4) ones(1,L/4) -ones(1,L/4)];

for k=3:L
    time(k)=k*h;
    xp(:,k)=xp0+h*(Ap*xp0+Bp*u0);
    yp(k)=Cp*xp(:,k)+Dp*u0;

    xm(:,k)=xm0+h*(Am*xm0+Bm*yr0);
    ym(k)=Cm*xm(:,k)+Dm*yr0;

    e(k)=ym(k)-yp(k);
    DD=e0*ym0/km/(alpha+(ym0/km)^2);
    if DD<-beta
         DD=-beta;
    end
    if DD>beta
         DD=beta;
    end
    kc=kc0+h*gamma*DD;
    u(k)=kc*yr(k);

    theta=[0.5*yp1(k-1)+1+0.1*x(k-2), 1, -1, 0.2];

    v_temp=[1, v(k-1), v(k-1)^2, v(k-1)^3];
    delta=[1, -1, 0.2]; % the derivative coefficicnts from alpha*u_temp'
                        % it will used for Newton_Raphason rool solving
    yp1(k)=0.5*yp1(k-1)+1+0.1*x(k-2)+v(k-1)-v(k-1)^2+0.2*v(k-1)^3;

    %v_j=v(k-1);

if k>2 % originally t>10 on 5/7/2001
% approach 1 --- standard matblab <roots> function
   p=theta(1)-yp(k);
   p=[p, theta(2:length(theta))];
```

```
    p=p(length(p):-1:1);
    root_temp=roots(p);
    real_root=[];
    for j=1:length(root_temp)
       test=isreal(root_temp(j));
       if test==1
          real_root=[real_root, root_temp(j)];
       end
    end
    [value, j]=min(abs(real_root));
    v(k)=real_root(j);

    % u(t)=w(t); %test open loop response
    v_temp=[1, v(k), v(k)^2, v(k)^3];
    x(k)=[1, 1, -1, 0.2]*v_temp';

     yr0=yr(k);u0=u(k);e0=e(k); ym0=ym(k);
     xp0=xp(:,k);     xm0=xm(:,k);
     kc0=kc;
end
end


subplot(2,1,1);set(gcf,'color','white')
ylim([-2 2])
plot(time,ym,'r',time,yp,'-.');
xlabel('t(s)');ylabel('y_m(t), y_p(t)');
legend('y_m(t)', 'y_p(t)');

subplot(2,1,2);
plot(time,v);
xlabel('t(s)'); ylabel('u(t)');
```

F. MATLAB code for section 5.4

The .m file for Lyapunov-MRAC controller when r=1.

```
clear all; close all;

h=0.1; L=100/h;
num=[2 1];den=[1 2 1]; n=length(den)-1;
kp=1; [Ap,Bp,Cp,Dp]=tf2ss(kp*num,den);
km=1; [Am,Bm,Cm,Dm]=tf2ss(km*num,den);

gamma=0.1;

yr0=0;u0=0;e0=0;
xp0=zeros(n,1); xm0=zeros(n,1);
kc0=0;
r=1;yr=r*[ones(1,L/4) -ones(1,L/4) ones(1,L/4) -ones(1,L/4)];

for k=1:L
    time(k)=k*h;
    xp(:,k)=xp0+h*(Ap*xp0+Bp*u0);
    yp(k)=Cp*xp(:,k);

    xm(:,k)=xm0+h*(Am*xm0+Bm*yr0);
    ym(k)=Cm*xm(:,k);

    e(k)=ym(k)-yp(k);
    kc=kc0+h*gamma*e0*yr0;
    u(k)=kc*yr(k);

    yr0=yr(k); u0=u(k);e0=e(k);
    xp0=xp(:,k);xm0=xm(:,k);
    kc0=kc;

end

subplot(2,1,1);set(gcf,'color','white')
plot(time,ym,'r',time,yp,'-.');
ylim([-1.5 1.5])
xlabel('time(s)');ylabel('System response');
legend('y_m(t)', 'y_p(t)');
subplot(2,1,2);

plot(time,u);ylim([-1.1 1.2])
xlabel('time(s)'); ylabel('Controller output');
```

The .m file for U-model based Lyapunov-MRAC controller when r=1

```
clear all; close all;

h=0.1; L=100/h;
num=[2 1];den=[1 2 1]; n=length(den)-1;
```

```
kp=1; [Ap,Bp,Cp,Dp]=tf2ss(kp*num,den);
km=1; [Am,Bm,Cm,Dm]=tf2ss(km*num,den);

gamma=0.1;

yr0=0;u0=0;e0=0;x=zeros(1,2); yp1=x; v=x;
xp0=zeros(n,1); xm0=zeros(n,1);
kc0=0;
r=1;yr=r*[ones(1,L/4) -ones(1,L/4) ones(1,L/4) -ones(1,L/4)];


for k=3:L
    time(k)=k*h;
    xp(:,k)=xp0+h*(Ap*xp0+Bp*u0);
    yp(k)=Cp*xp(:,k);

    xm(:,k)=xm0+h*(Am*xm0+Bm*yr0);
    ym(k)=Cm*xm(:,k);

    e(k)=ym(k)-yp(k);
    kc=kc0+h*gamma*e0*yr0;
    u(k)=kc*yr(k);

    %geenrate model output
    alpha=[0.5*yp1(k-1)+1+0.1*x(k-2), 1, -1, 0.2];

    v_temp=[1, v(k-1), v(k-1)^2, v(k-1)^3];
    beta=[1, -1, 0.2]; % the derivative coefficints from alpha*u_temp'
                       % it will used for Newton_Raphason rool solving
    yp1(k)=0.5*yp1(k-1)+1+0.1*x(k-2)+v(k-1)-v(k-1)^2+0.2*v(k-1)^3;

    %v_j=v(k-1);

if k>2 % originally t>10 on 5/7/2001
% approach 1 --- standard matblab <roots> function
   p=alpha(1)-yp(k);
   p=[p, alpha(2:length(alpha))];
   p=p(length(p):-1:1);
   root_temp=roots(p);
   real_root=[];
   for j=1:length(root_temp)
      test=isreal(root_temp(j));
      if test==1
         real_root=[real_root, root_temp(j)];
      end
   end
   [value, j]=min(abs(real_root));
   v(k)=real_root(j);

   % u(t)=w(t); %test open loop response
   v_temp=[1, v(k), v(k)^2, v(k)^3];
   x(k)=[1, 1, -1, 0.2]*v_temp';
    % controller design
    %
```

```
    yr0=yr(k); u0=u(k);e0=e(k);
    xp0=xp(:,k);xm0=xm(:,k);
    kc0=kc;
end
end

subplot(2,1,1);set(gcf,'color','white')
plot(time,ym,'r',time,yp,'-.');
ylim([-1.5 1.5])
xlabel('time(s)');ylabel('System response');
legend('y_m(t)', 'y_p(t)');
subplot(2,1,2);

plot(time,v);ylim([-1 -0.2])

%plot(time,v,'LineWidth',1.5);ylim([-1.1 0.3])
xlabel('time(s)'); ylabel('Controller output');
```

The .m file for Lyapunov-MRAC controller when r=2.

```
clear all; close all;

h=0.1; L=100/h;
num=[2 1];den=[1 2 1]; n=length(den)-1;
kp=1; [Ap,Bp,Cp,Dp]=tf2ss(kp*num,den);
km=1; [Am,Bm,Cm,Dm]=tf2ss(km*num,den);

gamma=0.1;

yr0=0;u0=0;e0=0;
xp0=zeros(n,1); xm0=zeros(n,1);
kc0=0;
r=2;yr=r*[ones(1,L/4) -ones(1,L/4) ones(1,L/4) -ones(1,L/4)];

for k=1:L
    time(k)=k*h;
    xp(:,k)=xp0+h*(Ap*xp0+Bp*u0);
    yp(k)=Cp*xp(:,k);

    xm(:,k)=xm0+h*(Am*xm0+Bm*yr0);
    ym(k)=Cm*xm(:,k);

    e(k)=ym(k)-yp(k);
    kc=kc0+h*gamma*e0*yr0;
    u(k)=kc*yr(k);

    yr0=yr(k); u0=u(k);e0=e(k);
    xp0=xp(:,k);xm0=xm(:,k);
    kc0=kc;

end
```

```
subplot(2,1,1);set(gcf,'color','white')
plot(time,ym,'r',time,yp,'-.');
ylim([-2.9 2.9])
xlabel('time(s)');ylabel('System response');
legend('y_m(t)', 'y_p(t)');
subplot(2,1,2);

plot(time,u);ylim([-3 3])
xlabel('time(s)'); ylabel('Controller output');
```

The .m file for U-model based Lyapunov-MRAC controller when r=2

```
clear all; close all;

h=0.1; L=100/h;
num=[2 1];den=[1 2 1]; n=length(den)-1;
kp=1; [Ap,Bp,Cp,Dp]=tf2ss(kp*num,den);
km=1; [Am,Bm,Cm,Dm]=tf2ss(km*num,den);

gamma=0.1;

yr0=0;u0=0;e0=0;x=zeros(1,2); yp1=x; v=x;
xp0=zeros(n,1); xm0=zeros(n,1);
kc0=0;
r=2;yr=r*[ones(1,L/4) -ones(1,L/4) ones(1,L/4) -ones(1,L/4)];


for k=3:L
    time(k)=k*h;
    xp(:,k)=xp0+h*(Ap*xp0+Bp*u0);
    yp(k)=Cp*xp(:,k);

    xm(:,k)=xm0+h*(Am*xm0+Bm*yr0);
    ym(k)=Cm*xm(:,k);

    e(k)=ym(k)-yp(k);
    kc=kc0+h*gamma*e0*yr0;
    u(k)=kc*yr(k);

    %geenrate model output
    alpha=[0.5*yp1(k-1)+1+0.1*x(k-2), 1, -1, 0.2];

    v_temp=[1, v(k-1), v(k-1)^2, v(k-1)^3];
    beta=[1, -1, 0.2]; % the derivative coefficints from alpha*u_temp'
                       % it will used for Newton_Raphason rool solving
    yp1(k)=0.5*yp1(k-1)+1+0.1*x(k-2)+v(k-1)-v(k-1)^2+0.2*v(k-1)^3;

    %v_j=v(k-1);

if k>2 % originally t>10 on 5/7/2001
```

```
% approach 1 --- standard matblab <roots> function
   p=alpha(1)-yp(k);
   p=[p, alpha(2:length(alpha))];
   p=p(length(p):-1:1);
   root_temp=roots(p);
   real_root=[];
   for j=1:length(root_temp)
      test=isreal(root_temp(j));
      if test==1
         real_root=[real_root, root_temp(j)];
      end
   end
   [value, j]=min(abs(real_root));
   v(k)=real_root(j);

   % u(t)=w(t); %test open loop response
   v_temp=[1, v(k), v(k)^2, v(k)^3];
   x(k)=[1, 1, -1, 0.2]*v_temp';
    % controller design
    %

    yr0=yr(k); u0=u(k);e0=e(k);
    xp0=xp(:,k);xm0=xm(:,k);
    kc0=kc;
end
end

subplot(2,1,1);set(gcf,'color','white')
plot(time,ym,'r',time,yp,'-.');
ylim([-2.9 2.9])
xlabel('time(s)');ylabel('System response');
legend('y_m(t)', 'y_p(t)');
subplot(2,1,2);

plot(time,v);ylim([-1.5 0.5])

%plot(time,v,'LineWidth',1.5);ylim([-1.1 0.3])
xlabel('time(s)'); ylabel('Controller output');
```

The .m file for Lyapunov-MRAC controller when r=4.

```
clear all; close all;

h=0.1; L=100/h;
num=[2 1];den=[1 2 1]; n=length(den)-1;
kp=1; [Ap,Bp,Cp,Dp]=tf2ss(kp*num,den);
km=1; [Am,Bm,Cm,Dm]=tf2ss(km*num,den);

gamma=0.1;

yr0=0;u0=0;e0=0;
xp0=zeros(n,1); xm0=zeros(n,1);
kc0=0;
```

```
r=4;yr=r*[ones(1,L/4) -ones(1,L/4) ones(1,L/4) -ones(1,L/4)];

for k=1:L
    time(k)=k*h;
    xp(:,k)=xp0+h*(Ap*xp0+Bp*u0);
    yp(k)=Cp*xp(:,k);

    xm(:,k)=xm0+h*(Am*xm0+Bm*yr0);
    ym(k)=Cm*xm(:,k);

    e(k)=ym(k)-yp(k);
    kc=kc0+h*gamma*e0*yr0;
    u(k)=kc*yr(k);

    yr0=yr(k); u0=u(k);e0=e(k);
    xp0=xp(:,k);xm0=xm(:,k);
    kc0=kc;

end

subplot(2,1,1);set(gcf,'color','white')
plot(time,ym,'r',time,yp,'-.');
ylim([-6 6])
xlabel('time(s)');ylabel('System response');
legend('y_m(t)', 'y_p(t)');
subplot(2,1,2);

plot(time,u);ylim([-6 6])
xlabel('time(s)'); ylabel('Controller output');
```

The .m file for U-model based Lyapunov-MRAC controller when r=4

```
clear all; close all;

h=0.1; L=100/h;
num=[2 1];den=[1 2 1]; n=length(den)-1;
kp=1; [Ap,Bp,Cp,Dp]=tf2ss(kp*num,den);
km=1; [Am,Bm,Cm,Dm]=tf2ss(km*num,den);

gamma=0.1;

yr0=0;u0=0;e0=0;x=zeros(1,2); yp1=x; v=x;
xp0=zeros(n,1); xm0=zeros(n,1);
kc0=0;
r=4;yr=r*[ones(1,L/4) -ones(1,L/4) ones(1,L/4) -ones(1,L/4)];


for k=3:L
    time(k)=k*h;
    xp(:,k)=xp0+h*(Ap*xp0+Bp*u0);
    yp(k)=Cp*xp(:,k);
```

```
    xm(:,k)=xm0+h*(Am*xm0+Bm*yr0);
    ym(k)=Cm*xm(:,k);

    e(k)=ym(k)-yp(k);
    kc=kc0+h*gamma*e0*yr0;
    u(k)=kc*yr(k);

    %geenrate model output
    alpha=[0.5*yp1(k-1)+1+0.1*x(k-2), 1, -1, 0.2];

    v_temp=[1, v(k-1), v(k-1)^2, v(k-1)^3];
    beta=[1, -1, 0.2]; % the derivative coefficints from alpha*u_temp'
                       % it will used for Newton_Raphason rool solving
    yp1(k)=0.5*yp1(k-1)+1+0.1*x(k-2)+v(k-1)-v(k-1)^2+0.2*v(k-1)^3;

    %v_j=v(k-1);

if k>2 % originally t>10 on 5/7/2001
% approach 1 --- standard matblab <roots> function
   p=alpha(1)-yp(k);
   p=[p, alpha(2:length(alpha))];
   p=p(length(p):-1:1);
   root_temp=roots(p);
   real_root=[];
   for j=1:length(root_temp)
      test=isreal(root_temp(j));
      if test==1
         real_root=[real_root, root_temp(j)];
      end
   end
   [value, j]=min(abs(real_root));
   v(k)=real_root(j);

   % u(t)=w(t); %test open loop response
   v_temp=[1, v(k), v(k)^2, v(k)^3];
   x(k)=[1, 1, -1, 0.2]*v_temp';
    % controller design
    %

    yr0=yr(k); u0=u(k);e0=e(k);
    xp0=xp(:,k);xm0=xm(:,k);
    kc0=kc;
end
end

subplot(2,1,1);set(gcf,'color','white')
plot(time,ym,'r',time,yp,'-.');
ylim([-6 6])
xlabel('time(s)');ylabel('System response');
legend('y_m(t)', 'y_p(t)');
subplot(2,1,2);

plot(time,v);ylim([-2 5])

%plot(time,v,'LineWidth',1.5);ylim([-1.1 0.3])
```

```
xlabel('time(s)'); ylabel('Controller output');
```

Appendix C

U-model enhanced control of non-minimum phase systems

Quanmin Zhu[1], Ji Qiu[1]*, Iman S. Delshad[2], Mokhtar Nibouche[1], and Yufeng Yao[1]

1 Department of Engineering Design and Mathematics, University of the West of England, Bristol, Coldharbour Lane BS16 1QY, United Kingdom, quan.zhu@uwe.ac.uk; ji2.qiu@live.uwe.ac.uk; mokhtar.nibouche@uwe.ac.uk; yufeng.yao@uwe.ac.uk

2 Autonomous Driving / ADAS Engineer AVL MTC Sweden, Transmissionsvägen 2, 151 48 Södertälje 48 Södertälje, iman.delshad@avl.com

* Corresponding author

Abstract: Control of non-minimum phase (NMP) dynamic systems has been a widely and intensively studied challenging topic from its academic research to industrial applications. With the emergence of U-model based control system design, this study presents a new solution in designing such control systems. In technique, different from pole-zero cancellation (i.e. in force of multiplication/division), this study proposes a pulling principle (i.e. in force of summation/subtraction) to relocate zeros and poles through Infinite Impulse Response (IIR) filters, then introduces a U-model based universal control framework with invariant controller plus U-model inverter so that it separates the treatment of NMP issue from stable control system design. By integration of the above techniques, this study presents a systematic procedure to complement those classical approaches in designing NMP control systems. The associated properties and performance are proved analytically and demonstrated numerically (with Matlab simulation of bench test examples). To follow the study, the computational experiment results provide a user-friendly step by step procedure for the readers/users with interest in their *ad hoc* applications.

Keywords: Non-minimum phase (NMP) systems; zero and pole pulling; IIR filters; U-model based control systems design, U-control.

1. Introduction

Linear time-invariant dynamic systems, for which their inverses are unstable, are known as non-minimum-phase (NMP) or internal unstable systems (Franklin, Powell, & Emami-Naeini, 2014). Correspondingly, the zeros of the discrete-time systems are outside the unit circle, and in the meantime, the zeros of the continuous-time system are on the right-hand side of the complex plane. A given non-minimum phase system has long time delay response than its minimum-phase (MP) system, because of a greater phase delay contrast to its MP system with the equivalent magnitude frequency response. In general, a stable numerator polynomial of a transfer function is called stable zero dynamics. It has been noted that a stable zero dynamic of a system is particular important for model matching control and adaptive control (Slotine & Li, 1991).

Non-minimum phase (NMP) systems with inverse response have been widely appeared in industry, of the fluidized bed combustor, continuous stirred tank reactor and water turbine (Sun, Shi, Chen, & Yang, 2016), also the other examples, an altitude-hold flight of an autopilot of Boeing 747 (Franklin et al., 2014), a rotary mechanical plant (Freeman*, Lewin, & Rogers, 2005). It has noted that discretized model from its continuous plant even could become a NMP delayed model due to the inappropriate sampling (K. J. Åström, Hagander, & Sternby, 1984; García, Albertos, & Hägglund, 2006).

Control of non-minimum phase (NMP) systems is challenging as its unstable reverse response in time domain and the additional phase lag in frequency domain (Sun et al., 2016). A number of literatures have investigated the cancellation of unstable zeros in NMP systems such as Zero-Phase-Error Tracking Controller (ZPETC) (Gross, Tomizuka, & Messner, 1994; Haack & Tomizuka, 1991; Tomizuka, 1987), Zero-Magnitude-Error Tracking Controller (ZMETC) (Butterworth, Pao, &

Abramovitch, 2008), and Nonminimum-Phase Zeros Ignore (NPZ-Ignore) (Rigney, Pao, & Lawrence, 2010) respectively. Some other methods, such as non-causal-series expansion (Gross et al., 1994; Rigney et al., 2010), and using the exact unstable inverse to maintain the stability of the system by pre-loading initial condition or using non-causal plant input. Most of existing solutions lead to complex controllers with intricate algorithms where the control methods are strictly based upon and/or limited to an explicit plant model. Actually, in modern industrial control process, it is almost impossible to have an exactly accurate filter to cancel the NMP zeros and unstable poles. Where there is any internal uncertainty appears, the cancellation could not execute successfully. Further, it is very much possible to cause problems in system stability.

To avoid those problems caused from pole-zero cancellation, Some non-pole-zero cancellation approaches have focused on sliding mode control (SMC) for its high levels of robust performance in terms of dealing with NMP systems (Do et al., 2016; Mirkin, Gutman, & Shtessel, 2014, 2012; Patil, Bandyopadhyay, Chalanga, & Arya, 2018). These controllers need a prior knowledge such as fuzzy logic or neural network to establish an online adaptation mechanism to avoid the restrictive constraint on the knowledge of the bounds of uncertain dynamics. Some other designs (García et al., 2006; Liu, Zhang, & Gu, 2005; Tan, Marquez, & Chen, 2003) have adopted conventional controllers, such as PID controllers and New Smith Predictor (NSP) to enhance the capability of reference input tracking and load disturbance rejection for various unstable processes with time delay. However these approaches involve in more than three controllers and increase the complexity of the design process (Liu et al., 2005).

Easy to design/tune is very important in the process industry. Contrast to model-based control (Slotine & Li, 1991) and model-free (or data driven) control (Xu & Tan, 2003), there is a feasibility using model-independent design framework so that control system design and controller output determination can be independently determined (Q. Zhu & Guo, 2002). U-model based control, U-control in short, is such a representative approach (Q. Zhu, Liu, Zhang, & Li, 2018; Q. M. Zhu, Zhao, & Zhang, 2016). However all above explained U-controls (Geng, Zhu, Liu, & Na, 2019; Q. Zhu, Zhang, Zhang, & Sun, 2019) have assumed stable zero dynamic (in the assumption of the inverse dynamic exist) of the controlled plant/process, that is MP models in linear systems. Consequently, one of the next stage progressions in U-control is in the interest of dealing with unstable zero dynamics and unstable pole dynamics. This is one of the motivations in the present study, which starts from linear dynamic systems at first in order to lay a solid foundation for dealing with nonlinear zero dynamics in future studies.

Despite enormous efforts have been devoted to handling with NMP system control, there is still space to further improve the efficiency/effectiveness and generality in such control system design, for example,

1) More flexible robust approaches to deal with NMP models, that is, convert NMP models into MP models with more tolerance for the pole/zero uncertainties.

2) Easy to design/tune is very important in the process industry. It is feasible to separate NMP model treatment from the whole control system design.

From the aforementioned analysis and critical review, this study justifies its contribution below.

1) It proposes a pulling principle to guide replacing zero/pole cancellation (this is a multiplication/division operation) with zero and pole pulling relocation (this is a summation/subtraction operation), which is much less sensitive in stability and NMP issues, and more generic compared with classical zero/pole cancellation. Further, this pulling operation is systematic and concise within a well-constructed framework.

2) Taking up U-model based design, i.e. --- a plant model independent design procedure, it separates the control system design and the conversion of NMP model into a stable MP model. The design is composed of an invariant controller with specified dynamic and static performance within a stable closed loop and a dynamic inverter of the plant model (implemented by resolving the plant U-model root).

3) It takes up a series of simulation experiments to demonstrate the efficiency/effectiveness of the proposed approaches. It is not only to demonstrate the derived analytical results with numerical tests, but also to confirm that the case studies are helpful to show the application procedure for potential reader/users.

The rest of the main study is divided into four sections. Section 2 provides relevant foundation and notations for the following section development. Section 3 proposes the pulling principle and proves the pulling theorems for zero and pole relocation using Infinite impulse response (IIR) filters within proper structures. Section 4 presents the details of the U-model based control system design procedure for NMP systems. Section 5 conducts simulation bench tests of some examples, not only for demonstration of developed results, but also for a user friendly showcase in application of the procedures.

2 Problem formulations

2.1 Non-minimum phase (NMP) systems

Consider a linear discrete time single input and single output (SISO) dynamic plant model in terms of *z* transform

$$\frac{Y(z)}{U(z)} = G_p(z) = G_p(A,B) = \frac{B(z)}{A(z)} = \frac{b_0 z^m + b_1 z^{m-1} + \ldots + b_{m-1} z + b_m}{z^n + a_1 z^{n-1} + \ldots + a_{n-1} z + a_n} \quad n \succ m \quad (2.1)$$

where $Y(z)$ is the *z* transform of output $y(t) \in \mathbf{R}$, $U(z)$ is the *z* transform of input $u(t) \in \mathbf{R}$, $t \in \mathrm{Z}^+$ is the sampling instance, $\begin{pmatrix} a_0 & \cdots & a_n \end{pmatrix} \in \mathbf{R}^n$ and $\begin{pmatrix} b_0 & \cdots & b_m \end{pmatrix} \in \mathbf{R}^m$ are the coefficient vectors of the denominator polynomial $A(z)$ and numerator polynomials $B(z)$ of the transfer function respectively, B and A are coprime, and $G(z)$ is monic and strictly proper.

The transfer function can be factorised in terms of poles and zeros below.

$$\frac{Y(z)}{U(z)} = G_p(z) = \frac{B(z)}{A(z)} = \frac{\prod (z+z_1)^{m_i}}{\prod (z+p_1)^{n_i}} \quad \sum m_i = m \quad \sum n_i = n \quad (2.2)$$

The plant is unstable if one of the poles outside of the unit circle and it is defined as a non-minimum phase plant if one of the zeros outside of the unit circle. Accordingly, the plant is unstable non-minimum phase while it has both pole(s) and zero(s) outside the unit circle.

Control objectives: For a general stable MP process, the control objective is specified as: 1) given a desired trajectory $y_d(t)$, find an executable $u(t)$ to drive the system output $y(t)$ to track the desired trajectory $y_d(t)$ with acceptable performance (such as transient response and steady-state error), while all signals of the control system are bounded within the permitted ranges. For an unstable NMP process, in addition to achieve objective 1, 2) it need convert the process into a specified stable MP one, 3) using U-model based control design procedure, objectives 1 and 2 are parallel, separately achieved, and connected by a dynamic inverter of the specifically determined MP model.

Remark 1: The primary insight of the control objectives described above is to separate the conversion of NMP to MP and the MP based control system design.

2.2 U-model (Q. Zhu & Guo, 2002)

For a general SISO ($u(t) \in \mathrm{R}$ /input and $y(t) \in \mathrm{R}$ /output) U-polynomial model with respect to $u(t-1)$, assign a triplet $\left(P, L, f_p\right)$ where $P = \left\{P_\alpha \,|\, \alpha \in u(t-1)\right\}$ is a vector field of appropriate dimension and analytic defined on input $u(t-1)$, and $L = \left\{L_{\lambda(t)} \,|\, \lambda(t) \notin u(t-1)\right\}$ is a dynamic absorbing vector filed of appropriate dimension and associated with P. Accordingly, the system U-model $\sum_{\text{U-polynomial model}}$ is defined as a dynamic system representative, where the polynomial function $f_p = \left\{f_p(\alpha) \,|\, \alpha \in L, U\right\}$ is a mapping $f_p : u(t-1) \xrightarrow{L \cup U} y(t) \in \mathrm{R}$} from the input space to the output space. Consequently, it is expressed as

$$y(t) = \sum_{j=0}^{M} \lambda_j(t) p_j(u(t-1)) \quad (2.3)$$

where $t \in \mathrm{Z}^+$ for discrete time instant, M is the number of model input $u(t-1)$ terms, the time varying parameter vector $\lambda(t) = \left[\lambda_0(t) \quad \cdots \quad \lambda_M(t)\right] \in \mathrm{R}^{M+1}$ is a time varying function associated with $p_j(u(t-1))$ to absorb the other terms such as past inputs and outputs, and the other parameters in those classical dynamic polynomials.

This derived control oriented U-model has proper algebra properties for the mapping between U-model and classical models (Q. M. Zhu et al., 2016). It can represent almost all the standard linear and nonlinear discrete time dynamic models such as Hammerstein and Wiener models (Babík & Dostál, 2012), and Non-linear AutoRegressive Moving Average with eXogenous input (NARMAX) models (Billings, 2013) which have been widely used in process engineering and the other engineering applications. Here an illustrative example is selected to show the conversion from classical model to the U-model, $y(t) = 0.5y(t-1) - 0.2\cos(y^3(t-2)) + u(t-1) + 0.4u(t-2)\sin(u(t-1)) - 0.1\exp(-u^2(t-1))$, its corresponding U-model can be determined by $y(t) = \lambda_0(t) + \lambda_1(t)u(t-1) + \lambda_2(t)\sin(u(t-1)) + \lambda_3(t)\exp(-u^2(t-1))$, $\lambda_0(t) = 0.5y(t-1) - 0.2\cos(y^3(t-2))$, $\lambda_1(t) = 1$, $\lambda_2(t) = 0.4u(t-2)$, and $\lambda_3(t) = -0.1$.

For the interest of this study, to express the linear discrete time dynamic model of (2.1) in term of U-model, firstly convert the Z transfer function into its corresponding difference equation.

$$y(t) = -\sum_{j=1}^{n} a_j y(t-j) + \sum_{j=0}^{m} b_j u(t-j-1) \quad b_0 \neq 0 \quad (2.4)$$

Consequently, its U-model can be realised by

$$y(t)=\lambda_0(t)+\lambda_1(t)u(t-1)$$
$$\lambda_0(t)=A_0+B_0=-\sum_{j=1}^{n}a_j y(t-j)+\sum_{j=1}^{m}b_j u(t-j-1) \quad (2.5)$$
$$\lambda_1(t)=b_0$$

Remark 2: Both models (2.5 and (2.5) are the same in representation of input/output relationship, but different in expression. The classical model (2.4) may provide the first principle driven physical/chemical component structured input/output representation and straightforward for model structure detection and parameter estimation from input and output measurements. However, for control system design, it must re-design controllers while the plant model changed, such as the well know pole placement control design (Karl J Åström & Wittenmark, 2013). U-model (2.5), on the other hand, a derived expression from classical model (2.4), is particularly suitable for control system design which has given a general platform for both linear and nonlinear control system design and it establishes a platform to design the closed loop system independent from the plant model (Q. M. Zhu et al., 2016).

2.3 Inversion of u-polynomial models (Q. Zhu & Guo, 2002)

U-model provides a generic platform for dynamic inversion, which could be very much beneficial in control system design. The inverse of the U-model is a critical part in design, it is formulated by solving the roots from a given output $y_d(t)$ and a set of time varying parameters $\lambda(t)=[\lambda_0(t) \quad \cdots \quad \lambda_M(t)]\in R^{M+1}$ in the form of an equation as

$$y_d(t)-\sum_{j=0}^{M}\lambda_j(t)p_j(u(t-1))=0 \quad (2.6)$$

Then determine the input $u(t-1)$ from one of the roots, normally select the one with minimum amplitude in account of the minimum energy cost and practical saturation limits. Therefore, it has

$$\min(u(t-1))\in G_p^{-1}\Big|_{y_d(t)-\sum_{j=0}^{M}\lambda_j(t)p_j(u(t-1))=0} \quad (2.7)$$

In fact, the inverse is claimed to achieve $y_d(t)\to G_p^{-1}(\text{U-model})\to u(t-1)\to G_p\to y_d(t)$, accordingly, determining G_p^{-1} is to solve the root $u(t-1)$ from U-model for a given $y_d(t)$. It is noted $u(t-1)$ being the G_p^{-1} output.

For a linear U-polynomial model, the root solver can be concisely expressed with reference to (2.5) as

$$u(t-1)=\frac{y_d(t)-\lambda_0(t)}{\lambda_1(t)} \quad (2.8)$$

For general nonlinear U-polynomial model, it can consider using the Newton-Raphson algorithm (Gerald, 2004; Q. Zhu & Guo, 2002) or the other available root solving algorithms (Chong & Zak, 2011).

2.4 Infinite impulse response (IIR) filters (Proakis & Manolakis, 1988)

Infinite impulse response (IIR) is one of the properties exhibited in linear time-invariant systems. Those with this property are known as IIR systems or IIR filters. The IIR filter structure contains two main parts: one is N delay connection for poles and the other is M delay connection for zeros. This type of IIR can separately adjust the locations of poles and zeros. Therefore an N order filter only need N delay unit.

Let the linear time-invariant discrete-time IIR systems described by a general linear constant-coefficient difference equation

$$y(t)=\sum_{k=0}^{M} b_k x(t-k)-\sum_{k=1}^{N} a_k y(t-k),\ N \geq M \quad (2.9)$$

This can be expressed as a rational system function in terms of Z transform

$$\frac{Y(z)}{X(z)}=H(z)=\frac{\sum_{k=0}^{M} b_k z^{-k}}{1+\sum_{k=1}^{N} a_k z^{-k}}=\frac{b_0+b_1 z^{-1}+b_2 z^{-2}+\ldots+b_M z^{-M}}{1+a_1 z^{-1}+a_2 z^{-2}+\ldots+a_N z^{-N}},\ N \geq M \quad (2.10)$$

where with time-shift operator z^{-k}, it gives $z^{-k}Y(z)=y(t-k)$.

Remark 3: This study uses the IIR filters to form a pulling force by subtracting/summating operation to relocate zeros/poles a linear dynamic process into a stable minimum phase system. Regarding the structure of the integrated system, IIR filters are placed in parallel with the underlying process, either in feedforward to relocate the process zeros and/or feedback to relocate the process poles.

3. Relocation of zeros and poles --- pulling principle guided procedure

The purpose of relocating poles and zeros of an unstable NMP plant $G_p(z)$ is to change it into a specified stable MP $\hat{G}_p(z)$ with assigned poles and zeros. Consequently, the control systems can be designed based on a stable MP plant model procedure. This study proposes a pulling principle to guide the relocation of the poles and zeros. The pulling principle heuristically appeared (though not rigorously defined) for the first time in one of the research group conference papers (Qiu, Delshad, Zhu, Nibouche, & Yao, 2017).

Pulling principle: For a linear system expanded by polynomials (A,B), there exist a unit circle region Ω centred at origin with $\forall\left(|A|<1,|B|<1\right)\in\Omega$. There exists a pulling algebra F_p to map the others $\left(|A|\geq 1,|B|\geq 1\right)$ into Ω, that is, $(A,B)\setminus\Omega \xrightarrow{F_p} \Omega$.

This is described as a systematic approach to pull zeros and poles from their original positions in a given linear dynamic transfer function (i.e. it could be unstable and NMP) to some specified points to form a pre-specified new transfer function (such as stable and MP). The pulling algebra F_p can be functioned with Infinite Impulse Response (IIR) filters in parallel to a given plant in a properly structured framework (i.e. feedforward and feedback loops).

Proposition: Specify a stable MP plant $\hat{G}_p\left(\hat{B},\hat{A}\right)$

$$\hat{G}_p(z) = \frac{\hat{B}(z)}{\hat{A}(z)} = \frac{\hat{b}_0 z^m + \hat{b}_1 z^{m-1} + \ldots + \hat{b}_{m-1} z + \hat{b}_m}{z^n + \hat{a}_1 z^{n-1} + \ldots + \hat{a}_{n-1} z + \hat{a}_n}$$

$$= \frac{\prod_{j=1}^{m} (z - r_{zj})}{\prod_{j=1}^{n} (z - r_{pj})} \qquad n \succ m \quad (3.1)$$

where $(\hat{b}_0 \quad \ldots \quad \hat{b}_m) \in \mathrm{R}^m$ and $(1 \quad \hat{a}_1 \quad \cdots \quad \hat{a}_n) \in \mathrm{R}^n$ are the coefficient vectors of polynomials $\hat{B}(z)$ and $\hat{A}(z)$ respectively, and the factorised zeros ($(z - r_{zj})$) and poles ($(z - r_{pj})$) of $\hat{G}_p$ are all within the unit circle.

Define zero differencing operator and pole differencing operator between $G_p(B,A)$ and $\hat{G}_p(\hat{B},\hat{A})$ as B_{FF} and A_{FF} respectively, that is

$$\begin{aligned} B_{FF} &= B - \hat{B} = b_{FF0} z^m + b_{FF1} z^{m-1} + \ldots + b_{FF(m-1)} z + b_{FFm} \\ A_{FB} &= A - \hat{A} = a_{FB0} z^n + a_{FB1} z^{n-1} + \ldots + a_{FB(n-1)} z + a_{FBn} \end{aligned} \quad (3.2)$$

where

$$\begin{aligned} b_{FFi} &= b_i - \hat{b}_i, \quad i = 0,\ldots,m \\ a_{FBj} &= a_j - \hat{a}_j, \quad j = 0,\ldots,n \end{aligned} \quad (3.3)$$

Accordingly, the specified stable MP plant $\hat{G}_p(\hat{B},\hat{A})$ zeros and poles can be realised with the differencing operators. It gives by re-arranging (3.2)

$$\begin{aligned} \hat{B} &= B - B_{FF} \\ \hat{A} &= A - A_{FB} \end{aligned} \quad (3.4)$$

The following three theorems are presented for the existence of such pulling principle, which is implementable through IIR filters within proper structures.

Zero pulling theorem: Let $G_p(B,A)$ be a stable NMP transfer function, and $\hat{G}_p(\hat{B},A)$ is a correspondingly specified stable MP transfer function, G_{zpf} a zero pulling filter, and F_z a structure mapping function. Then it has $\hat{G}_p(\hat{B},A) = F_z(G(B,A),G_{zpf})$.

Proof: Define the zero pulling filter as

$$G_{zpf} = \frac{-B_{FF}}{A} \quad (3.5)$$

And the structure function is defined in Figure 3.1 (a). By using block diagram operational algebra, it can derive the transfer function of output against input as

$$\frac{Y}{U} = F_z\left(G(B,A),G_{zpf}\right) = G_p + G_{zpf} = \frac{B - B_{FF}}{A} = \frac{\hat{B}}{A} = \hat{G}\left(\hat{B},A\right) \quad (3.6)$$

Pole pulling theorem: Let $G_p\left(B,A\right)$ be an unstable MP transfer function, and $\hat{G}_p\left(B,\hat{A}\right)$ is the correspondingly specified stable MP transfer function, G_{ppf} a pole pulling filter, and F_p a structure mapping function. Then it has $\hat{G}_p\left(B,\hat{A}\right) = F_p\left(G(B,A),G_{ppf}\right)$.

Proof: Define the pole pulling filter as

$$G_{ppf} = \frac{-A_{FB}}{B} \quad (3.7)$$

With the structure mapping function F_p, it can build up a block diagram connecting $\left(G(B,A),G_{ppf}\right)$ in Figure 3.1(b). The whole transfer function between the output and the input is derived by block diagram operation algebra as

$$\frac{Y}{U} = F_p\left(G(B,A),G_{ppf}\right) = \frac{G_p}{1 + G_p G_{ppf}} = \frac{\frac{B}{A}}{1 + \frac{B}{A}\left(\frac{-A_{FB}}{B}\right)} = \frac{B}{A - A_{FB}} = \hat{G}\left(B,\hat{A}\right) \quad (3.8)$$

Zero and pole pulling theorem: Let $G_p\left(B,A\right)$ be an unstable NMP transfer function, and $\hat{G}_p\left(\hat{B},\hat{A}\right)$ is a correspondingly specified stable MP transfer function, G_{zpf} the zero pulling filter, G_{ppf} the pole pulling filter, and F_{zp} a structure mapping function. Then it has $\hat{G}_p\left(\hat{B},\hat{A}\right) = F_{zp}\left(G(B,A),G_{zpf},G_{ppf}\right)$.

Proof: With the structure mapping function F_{zppf}, it can build up a block diagram connecting $\left(G(B,A),G_{zpf},G_{ppf}\right)$ in Figure 3.1(c). The whole transfer function between the output and the input is derived as

$$\frac{Y}{U} = F_{zppf}\left(G(B,A),G_{zpf},G_{ppf}\right) = \frac{\frac{\hat{B}}{A}}{1 - \frac{\hat{B}}{A}\frac{A_{FB}}{\hat{B}}} \quad (3.9)$$

$$= \frac{\hat{B}}{A - A_{FB}} = \frac{\hat{B}}{\hat{A}} = \hat{G}\left(\hat{B},\hat{A}\right)$$

Remark 4: The above theorems indicate that pulling zeros can be achieved by feedforward path in open loops and pulling poles can be achieved by feedback path in closed loops.

Remark 5: It is noted that there is a full range within the unit circle to specify the stable MP plant $\hat{G}_p\left(\hat{B},\hat{A}\right)$, that is,

$$\begin{aligned} \left|b_0 z^m\right| \le \left|b_0 z^m + b_1 z^{m-1} + \ldots + b_m\right| \le \left|b_0\left(z+1\right)^m\right| \\ \left|z^n\right| \le \left|z^n + a_1 z^{n-1} + \ldots + a_n\right| \le \left|\left(z+1\right)^n\right| \end{aligned} \quad (3.10)$$

Compared with those zero/pole cancellation approaches explained above, this zero/pole pulling property gives much more flexibility in designing the consequent control systems and more robust against internal uncertainty and external disturbance.

Proposition: In converting NMP system models into MP system models, parallel pulling (through summation/subtraction) is much more robust than serial cancellation (through multiplication/division). To prove it, consider a stable NMP model

$$G = (z - z_1)\frac{B_1}{A_1} \quad (3.11)$$

where $(z - z_1)$ is a factor with zero amplitude outside the unit circle, that is $|z_1| \succ 1$ and $\frac{B_1}{A_1}$ is a stable MP model. Accordingly, with parallel pulling to convert the NMP model G into a MP model, it gives rise to

$$\hat{G} = (z - z_1 - \Delta_1)\frac{B_1}{A_1} - (z - z_1)\frac{B_1}{A_1} = (z - \Delta_1)\frac{B_1}{A_1} \quad (3.12)$$

where Δ_1 denotes inaccuracy. Clearly $\hat{G}$ is a stable MP model if and only if $0 \le |\Delta_1| \prec 1$. For serial cancellation it forms

$$\hat{G} = \frac{1}{(z - z_1 - \Delta_1)_1}(z - z_1)\frac{B_1}{A_1} = \frac{(z - z_1)}{(z - z_1 - \Delta_1)_1}\frac{B_1}{A_1} \quad (3.13)$$

Clearly $\hat{G}$ is a stable MP model if and only if $\Delta_1 = 0$. From the above analysis, it is clear that the pole/zero cancellation must be accurate and the zero pulling relocation has a unit circle tolerance $0 \le |\Delta_1| \prec 1$ to guarantee the MP conversion from NMP models.

4. U-control --- model-independent control system design

4.1 Control system design

Figure 4.1 shows the fundamental block diagram for U-model based control, U-control in short, of stable MP systems. Figure 4.2 shows the newly developed structure for U-control of unstable NMP systems. With reference to the block diagram in Figure 4.2, the whole control system design is split into two parallel steps of computations plus a connection of the design into an appropriate control system.

1) Conversion of a NMP G_p plant into a MP plant $\hat{G}_p$ by pulling filters as indicated by dotted block in the system.
2) Design of the invariant controller G_{c1} and the inversion of the converted MP $\hat{G}_p$ by U-model based root solving.
3) Building up the whole control system following the block diagram connection guidelines.

The above three step procedure is detailed below.

Step 1: Conversion of the NMP G_p plant into a MP plant $\hat{G}_p$

By the pulling mapping $G_p \xrightarrow{F_{zp}\left(G(B,A),G_{zpf},G_{ppf}\right)} \hat{G}_p$, design the pulling filters by (3.2) to (3.4), then determined the converted MP plant $\hat{G}_p$ in terms of the derived formulae (3.5) to (3.9).

Step 2: Design of U-control system $\sum = \left(F_{lfbc} \quad G_{c1} \quad G_{ip}\right)$

For simplicity consider this as single input $u \in \mathrm{R}$ and single output $y \in \mathrm{R}$ autonomous linear feedback control system. The triplet is defined as, F_{lfbc} is a linear feedback control framework, $G_{c1}: y \to u$ is the linear invariant controller, $G_{ip} = \hat{G}_P^{-1}\hat{G}_p = 1: u \to y$. As $\hat{G}_p$ has been converted into stable MP, the U-control pre-request, G_p^{-1} exists, is satisfied. Subsequently,

1) Specify a desired closed loop linear transfer function G with proper damping ratio, natural frequency, and steady stead gain. Determine the invariant controller G_{c1} by inversing the closed loop transfer function, that is, $G_{c1} = \dfrac{G}{1-G}$.
2) For getting the inverse of $\hat{G}_p$, convert $\hat{G}_p$ into U-model expression $\sum_{\text{U-polynomial model}}$ and then solve $v(t) - \sum_{\text{U-model}} = 0$ for controller output $u(t-1)$, where $v(t)$ is the invariant controller G_{c1} output. For linear plant,

the control output is particularly expressed as $u(t-1)=\frac{v(t)-\lambda_0(t)}{\lambda_1(t)}, \lambda_1(t)\neq 0$. Therefore, this makes $G_{ip}=\hat{G}_P^{-1}\hat{G}_p=1:u\rightarrow y$ satisfied, and further the specified closed loop performance G achieved.

Step 3: Formation of the who control system

Figure 4.2 shows the designed system structure for connection.

4.2 Control system analysis

Several properties are presented in relation to the U-control of the NMP systems.

1) Bounded-input, bounded-output (BIBO) stability of deterministic U-model control systems: With reference to Figure 4.1, The U-control system is BIBO stable and follows a bounded reference signal $w(t)$ properly while the following conditions are satisfied:

(i) Invariant controller G_{c1} is closed-loop stable that is all poles of the closed loop transfer function G are located with the unit circle.

(ii) The converted plant model $\hat{G}_p$ is a BIBO and MP.

(iii) The inverse of the plant model G_p^{-1} exits.

Similar property has been proved for U-control of stable MP systems before (Q. Zhu et al., 2018). For the unstable NMP system, while condition (ii) satisfied means condition (iii) satisfied as well. Therefore the claimed property holds.

2) Disturbance added at the output: The U-control system is BIBO stable and tracking the reference signal with a bounded error while the following conditions are satisfied:

(i) Invariant controller G_{c1} is closed loop stable.

(ii) The converted plant model $\hat{G}_p$ is a BIBO and MP.

(iii) The inverse of the plant model $\hat{G}_p^{-1}$ exits.

(iv) The disturbance $d(t)$ has upper bound.

Similarly the proof of the property follows that presented in the U-control of stable MP systems (Q. Zhu et al., 2018). The resultant output response with output added disturbance can be expressed below.

$$y=\frac{wG_{c1}}{1+G_{c1}}+\frac{d}{1+G_{c1}} \quad (4.1)$$

3) Robustness stability: Let the uncertainty Δ occurred in the product of $G_{ip} = \hat{G}_P^{-1}\hat{G}_p = \Delta$, from small gain theorem (Kravaris & Wright, 1989), it gives

$$\left|\frac{G_{c1}(j\omega)\Delta(j\omega)}{1+G_{c1}(j\omega)\Delta(j\omega)}\right| \prec \frac{1}{\lambda(j\omega)} \quad (4.2)$$

where $\lambda(j\omega)$ is an upper bound of the multiplicative uncertainty all over the system.

5. Simulation studies

The main purposes of the simulation studies include

1) Validate the effectiveness of the pulling principle in converting NMP plants into MP models
2) Validate the effectiveness of the integrated U-control of NMP systems
3) Compare with the other representative work with the proposed U-control approach
4) Show step by step U-control system design procedure with simulated case studies

Example 1: Control of rotary mechanical system --- NMP plant

Consider a rotary mechanical plant (Freeman* et al., 2005). It consists of inertias, dampers, torsional springs, a timing belt, pulleys, and gears in function blocks. The transfer function of the plant has been identified with

$$\frac{Y(s)}{U(s)} = G_p(s) = \frac{123.853\times 10^4(-s+3.5)}{(s^2+6.5s+42.25)(s+45)(s+190)} \quad (5.1)$$

where $Y(s)$ and $U(s)$ are the Laplace transforms of the angular position and driving torque respectively. Obviously this is a NMP plant as the zero of $s = 3.5$, but stable as all the four poles are on the left-half of the s-plane.

Test 1.1: U-control of NMP system

The main purpose of the original work (Freeman* et al., 2005) was to use the plant model for experimental evaluation of iterative learning control algorithms for non-minimum phase plants.

For designing the U-control system, convert the plant into its discrete time model with sampling interval $T_s = 0.1\sec$, it gives the Z transfer function as

$$G_p(z) = \frac{B(z)}{\hat{A}(z)} = \frac{-6.69z^3 + 7.856z^2 + 2.392z + 0.001445}{z^4 - 1.233z^3 + 0.5356z^2 - 0.005799z + 3.249e-11} \quad (5.2)$$

which the zeros locate at $z_1 = 1.4252$, $z_2 = -0.2504$ and $z_3 = -0.0006$ in the numerator polynomial B, and poles $p_1 = 0.6109 + 0.3857i$, $p_2 = 0.6109 - 0.3857i$, and $p_3 = 0.0111$ in the denominator polynomial $\hat{A}$. Obviously, this is a NMP plant because of $z_1 = 1.4252$ outside of the unit circle.

Design step 1: Convert NMP plant into an assigned MP plant

To deal with the NMP plant of (5.2), assign a desired casual MP plant model $\hat{G}_p$

$$\hat{G}_p(z) = \frac{\hat{B}(z)}{\hat{A}(z)} = \frac{-6.69z^3 + 1.666z^2 + 0.8386z + 0.0005026}{z^4 - 1.233z^3 + 0.5356z^2 - 0.005799z + 3.249e-11} \quad (5.3)$$

where all the zeros and poles are located within the unit circle, where $z_{d1} = 0.5$, $z_{d2} = -0.2504$ and $z_{d3} = -0.0006$, and poles $p_{d1} = 0.6109 + 0.3857i$, $p_{d2} = 0.6109 - 0.3857i$, and $p_{d3} = 0.0111$.

To force the NMP plant (5.2) converted to the desired MP model (5.3), design the pulling filter as

$$G_{zpf}(z) = G_p(z) - \hat{G}(z) = \frac{6.19z^2 + 1.5534z + 0.0009424}{z^4 - 1.233z^3 + 0.5356z^2 - 0.005799z + 3.249e-11} \quad (5.4)$$

Accordingly, find out feedforward parameters, which can pull the plant model's zeros back to unit circle as specified in the desired plant model (5.3).

Design step 2: Determine the invariant controller G_{c1} and the MP plant inverter $\hat{G}_p^{-1}$

Specify a linear feedback control system structure with the invariant controller G_{c1} and unite constant plant $G_{ip} = 1$. The desired closed loop transfer function is specified with

$$G_d(z) = \frac{0.1761z^{-1}}{1 - 1.3205z^{-1} + 0.4966z^{-2}} \quad (5.5)$$

This a representative of decayed oscillatory response (0.7/damping ratio and 1/undamped natural frequency) and zero steady error (Karl J Åström & Wittenmark, 2013). According to the design procedure explained in Section 4.1, determine the invariant controller as

$$G_{c1}(z) = \frac{G_d(z)}{1 - G_d(z)} = \frac{0.1761z^{-1}}{1 - 1.4966z^{-1} + 0.4966z^{-2}} \quad (5.6)$$

For obtaining the inverse $\hat{G}_p^{-1}$, considering $G_{ip} = \hat{G}_P^{-1}\hat{G}_p = 1 : u \to y$. First of all, write $\hat{G}_p$ into U-model of

$$\hat{y}(t) = \hat{\lambda}_0(t) + \hat{\lambda}_1(t)u(t-1) \quad (5.7)$$

where

$$\begin{aligned} \hat{\lambda}_0(t) &= 1.233y(t-1) - 0.5356y(t-2) + 0.005799y(t-3) - 3.249e-11y(t-4) + \\ &\quad 1.666u(t-2) + 0.8386u(t-3) + 0.0005026u(t-4) \\ \hat{\lambda}_1(t) &= -6.69 \end{aligned} \quad (5.8)$$

Secondly let the desired output be $y_d(t)$ (that is, the invariant controller output), then determine $\hat{G}_p^{-1}$ output $u(t-1)$ by

$$u(t-1) = \frac{\hat{y}_d(t) - \hat{\lambda}_0(t)}{\hat{\lambda}_1(t)} \quad (5.9)$$

Design step 3: Form the control system following the integrated control system structure as shown in Figure 4.2.

Figure 4.2 shows the integrated control system structure (deleting the pole pulling filter G_{ppf} as it is not used in this case). Figure 5.1 shows the simulated results.

Compared with the Iterative Learning Control (ILC) approaches (Freeman* et al., 2005), U-control design is straightforward without requiring repetitive stimulation, selecting gain with heuristic experience, and finding phase lead factor λ from determining maximum impulse response peak. However U-control is relative new and just initially tested for NMP plants in this journal level study.

Test 1.2: U-control of uncertain NMP system

Still consider the rotary plant to be controlled, but the plant model is assumed now changing to a nominal model of the plant (Wang & Su, 2015)

$$\frac{Y(s)}{U(s)} = G_{np}(s) = \frac{144.86(-s+3)}{(s^2+6.5s+42.25)} \quad (5.10)$$

being selected for the robustness tests with the U-control system designed in **Test 1.1**, that is, there exist uncertainty in the plant model against a designed control system. Convert the nominal model into its discrete time model with sampling interval $T_s = 0.1\text{sec}$, it gives the Z transfer function as

$$G_{np}(z) = \frac{-8.211z + 11.3}{z^2 - 1.222z + 0.522} \quad (5.11)$$

as the zero located outside the unit circle at $z = 1.3757$. It should be noted the NMP model in Test 1.1 has its unstable zero at $z = 1.4252$. If zero-pole cancellation scheme used, this mismatched is still a NMP model and it could cause unstable problem in control system operation. The system response performance is shown in Figure 5.2.

The system performance is very similar to that obtained from disturbance observer (DOB) based control scheme (Wang & Su, 2015). In regarding to the comparative approach (Wang & Su, 2015), which a robust DOB based control structure is proposed to stable NMP systems, particularly compensate the uncertain plant into a nominal one. This design procedure takes a set of sophisticated tools such as pre-filtering, DOB formation/design, H_∞ based formulation/optimal solution, virtual controller design. In contrast, by U-model scheme, it is independent of designing the invariant controller by inverting a specified closed loop transfer function, from converting NMP to MP. That is, control system design and NMP model conversion are separately conducted, and then connected/integrated by an inverter of the converted MP model (in terms of U-model) and the invariant controller designed within a linear feedback control loop.

Example 2: Control of unstable NMP process

Consider a delayed process with two unstable poles, which has been used as a bench test of some approaches in control of NMP systems (García et al., 2006; Liu et al., 2005).

$$G_p(s) = \frac{2e^{-03s}}{(3s-1)(s-1)} = \frac{2e^{-03s}}{3s^2-4s+1} \quad (5.12)$$

For $T_s = 0.1\text{sec}$, the discretised plant model is

$$G_p(z) = z^{-3} \times \frac{0.01763z + 0.01822}{z^2 - 2.139z + 1.143} \quad (5.13)$$

It becomes unstable NMP plant model with a zero $z_1 = -1.0335$ and two poles $p_1 = 1.0983$ and $p_2 = 1.0407$, which has been a typical issue related to zeros going out the unit circle in sampled systems, induced by some sampling rates (K. J. Åström et al., 1984).

Test 2.1: U-control compared with two degree-of freedom control --- Tao-control (Liu et al., 2005)

For U-control, following the step by step procedure, it gives

Design step 1: Convert unstable NMP plant into an assigned stable MP plant

Assign the corresponding converted MP plant model as

$$\hat{G}_p(z) = \frac{0.1188z + 0.0819}{z^2 - 1.1z + 0.3} \quad (5.14)$$

To achieve the assigned $\hat{G}_p$, break zero pulling filter into two parts, one for eliminating the pure input delay, and the other for pulling the zero back into the unit circle.

$$G_{zpf1}(z) = \frac{-0.1188z - 0.0819}{z^2 - 2.139z + 1.143} \quad (5.15)$$

$$G_{zpf2}(z) = z^{-3} \times \frac{0.01763z + 0.01822}{z^2 - 2.139z + 1.143} \quad (5.16)$$

And the pole pulling filter is derived as below

$$G_{ppf}(z) = \frac{1.039z - 0.843}{0.1188z + 0.0819} \quad (5.17)$$

Design step 2: Determine the invariant controller G_{c1} and the inverter $\hat{G}_p^{-1}$

To achieve a specified close-loop transfer function performance, it assigns

$$G_d(z) = \frac{0.03551z + 0.02448}{z^2 - 1.5z + 0.56} \quad (5.18)$$

Then inverse the close-loop transfer function G_d within a linear feedback loop to obtain the invariant controller

$$G_{c1} = \frac{G_d}{1 - G_d} = \frac{0.03551z + 0.02448}{z^2 - 1.53551z + 0.53552} \quad (5.19)$$

For obtaining the inverse $\hat{G}_p^{-1}$ in terms of $G_{ip} = \hat{G}_P^{-1}\hat{G}_p = 1 : u \to y$. First, convert $\hat{G}_p$ into U-model of

$$\hat{y}(t) = \hat{\lambda}_0(t) + \hat{\lambda}_1(t)u(t-1) \quad (5.20)$$

where

$$\hat{\lambda}_0(t) = 1.1y(t-1) - 0.3y(t-2) + 0.0819u(t-2)$$
$$\hat{\lambda}_1(t) = 0.1188 \quad (5.21)$$

Secondly let the desired output be $y_d(t)$ (that is, the invariant controller output), then determine $\hat{G}_p^{-1}$ output $u(t-1)$ by

$$u(t-1) = \frac{\hat{y}_d(t) - \hat{\lambda}_0(t)}{\hat{\lambda}_1(t)} \quad (5.22)$$

Consequently, working out $\hat{G}_p^{-1}$ is a process to obtain the controller output $u(t-1)$ for a given $y_d(t)$ through the derived U-model.

Design step 3: Form the control system following the integrated control system structure as shown in Figure 4.2.

For the two degree-of freedom control (Liu et al., 2005), just use its designed system to generate the simulated system output response.

Figure 5.3 shows both simulated results, which give similar performance. Tao-control requires three designs of setpoint tracking controller, disturbance estimator, and stabilising controller plus online tuning rule. U-control requires three designs of zero pulling filter (pulling unstable zeros into unit cycle and cancel time delay), pole pulling filter (pulling unstable poles into the unit cycles), and invariant controller (specified the whole closed loop performance). Tao-control is relative popular, but need high skills in design and online tuning. U-control is just appeared and easy to design with basic control system design skills.

Test 2.2: U-control compared with Tao-control (Liu et al., 2005) with plant model uncertainty

While there exist modelling errors, such as an example (García et al., 2006)

$$\begin{aligned} G_r(s) &= G_p(s)\left(1 + W_m(s)\right) \\ &= G_p(s)\left(1 + \frac{0.5}{0.1s+1}\right) = G_p(s)\left(\frac{0.1s+1.5}{0.1s+1}\right) \end{aligned} \quad (5.23)$$

Tao's method is unstable, because its robust stability condition (García et al., 2006) is not held,

$$\left\| H_1 \frac{KG}{1+KG} W_m \right\|_\infty > 1 \quad (5.24)$$

Correspondingly, in U-model based methods, the real plant become

$$G_r(z) = G_p(z)(1 + W_m(z)) = G_p(z)\left(\frac{z - 0.05182}{z - 0.3679}\right) \quad (5.25)$$

Apply all designed U-control in test 2.1 to the uncertain plant, it still gives stable response from generated plots.

Figure 5.4 shoes both system response performance.

Example 3: Comparison with Linear-quadratic-Gaussian (LQG) control of autopilot of Boeing 747

Consider an altitude-hold flight of an autopilot of Boeing 747 (Franklin et al., 2014). The dynamics of the system regarding to height h and elevator δe has been modelled with the following transfer function

$$\frac{h(s)}{\delta e(s)} = \frac{32.7(s+0.0045)(s+5.64)(s-5.61)}{s(s+2.25 \pm j2.99)(s+0.0105)(s+0.0531)} \quad (5.26)$$

Discretise the transfer function with sampling interval $T_s = 0.05\sec$, it gives

$$\frac{h(z)}{\delta e(z)} = \frac{0.03759z^4 - 0.08083z^3 + 0.008311z^2 + 0.06973z - 0.03481}{z^5 - 4.764z^4 + 9.088z^3 - 8.68z^2 + 4.152z - 0.796} \quad (5.27)$$

To factorise both numerator and denominator polynomials respectively, it gives zeros at ($z_1 = -0.9276$, $z_2 = 1.3246$, $z_3 = 0.9984$, $z_4 = 0.7549$) and poles at ($p_1 = 0882 + 0.1344j$, $p_2 = 0.882 - 0.1344j$, $p_3 = 1$, $p_4 = 1$, and $p_5 = 1$) respectively. Obviously this is a NMP plant because of the zero $z_2 = 1.3246$ outside the unit circle. In addition, the triple poles p_3, p_4, p_5 on the unit circle are not also favourable, and should be pulled back inside of the unit circle.

3.1 U-control

Design step 1: Convert NMP plant into an assigned MP plant

To deal with the NMP plant, assign a desired casual MP plant model $\hat{G}_p$ as

$$\hat{G}_p = \frac{0.0168z^3 - 0.0126z^2 + 0.0034z - 0.004}{z^5 - 3z^4 + 3.55z^3 - 2.07z^2 + 0.5944z - 0.0672} \quad (5.28)$$

where all the zeros and poles are located within the unit circle, in which the zeros are at $z_{d1} = 0$, $z_{d2} = 0.3849$ $z_{d3} = 0.1825 + 0.1689j$, and $z_{d4} = 0.1825 - 0.1689j$, and the poles are at $p_{d1} = 0.8$, $p_{d2} = 0.7$, $p_{d3} = 0.6$ and $p_{d4} = 0.5$, and $p_{d5} = 0.4$ respectively.

To force the NMP plant converted to the desired MP model, design the zero pulling filter as

$$G_{zpf}(z) = G_p(z) - \hat{G}(z) = \frac{0.0208z^4 - 0.0682z^3 + 0.0049z^2 - 0.3698z - 0.0348}{z^5 - 4.764z^4 + 9.088z^3 - 8.68z^2 + 4.152z - 0.796} \quad (5.29)$$

will construct the desired plant located zeros at $z_1 = 0$, $z_2 = 0.3849$ $z_3 = 0.1825 + 0.1689j$, and $z_4 = 0.1825 - 0.1689j$.

Correspondingly, with reference to (3.7), design the pole pulling filter as

$$G_{ppf}(z) = \frac{1.764z^4 - 5.538z^3 + 6.61z^2 - 3.5576z + 0.7288}{0.0168z^4 - 0.0126z^3 + 0.0034z^2 + 0.0004z} \quad (5.30)$$

Design step 2: Determine the invariant controller G_{c1} and the inverter $\hat{G}_p^{-1}$

Specify a linear feedback control system structure with the invariant controller G_{c1} and unite constant plant $G_{ip} = 1$. The desired closed loop transfer function is specified with

$$G_d(z) = \frac{0.1761z^{-1}}{1 - 1.3205z^{-1} + 0.4966z^{-2}} \quad (5.31)$$

This is a representative of decayed oscillatory response (0.7/damping ratio and 1/undamped natural frequency) and zero steady state error. According to the design procedure explained in Section 4.1, determine the invariant controller as

$$G_{c1}(z) = \frac{G_d(z)}{1 - G_d(z)} = \frac{0.1761z^{-1}}{1 - 1.4966z^{-1} + 0.4966z^{-2}} \quad (5.32)$$

For obtaining the inverse $\hat{G}_p^{-1}$, It can be determined in terms of $G_{ip} = \hat{G}_P^{-1}\hat{G}_p = 1 : u \to y$. First of all, write $\hat{G}_p$ into U-model of

$$\hat{y}(t) = \hat{\lambda}_0(t) + \hat{\lambda}_1(t)u(t-1) \quad (5.33)$$

where

$$\hat{\lambda}_0(t) = 3y(t-1) - 3.55y(t-2) + 2.07y(t-3) - 0.5944y(t-4) + 0.0672y(t-5) - 0.0126u(t-2) + 0.0034u(t-3) - 0.0004u(t-4) \quad (5.34)$$

$$\hat{\lambda}_1(t) = 0.0168$$

Secondly let the desired output be $y_d(t)$ (that is, the invariant controller output), then determine $\hat{G}_p^{-1}$ output $u(t-1)$ by

$$u(t-1) = \frac{\hat{y}_d(t) - \hat{\lambda}_0(t)}{\hat{\lambda}_1(t)} \quad (5.35)$$

Design step 3: Form the control system structure following the integrated control system structure as shown in Figure 4.2.

3.2 LQG control

Following a classical reference (Franklin et al., 2014), the plant state space description was given by

$$A = \begin{bmatrix} -0.00643 & 0.0263 & 0 & -32.2 & 0 \\ -0.0941 & -0.624 & 820 & 0 & 0 \\ -0.000222 & -0.00153 & -0.668 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & -1 & 0 & 830 & 0 \end{bmatrix},$$

$$B = \begin{bmatrix} 0 \\ -32.7 \\ -2.08 \\ 0 \\ 0 \end{bmatrix} \text{ and } C = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (5.36)$$

By MATLAB LQG toolbox, specify the feedback controller in form of $u = -Kx$, accordingly obtain the Kalman gain $L = \begin{bmatrix} -3.344 & 650.943 & 1.126 & 0.922 & 15.123 \end{bmatrix}$ and the feedback gain matrix $K_{LQG} = \begin{bmatrix} -4 & 2.7 & -112.6 & -4899.1 & -3.2 \end{bmatrix}$.

Figure 5.5 shows the computational experimental results obtained from the two approaches.

The comparison of the two approaches includes

1) Design procedure, for a specified system response, U-control is once off transparent design, LQG can achieve the same response as the U-control, but it has to using trial and error in many times to find out the design parameters (K_{LQG} and L). LQG need computer added design package because of the computational burden/complexity. This is a particular difficulty to users. In this design U-control only requires output feedback, LQG requires state feedback,

which a state observer should have been designed if full state variables are not measureable. However, U-control, not as popular as LQG, need more study to provide comprehensive understanding and formulations.

2) Simulated results, as mentioned above, LQG can achieve the same system output response as U-control, but need many times of trial and errors. LQG requires less controller output power than U-control, because its required output response is slower than U-control.

6. Conclusions

Analytically this study has developed a comprehensive framework to provide a new insight and procedure in dealing with unstable NMP control systems. Correspondingly this study has provided a number of simulated bench tests numerically from various aspects to demonstrate the analytical system configuration and formulations. There are several conclusive points as summarised below.

1) Pulling principle can be largely applied to accommodate zero/pole outside the unit circle, therefore systematically convert unstable NMP plants into pre-assigned MP models, then U-control based control is applied to design the control systems within a closed loop feedback framework compared with some of the other representative approaches.
2) The proposed separation designs have made the U-control of unstable NMP systems more concise, effective and robust to uncertainties compared with pole/zero cancelation approaches.
3) Although it is not completely understood yet, the combination of pulling principle and U-control based design approach might even provide new understanding/solution to zero dynamic issues in nonlinear control system design.

References

Åström, K. J., Hagander, P., & Sternby, J. (1984). Zeros of sampled systems. *Automatica*, *20*(1), 31–38. https://doi.org/10.1016/0005-1098(84)90062-1

Åström, Karl J, & Wittenmark, B. (2013). *Adaptive control*. Courier Corporation.

Babík, Z., & Dostál, P. (2012). Hammerstein and Wiener Models in nonlinear control of servo-speed mechanism AMIRA DR300. *Proceedings in ARSA-Advanced Research in Scientific Areas*, (1).

Billings, S. A. (2013). *Nonlinear system identification: NARMAX methods in the time, frequency, and spatio-temporal domains*. John Wiley & Sons.

Butterworth, J. A., Pao, L. Y., & Abramovitch, D. Y. (2008). Architectures for tracking control in atomic force microscopes. *IFAC Proceedings Volumes (IFAC-PapersOnline)*, *17*(1 PART 1). https://doi.org/10.3182/20080706-5-KR-1001.0876

Chong, E. K. P., & Zak, S. H. (2011). An Introduction to Optimization: Third Edition. In *An Introduction to Optimization: Third Edition*. https://doi.org/10.1002/9781118033340

Do, M. T., Man, Z., Jin, J., Zhang, C., Zheng, J., & Wang, H. (2016). Sliding mode learning control of non-minimum phase nonlinear systems. *International Journal of Robust and Nonlinear Control*, *26*(11), 2281–2298.

Franklin, G. F., Powell, J. Da, & Emami-Naeini, A. (2014). *Feedback control of dynamic systems*. Prentice Hall Press.

Freeman*, C. T., Lewin, P. L., & Rogers, E. (2005). Experimental evaluation of iterative learning control algorithms for non-minimum phase plants. *International Journal of Control*, *78*(11), 826–846.

García, P., Albertos, P., & Hägglund, T. (2006). Control of unstable non-minimum-phase delayed systems. *Journal of Process Control*, *16*(10), 1099–1111. https://doi.org/10.1016/j.jprocont.2006.06.007

Geng, X., Zhu, Q., Liu, T., & Na, J. (2019). U-model based predictive control for nonlinear processes with input delay. *Journal of Process Control*, *75*, 156–170.

Gerald, C. F. (2004). *Applied numerical analysis*. Pearson Education India.

Gross, E., Tomizuka, M., & Messner, W. (1994). Cancellation of discrete time unstable zeros by feedforward control. *Journal of Dynamic Systems, Measurement, and Control*, *116*, 33–38. https://doi.org/10.1115/1.2900678

Haack, B., & Tomizuka, M. (1991). The Effect of Adding Zeroes to Feedforward Controllers. *Journal of Dynamic Systems Measurement and Control*, *113*(March), 6. https://doi.org/10.1115/1.2896362

Kravaris, C., & Wright, R. A. (1989). Deadtime compensation for nonlinear processes. *AIChE Journal*, *35*(9), 1535–1542.

Liu, T., Zhang, W., & Gu, D. (2005). Analytical design of two-degree-of-freedom control scheme for open-loop unstable processes with time delay. *Journal of Process Control*, *15*(5), 559–572.

Mirkin, B., Gutman, P.-O., & Shtessel, Y. (2014). Decentralized continuous MRAC with local asymptotic sliding modes of nonlinear delayed interconnected systems. *Journal of the Franklin Institute*, *351*(4), 2076–2088.

Mirkin, B., Gutman, P. O., & Shtessel, Y. (2012). Coordinated decentralized sliding mode MRAC with control cost optimization for a class of nonlinear systems. *Journal of the Franklin Institute*, *349*(4), 1364–1379.

Patil, M., Bandyopadhyay, B., Chalanga, A., & Arya, H. (2018). Output Tracking of Nonminimum-Phase Systems via Reduced-Order Sliding-Mode Design. *IEEE/ASME Transactions on Mechatronics*, *23*(4), 1981–1992.

Proakis, J. G., & Manolakis, D. G. (1988). *Introduction to digital signal processing*. Prentice Hall Professional Technical Reference.

Qiu, J., Delshad, I. S., Zhu, Q., Nibouche, M., & Yao, Y. (2017). A U-model based controller design for non-minimum phase systems: Application to Boeing 747 altitude-hold autopilot. *2017 9th International Conference on Modelling, Identification and Control (ICMIC)*, 122–127. IEEE.

Rigney, B. P., Pao, L. Y., & Lawrence, D. A. (2010). Nonminimum phase adaptive inverse control for settle performance applications. *Mechatronics*, *20*(1), 35–44. https://doi.org/10.1016/j.mechatronics.2009.06.007

Slotine, J.-J. E., & Li, W. (1991). *Applied nonlinear control* (Vol. 199). Prentice hall Englewood Cliffs, NJ.

Sun, X., Shi, Z., Chen, L., & Yang, Z. (2016). Internal model control for a bearingless permanent magnet synchronous motor based on inverse system method. *IEEE Transactions on Energy Conversion*, *31*(4), 1539–1548.

Tan, W., Marquez, H. J., & Chen, T. (2003). IMC design for unstable processes with time delays. *Journal of Process Control*, *13*(3), 203–213.

Tomizuka, M. (1987). Zero phase error tracking algorithm for digital control. *Journal of Dynamic Systems, Measurement, and Control*, *109*(December), 65–68. https://doi.org/10.1115/1.3143822

Wang, L., & Su, J. (2015). Disturbance rejection control for non-minimum phase systems with optimal disturbance observer. *ISA Transactions*, *57*, 1–9. https://doi.org/10.1016/j.isatra.2014.08.003

Xu, J.-X., & Tan, Y. (2003). *Linear and nonlinear iterative learning control* (Vol. 291). Springer.

Zhu, Q., & Guo, L. Z. (2002). A pole placement controller for non-linear dynamic plants. *Proceedings of the I MECH E Part I Journal of Systems*, *216*(6), 467–476. https://doi.org/10.1243/095965102321042227

Zhu, Q., Liu, L., Zhang, W., & Li, S. (2018). Control of complex nonlinear dynamic rational systems. *Complexity*, *2018*.

Zhu, Q. M., Zhao, D. Y., & Zhang, J. (2016). A general U-block model-based design procedure for nonlinear polynomial control systems. *International Journal of Systems Science*, *47*(14), 3465–3475.

Zhu, Q., Zhang, W., Zhang, J., & Sun, B. (2019). U-neural network-enhanced control of nonlinear dynamic systems.

Neurocomputing.

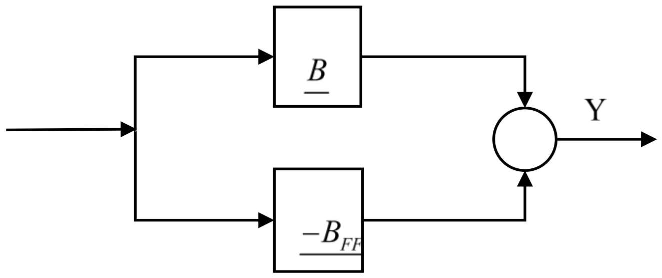


Figure 3.1(a) Zero relocation

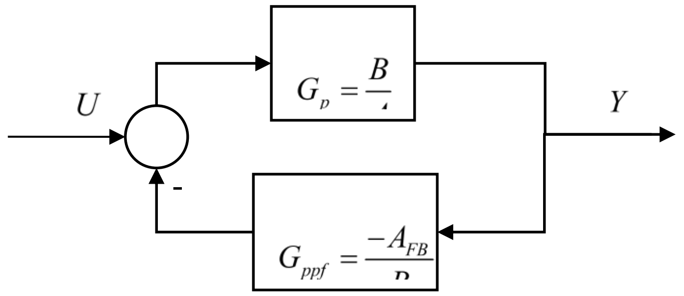


Figure 3.1(b) Pole relocation

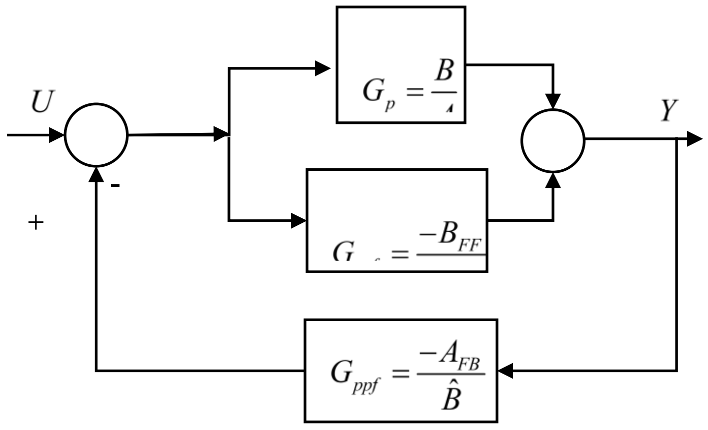


Figure 3.1(c) Zero and pole relocation

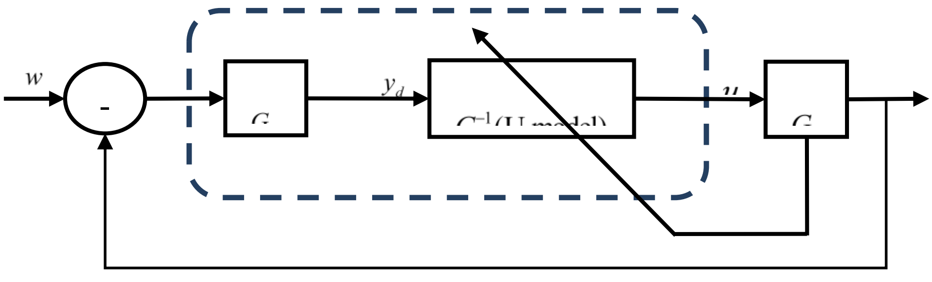


Figure 4.1 U-model based control of stable MP systems

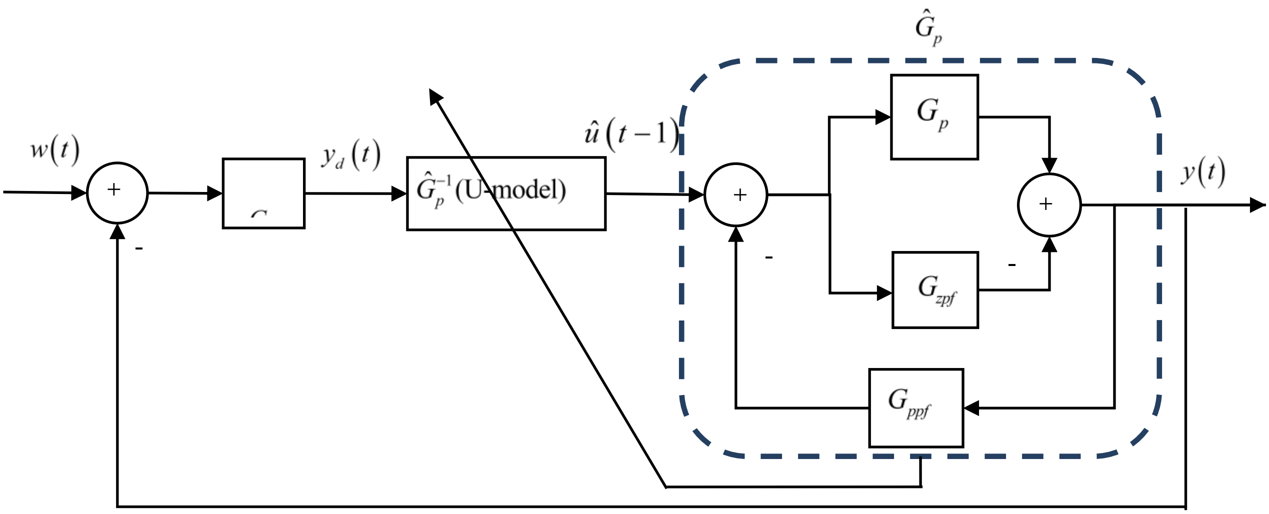


Figure 4.2 U-model enhanced control of NMP systems

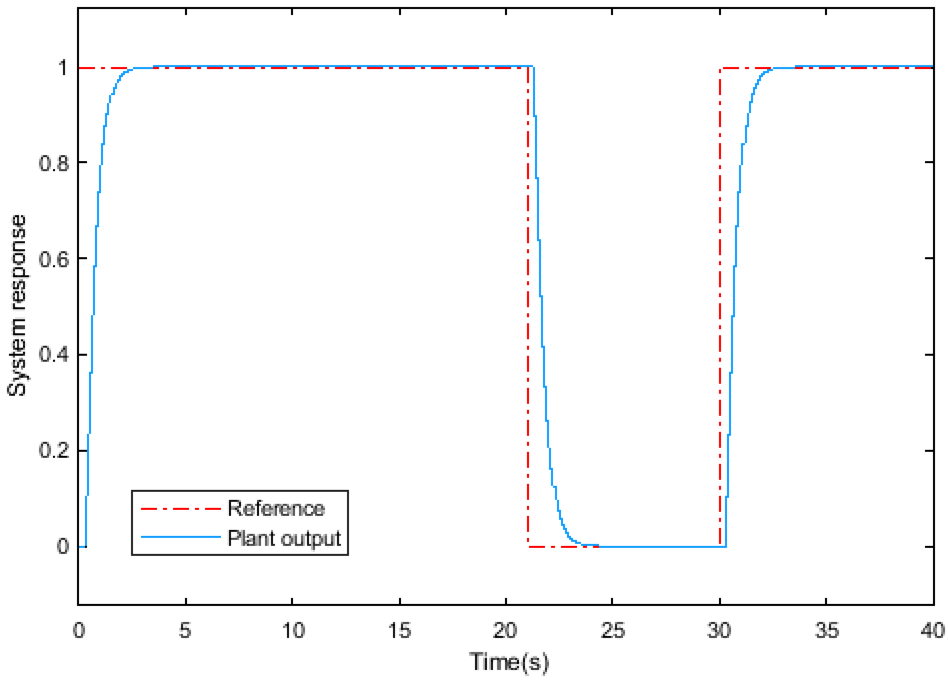


Figure 5.1(a) System response of test 1.1 of example 1

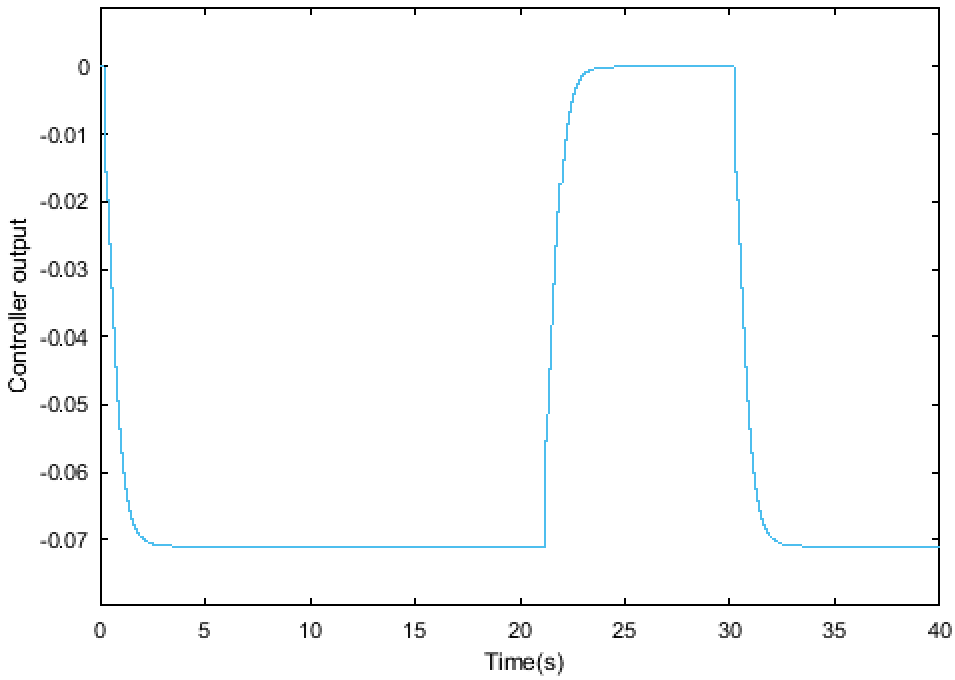


Figure 5.1(b) U-controller output in test 1.1 of example 1

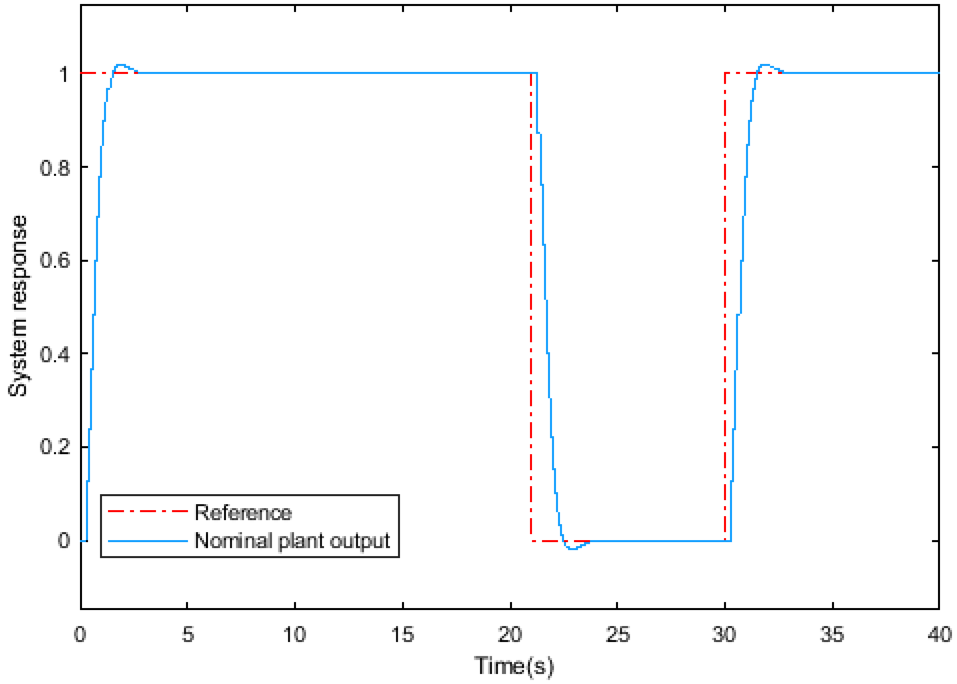


Figure 5.2(a) System response of test 1.2 of example 1

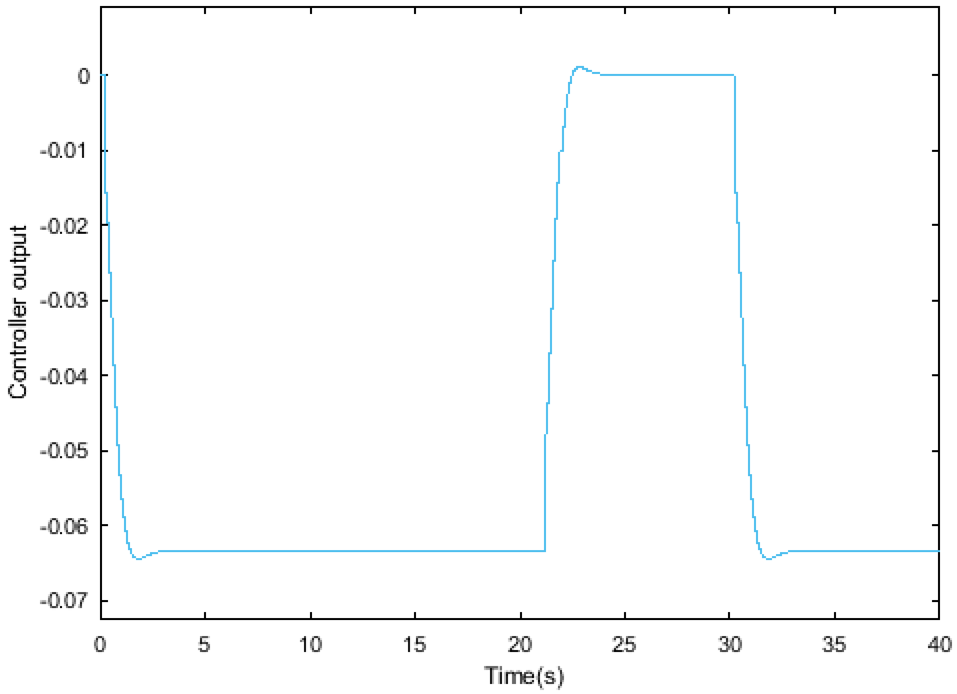


Figure 5.2(b) U-controller output of test 1.2 of example 1

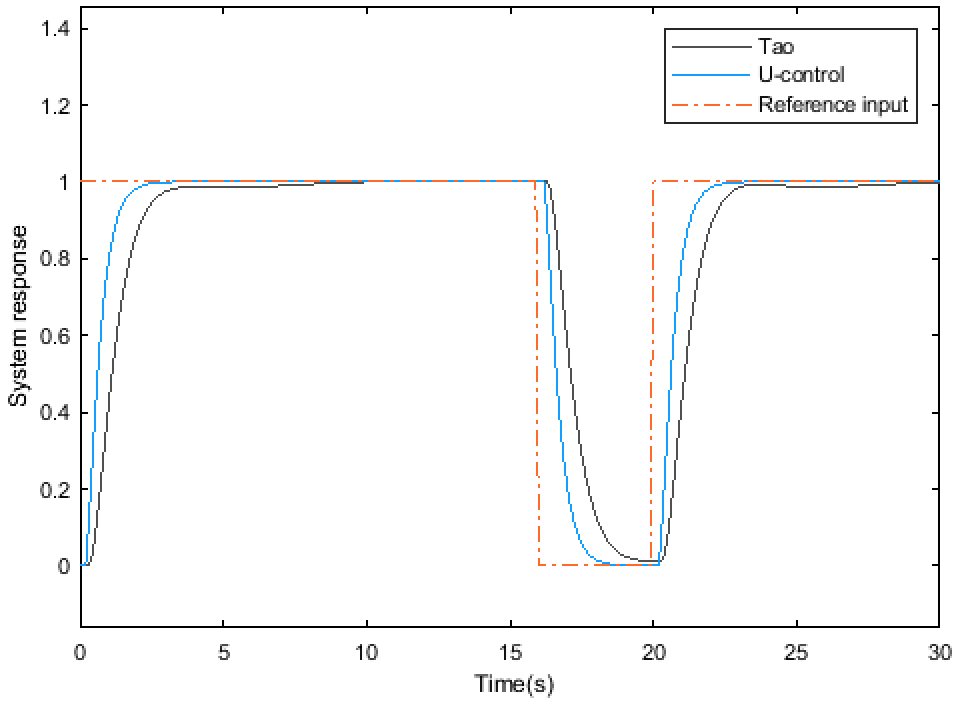


Figure 5.3 (a) System responses of test 2.1 of example 2

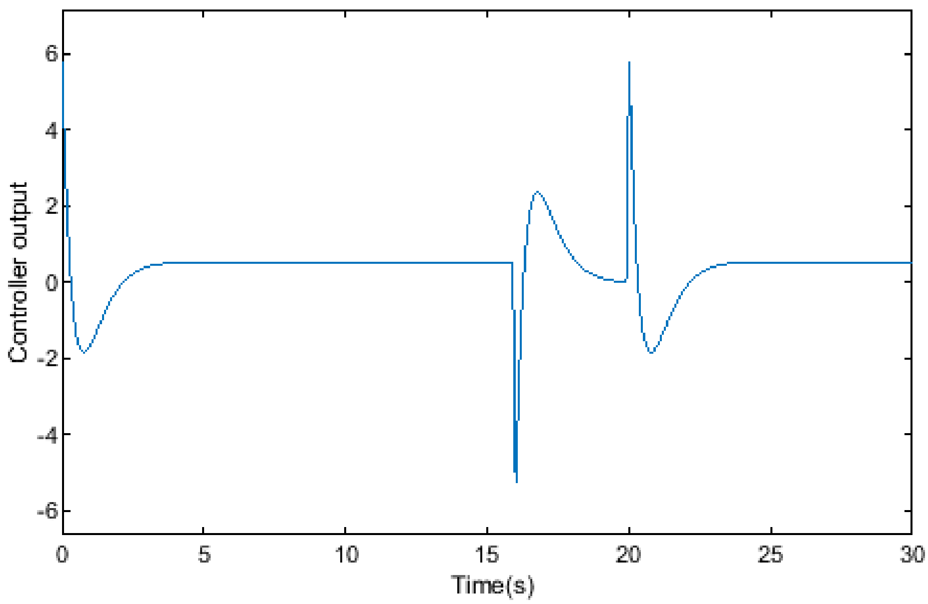


Figure 5.3(b) Tao-controller output of test 2.1 of example 2

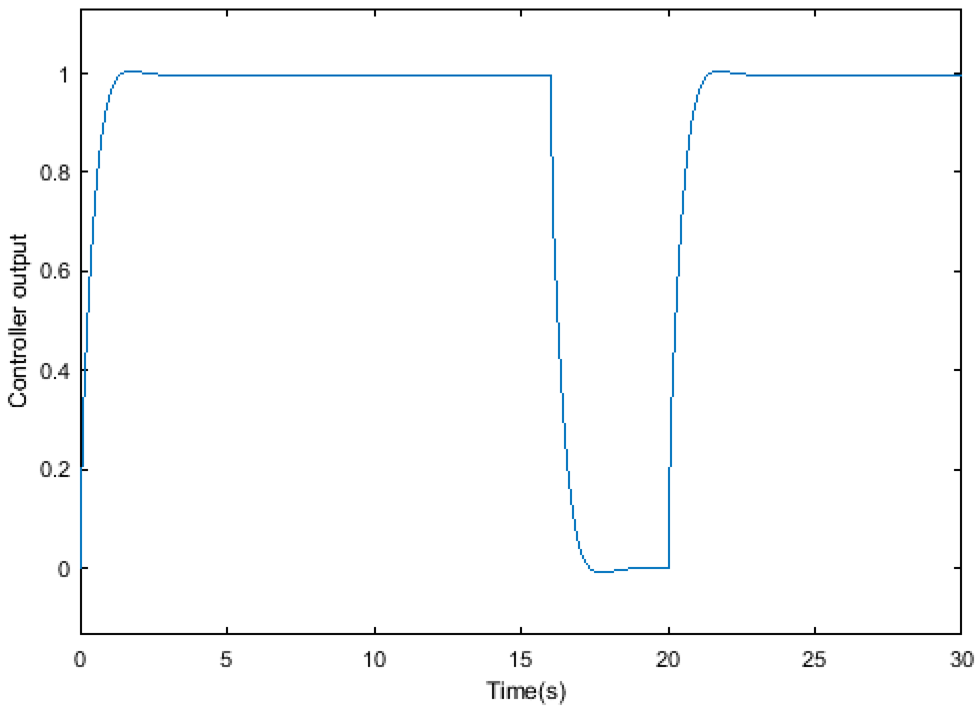


Figure 5.3(c) U-controller output of test 2.1 of example 2

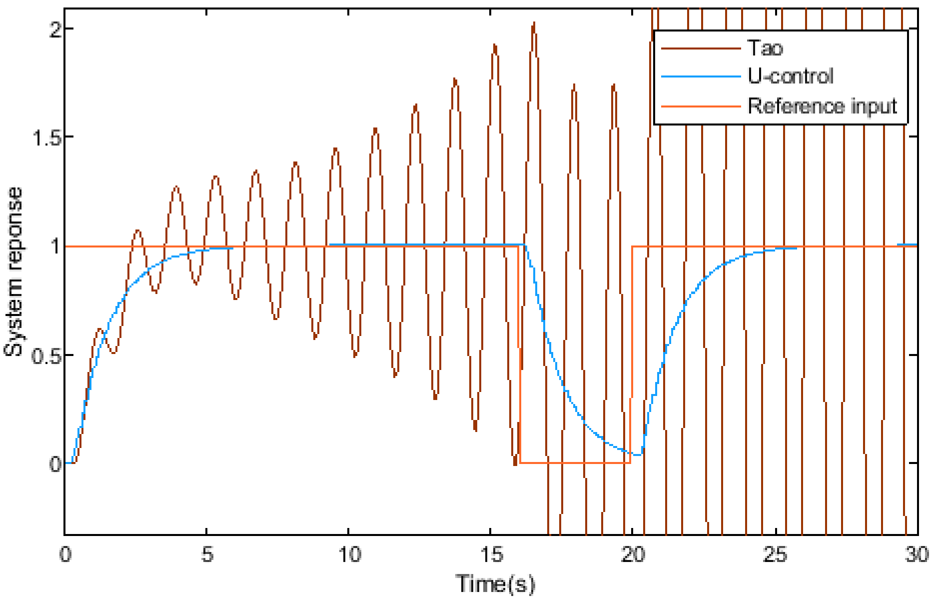


Figure 5.4(a) System responses with model uncertainty of test 2.2 of example 2

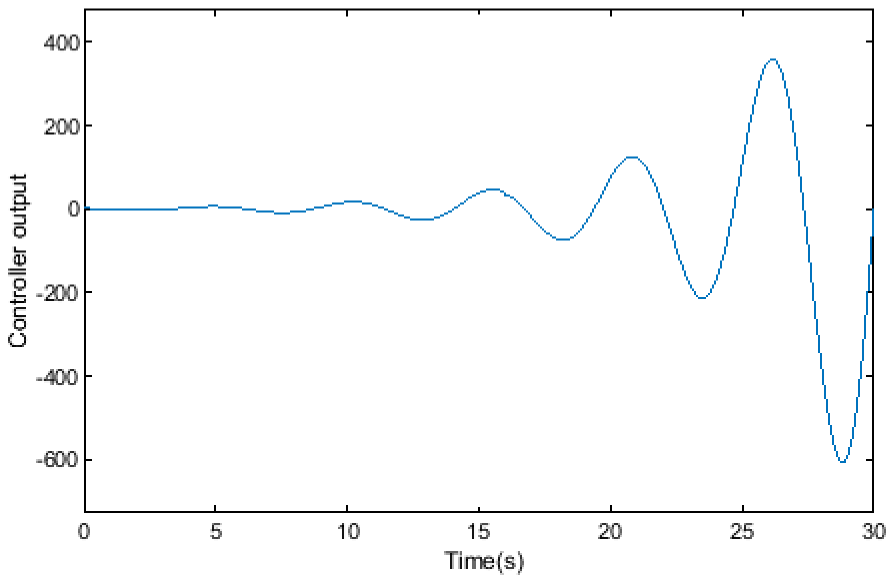


Fig 17 Figure 5.4(b) Tao-controller output with model uncertainty of test 2.2 of example 2

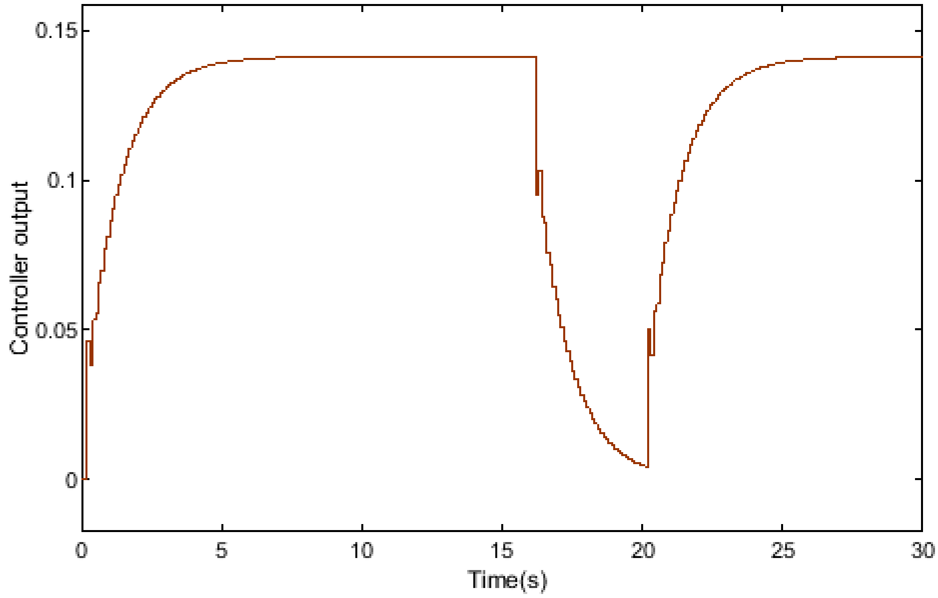


Figure 5.4(c) U-controller output with model uncertainty of test 2.2 of example 2

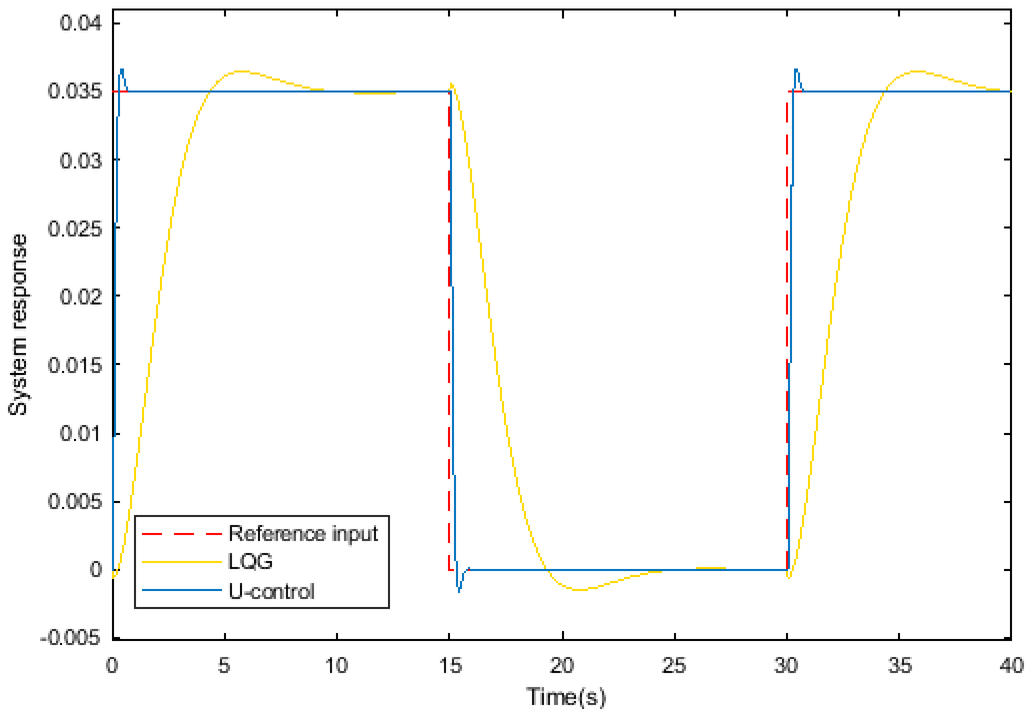


Figure 5.5 (a) Compared to LQG design

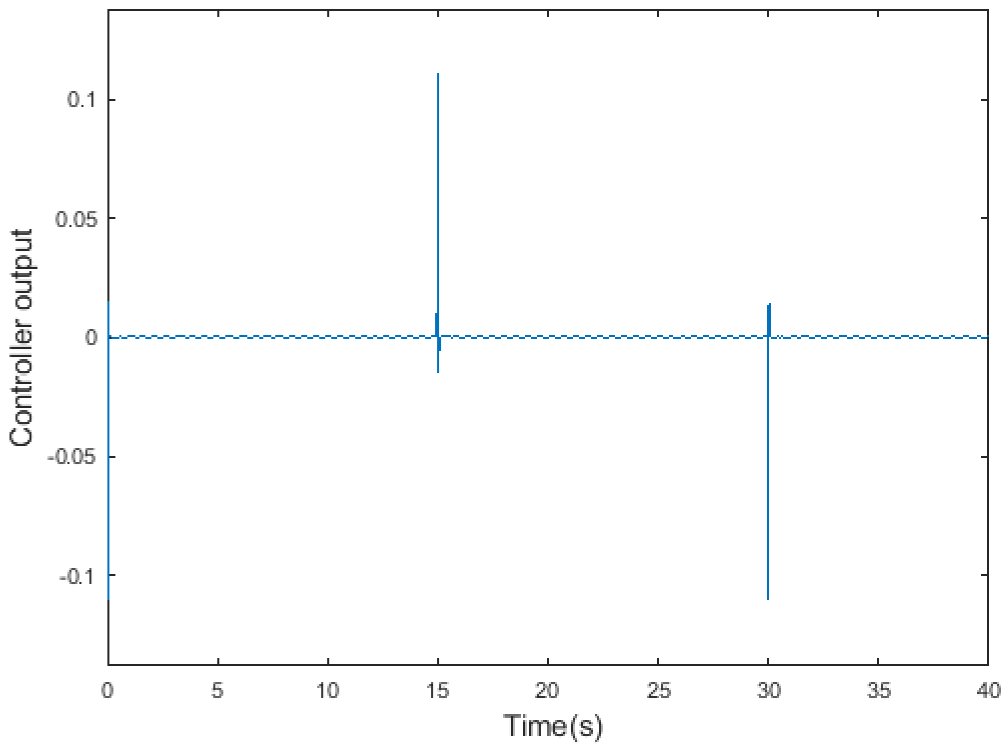


Figure 5.5 (b) LQG-controller output of example 3

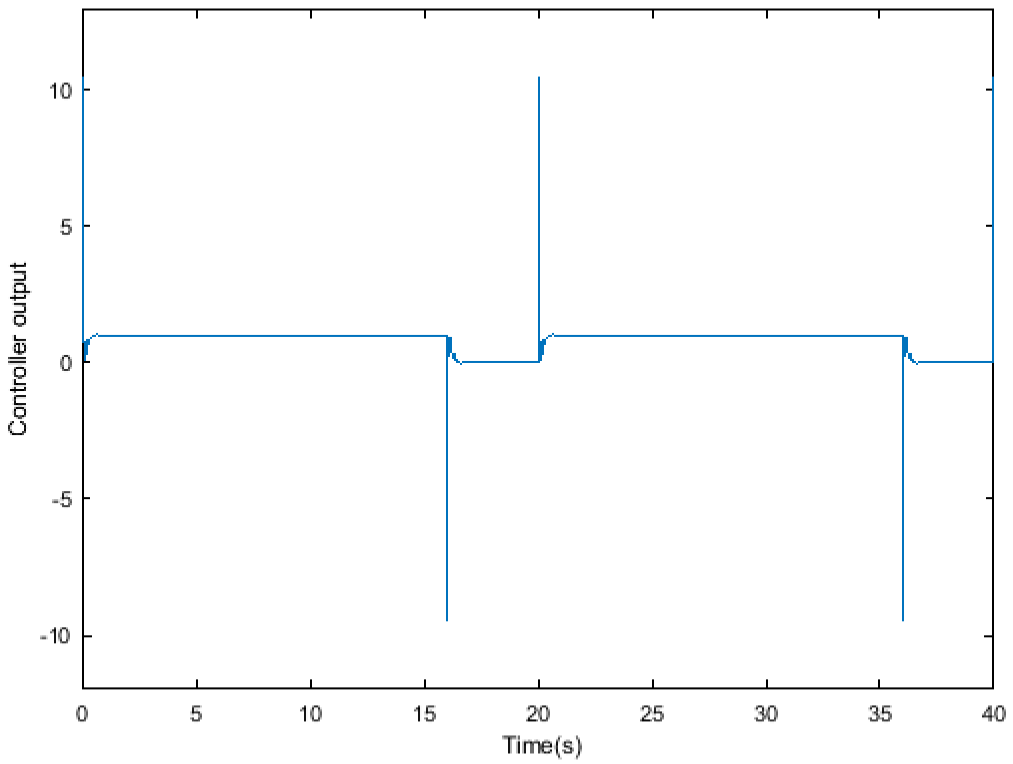


Figure 5.5 (c) U-controller output of example 3

U-Control: A model-independent control system design framework

Quanmin Zhu[1], Weicun Zhang[2*], Saleh Mobayen[3], Hao Yan[4], Ji Qiu[1], Pritesh Narayan[1]

1 University of the West of England, Coldharbour Lane, Bristol, BS16 1QY, UK
2 School of Automation and Electrical Engineering, University of Science and Technology Beijing, Beijing 10083, China
3Advanced Control Systems Laboratory, Department of Electrical Engineering, University of Zanjan, University Blvd., 45371-38791, Zanjan, Iran
4 School of Mechanical, Electronic and Control Engineering, Beijing Jiaotong University, Beijing 100044, China

Corresponding author: Weicun Zhang (e-mail: weicunzhang@ustb.edu.cn).

This work was supported in part by the Natural Science Foundation of China (NSFC) grants 61520106010. 61741302, and 51775032.

ABSTRACT This study presents the fundamental concepts and technical details of a U-model-based control (U-control for short) system design framework, including U-model realisation from classic model sets, control system design procedures, and simulated showcase examples. Consequently, the framework provides readers with clear understandings and practical skills for further research expansion and applications. In contrast to the classic model-based design and model-free design methodologies, this model-independent design takes two parallel formations: 1) it designs an invariant virtual controller with a specified closed loop transfer function in a feedback control loop, and 2) it determines the real controller output by resolving the inverse of the plant U-model. It should be noted that 1) this U-control provides a universal control system design platform for many existing linear/nonlinear and polynomial/state space models, and 2) it complements many existing design approaches. Simulation studies are used as examples to demonstrate the analytically developed formulations and guideline for potential applications.

KEYWORDS U-model, U-control framework, model-independent control system design, dynamic inversion, simulation demonstrations

1 INTRODUCTION

In general, there are three frameworks for control system design. The two popular frameworks are 1) the model-based approach and 2) the model-free/data-driven approach. The third is a relatively new and that is 3) the model-independent/U-model-based approach. Here, is a brief introduction to the three frameworks.

1.1 MODEL-BASED CONTROL SYSTEM DESIGN

To show this framework, consider the general cascade feedback control system showing in Figure 1.1, consisting of the following elements:

G_p : Plant, which could be modelled as a linear transfer function or a nonlinear dynamic equation in either the polynomial or state space expression
G_c : Classic controller
G : Closed loop performance function, specified in advance by designers and/or users

For a linear plant G_p , the controller G_c could be designed by means of

$$G_c = G_p^{-1} \frac{G}{1-G} \qquad (1.1)$$

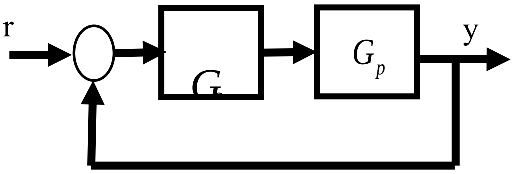
For a nonlinear plant G_p , the controller could be designed as follows:

$$G_c = f(G_p, G) \qquad (1.2)$$

where $f(*)$ is a function that links the plant and closed loop performance to determine the control through a certain type of inversion.

Here, are some remarks on the control-design framework.

- The model of the plant G_p is requested in advance, which the model sets include the linear/nonlinear polynomial and state space expressions.
- Advantages: there are many mature approaches available for this design framework [1-3]. It has been the predominant approach in academic research and industrial applications.
- Disadvantage 1: the framework features unnecessary repetition in design. Taking a linear plant model as an example, it unnecessarily repeats the calculation of $\frac{G}{1-G}$ if the plant model changed in (1.1).
- Disadvantage 2: it is difficult to design nonlinear plant-based control systems and difficult to specify the transient responses of nonlinear control systems with this framework.
- Disadvantage 3: the model structure affects the approach needed for the linear/nonlinear and polynomial/state space models, which is a common feature of model-based design frameworks.



1.2 MODEL-FREE/DATA-DRIVEN CONTROL SYSTEM DESIGN

There are various approaches to model-free control system design. A few well-known designs are described below.

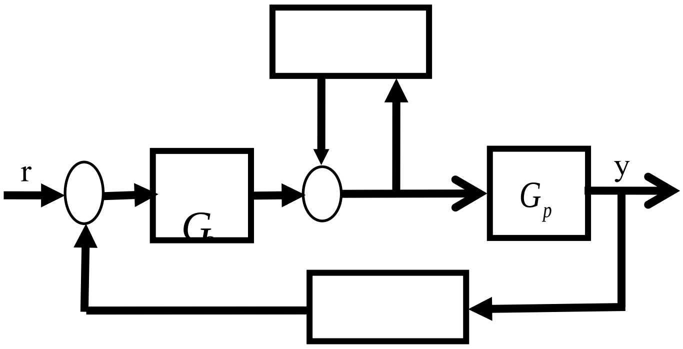
1) PID control by the Ziegler-Nichols approach [4]

This heuristic method of tuning a PID controller G_c (see Figure 1.1) has the following features:

- No need for a model of the plant G_p , even when mild conditions are required for the controlled plants.

- Advantage: it is the most commonly and easily used trial and error approach.
- Disadvantages: this approach wastes experimental work to obtain plant models. Almost all engineering plants/processes and input/output measurements are possible to model in principle, although it is sometimes a difficult task.

2) Iterative learning control (ILC) [5]
This framework (see Figure 1.2) has the following features:
- No need for a model of the plant G_p in design, even when mild conditions are required for the controlled plant.
- Requires iterative learning to improve the controller G_c with repeated reference stimulation; we finally achieve $G_c G_p = G_p^{-1} G_p = 1$.
- Advantages: this approach considers every possibility for integrating past control information into the next round of control design. No need for a clear model structure.
- Disadvantage 1: this approach wastes experimental work to obtain plant models, which is an issue with almost every engineering process.
- Disadvantage 2: this approach is only available in a repeatable control environment under strict conditions
- Disadvantage 3: it is challenging to control nonlinear dynamic plants with this approach.



3) Model-free control (MFC) [6]
This framework, inspection of (see Figure 1) has the following features:

- No need for a model of the plant G_p , even under mild constraints (e.g., an ultra-local model $y^{(v)} = F + \alpha u$ where α is a coefficient and u is the controller output) on the controlled plants.
- This approach is an enhanced PID controller ($u = -\dfrac{F - y^{(v)} + k_p e + k_i \int e + k_d \dot{e}}{\alpha}$) in which *F* needs to be estimated each time.
- Advantages: the ultra-local model can be used to approximate complex dynamic plants and improve control performance in this approach.
- Disadvantages: those of this approach are similar to those of PID controllers.

1.3 MODEL-INDEPENDENT CONTROL SYSTEM DESIGN [7-11]
This framework (see Figure 1.3) consists of the following:

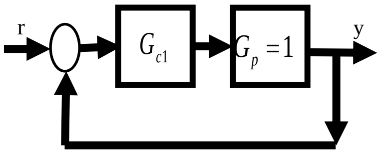
G_{c1} : Linear invariant controller
G_p^{-1} : Dynamic inversion of plant

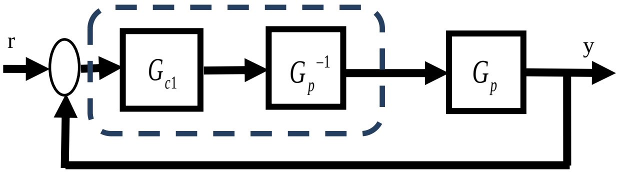
$$G_{c1} = \left. \frac{G}{1-G} \right|_{G_p=1} \quad (1.3)$$

$$G_c = G_{c1} G_p^{-1} \qquad (1.4)$$

Some remarks are given on the control framework.

- It features model-independent controller design.
- Advantage 1: the parallel design controller and dynamic inversion make the design procedure applicable to linear/nonlinear polynomial/state space model structures. Transient responses can be specified for nonlinear systems. It is neat in design without waste/repetition if the plant model changes.
- Advantage 2: this approach complements most existing design approaches.
- Disadvantages: this approach is sensitive to model uncertainty; robustness is the paramount issue in designing control systems.





2 Discrete time U-model set

The U-model expresses an explicit input-output relationship $U\left(y(t), u(t-1), \Psi(t)\right)$ at time t with time-varying parameters $\Psi(t)$ to absorb dynamics implicitly. This is a control-oriented model and derived from existing principle models or data-fitting models. This section explains 1) the definition of the U-model, the principles of converting classic models into U-models, 2) the dynamic inversion of U-polynomial models and 3) the dynamic inversion of U-state space models.

2.1 U-MODELS

Definition: For a single input ($u \in \mathrm{R}$ single) and single output ($y \in \mathrm{R}$) dynamic system $\sum$, Assign to it a triplet (U, Ψ, f_U) where $U = \{U_\alpha | \alpha \in u(t-1)\}$ is a vector of appropriate dimension and $\Psi = \{\Psi_\alpha | \alpha \notin u(t-1)\}$ is a dynamic absorbing vector of appropriate dimension that is associated with U . Accordingly, the system U-model $\sum_{\text{U-model}}$ is defined as a polynomial/rational system, where the polynomial/rational function $f_U = \{ f_U(\alpha) | \alpha \in \Psi, U\}$ is a mapping $f_U : u(t-1) \xrightarrow{\Psi \cup U} y(t) \in \mathrm{R}$) from the input space to the output space.

2.1.1 U-MODEL REALISATION FROM CLASSIC POLYNOMIALS

Consider a general classical SISO polynomial model in form of

$$\begin{aligned} y(t) &= f_P(\Phi(*),\ \Theta) \\ \Phi(*) &= \Phi(Y_{t-1},\ U_{t-1}) \end{aligned} \tag{2.1}$$

where $\begin{cases} y(t) \in \mathrm{R} \\ u(t-1) \in \mathrm{R} \end{cases}$ are the output/input respectively, at $t \in \mathrm{Z}^+$ and $\Phi(*) = [\phi_0(*) \ \ \cdots \ \ \phi_L(*)] \in \mathrm{R}^{L+1}$ where $Y_{t-1},\ U_{t-1}$ are expanded from the output and input, respectively, in the proper dimensions. Let $\phi_i(*) = \begin{cases} \phi_i\,(y(t-1)\ ,...,\ y(t-n) \\ u(t-1)\ ,...,\ u(t-n)) \end{cases}$ $,\forall i = 0 \cdots n$, where *n* is the plant dynamic order and $\Theta = [\theta_0 \ \ \cdots \ \ \theta_L] \in \mathrm{R}^{L+1}$ is the associated parametric vector. Let the function $f_P : u \to y$ be a polynomial mapping of the input space to the output space. The vector form of the expanded equation (2.1) is given as follows:

$$y(t) = \Phi^T \Theta = \sum_{l=0}^{L} \phi_l(*)\theta_l \tag{2.2}$$

where the basis $\phi_l(*)$ are the smooth functions in the space expanded from the past inputs/outputs, e.g., $y^3(t-2)u(t-1)$, $u^3(t-1)$, $y(t-1)y(t-5)$, and the associated coefficients θ_l are real constants.

In the other terms, this is a general expression of a non-linear autoregressive moving average with exogenous input model (NARMAX) [12].

To realise a U-model from this classical polynomial, set up an absorbing rule.

Absorbing rule: Let $\mu: \quad R^{L+1} \to R^{M+1}$ be a map from a polynomial f_P to its U-polynomial f_U and suppose that its inverse μ^{-1} exists; therefore it has:

$$f_P(P(*),\ \Theta) \xrightarrow{\mu} f_U(\Psi(*),\ U(u(t-1))) \qquad (2.3)$$

The mapping has some proper algebra properties as [8]

$$\begin{aligned}
&a)\ \forall(P(*),\ \Theta),(\Psi(*),\ U(u(t-1))),\\
&f_P(*) = f_U(*) \Rightarrow (P(*),\ \Theta) = (\Psi(*),\ U(u(t-1)))\\
&b)\ \forall(P(*),\ \Theta) \in f_P, \exists(\Psi(*),\ U(u(t-1))) \in f_U, \qquad (2.4)\\
&\mu((P(*),\ \Theta)) = (\Psi(*),\ U(u(t-1)))\\
&c)\ \mu^{-1} \circ \mu = I
\end{aligned}$$

Accordingly with reference to (2.4), the mapping is a) injective (one to one), b), surjective (onto), and bijective as both a) and b), c) invertible (I is an identity function). In system aspect, the map, except making the structure expression changed, does not change any characteristics of the both models, such as output response, stability, dynamics and statics.

The absorbing rule is a formation of $\Psi(*)$ from the polynomial f_P with reference to $u(t-1)$: first identify a control basis function $U(u(t-1))$ and then absorb all the other associated functions as a coefficient that varies with time.

Therefore, using the absorbing rule, realising the f_U mapped from polynomial f_P (2.2) gives the following:

$$y(t) = \Psi^T U = \sum_{j=0}^{M} \psi_j(t) U_j\left(u(t-1)\right) \qquad (2.5)$$

This function is expanded from the above nonlinear function f_P as a polynomial in terms of $u(t-1)$. M is the number of items associated with input $u(t-1)$ and the time varying parameter vector $\Psi(*) = [\psi_0(t) \quad \cdots \quad \psi_M(t)] \in \mathrm{R}^{M+1}$ is a function derived from absorbing the other regression terms and the coefficients.

Example: consider the polynomial model as shown below:

$$\begin{aligned} y(t) &= 0.2 * \sin(y(t-1)) + u(t-1)\exp(-y^2(t-1)) \\ &-0.8y(t-2)u(t-2)u^3(t-1) \end{aligned} \qquad (2.6)$$

Absorbing the terms associated with $u(t-1)$ into the vector $\Psi(*)$ gives the corresponding U-model realisation as follows:

$$y(t) = \psi_0(t) + \psi_1(t)U_1(u(t-1)) + \psi_2(t)U_2(u(t-1)) \qquad (2.7)$$

where

$\psi_0(t) = 0.2 * \sin(y(t-1))$, $\psi_1(t) = \exp(-y^2(t-1))$,

$\psi_2(t) = -0.8y(t-2)u(t-2)$, $U_1(u(t-1)) = u(t-1)$ $U_2(u(t-1)) = u^3(t-1)$.

2.1.2 U-MODEL REALISATION FROM CLASSIC RATIONAL MODELS

Rational model, also known as total nonlinear model [13], is a ratio of two polynomials as follows:

$$y(t) = f_r(\Phi(*),\ \Theta) = \frac{f_{pn}(\Phi_n(*),\ \Theta_n)}{f_{pd}(\Phi_d(*),\ \Theta_d)} \qquad (2.8)$$

Here, f_r is a rational function, the ratio of the f_{pn} /numerator polynomial and f_{pd} / denominator polynomial, which are maps of the input space into the output space. The other definitions follow from the polynomial model above. Note that this rational model is totally nonlinear in terms of parameter estimation and control input design [13].

Continuing with the U-polynomial model conversion, formulate the U-rational model expression as follows:

$$y(t)=\frac{\Psi_n^T U_n}{\Psi_d^T U_d}=\frac{\sum_{j=0}^{M_n}\psi_{jn}(t)U_{jn}\left(u(t-1)\right)}{\sum_{j=0}^{M_d}\psi_{jd}(t)U_{jd}\left(u(t-1)\right)} \qquad (2.9)$$

To obtain the model inversion for solving the roots, expand the model as the following:

$$y(t)\left(\sum_{j=0}^{M_d}\psi_{jd}(t)U_{jd}\left(u(t-1)\right)\right)=\sum_{j=0}^{M_n}\psi_{jn}(t)U_{jn}\left(u(t-1)\right)$$

(2.10)

Example: consider the rational model as follows:

$$y(t)=\frac{0.1y^3(t-2)+\sin(u(t-1))+0.5u^3(t-1)}{1+\cos^2(y(t-2))+u^2(t-1)} \qquad (2.11)$$

Absorbing the terms associated with $u(t-1)$ into the vectors $\Psi_n(*)$, $\Psi_d(*)$ gives the corresponding U-model realisation as follows:

$$f_{pn}(u(t-1))=\psi_{0n}(t)+\psi_{1n}(t)U_{1n}(u(t-1))+\psi_{2n}(t)U_{2n}(u(t-1))$$
$$f_{pd}(u(t-1))=\psi_{0d}(t)+\psi_{1d}(t)U_{1d}(u(t-1))$$

(2.12)

where

$$\psi_{0n}(t)=0.1y^3(t-2) \qquad \psi_{1n}(t)=1 \quad \psi_{2n}=0.5$$
$$\psi_{0d}(t)=1+\cos^2(y(t-2)) \quad \psi_{1d}(t)=1$$

and

$$U_{1n}(u(t-1))=\sin(u(t-1)) \quad U_{2n}(u(t-1))=u^3(t-1)$$
$$U_{1d}(u(t-1))=u^2(t-1)$$

2.1.3 U REALISATION FROM A CLASSICAL STATE SPACE MODE – MULTI-LAYER U MODEL

For a general SISO state-space system model, it has:

$$\begin{aligned} X(t+1) &= F(X(t),u(t)) \\ y(t) &= h(X(t)) \end{aligned} \qquad (2.13)$$

where $\begin{cases} X \in \mathrm{R}^n \\ u \in \mathrm{R} \\ y \in \mathrm{R} \end{cases}$ denotes the state, the control, and the output at time $t \in \mathrm{Z}^+$ respectively. $F \in \mathrm{R}^n$ is a smooth mapping to represent the input to the state output, and $h \in \mathrm{R}$ is a smooth mapping to drive the states to the outputs. In this study, assume that the system relative degree r equals to the system order n and has no unstable zero dynamics (i.e., the model reversible) and that the state X can be obtained through measurement or observation.

Convert state-space model (2.13) into a multi-layer U-model expression as follows:

$$\begin{cases} x_1(t+1) = \sum_{j=0}^{M_1} \psi_{1j}(t) U_{1j}(x_2(t)) \\ \vdots \\ x_{n-1}(t+1) = \sum_{j=0}^{M_{n-1}} \psi_{(n-1)j}(t) U_{(n-1)j}(x_n(t)) \\ x_n(t+1) = \sum_{j=0}^{M_n} \psi_{nj}(t) U_{nj}(u(t)) \\ y(t) = h(X(t)) \end{cases} \qquad (2.14)$$

For each line, M_j is the number of terms associated with the next line state variable $x_{j+1}(t)$ and $\psi_{ij}(t) = \left[\psi_{j0}(t) \quad \cdots \quad \psi_{jM_j}(t)\right] \in \mathrm{R}^{M_j+1}, \quad i \in 1 \cdots n$ are

time-varying parameter vector functions absorbing the other state variables. In the penultimate line, M_n consists of the terms associated with control $u(t)$ and the time-varying vectors $\left[\psi_{n0}(t) \quad \cdots \quad \psi_{nM_n}(t)\right] \in \mathrm{R}^{M_n+1}$ absorb all the states associated with the control vector $\left[U_{n0}(t) \quad \cdots \quad U_{nM_n}(t)\right] \in \mathrm{R}^{M_n+1}$. Therefore, each line of the state space equation is a U-polynomial model, consisting of a multi-layer U-model expression.

To illustrate the realisation, consider a nonlinear system represented in terms of state-space model:

$$\begin{cases} x_1(t+1) = x_2(t) + 0.1x_1(t)x_2(t) \\ x_2(t+1) = -0.1x_1(t) - 0.7x_2(t) + u(t) \\ y(t) = x_1(t) \end{cases} \qquad (2.15)$$

Take realisation of the corresponding multi-layer U-model by using the absorbing rule as below:

$$\begin{cases} x_1(t+1) = \psi_{11}(t)U_{11}(x_2(t)) \\ x_2(t+1) = \psi_{20}(t) + \psi_{21}U_{12}(u(t)) \\ y(t) = x_1(t) \end{cases} \tag{2.16}$$

where

$\psi_{11}(t) = 1 + 0.1x_1(t)$, $\psi_{20}(t) = -0.1x_1(t) - 0.7x_2(t)$, $\psi_{21}(t) = 1$, $U_{11}(x_2(t)) = x_2(t)$
$U_{12}(u(t)) = u(t)$

2.2 INVERSION OF U-POLYNOMIAL MODELS

For simplicity, consider the SISO polynomial U-model (3.4). Newton-Raphson algorithm [14] is a choice to determine the roots of U-models; that is, the roots are the candidates of controller output $u(t-1)$.

Iteratively, the root searching computation gives rise to the following formulation:

$$u_{k+1}(t-1) = u_k(t-1) - \left. \frac{y(t) - \sum_{j=0}^{M} \lambda_j(t) u_k^j(t-1)}{\dfrac{d\left[\sum_{j=0}^{M} \lambda_j(t) u_j(t-1)\right]}{du(t-1)}} \right|_{u_k(t-1)=u_k^j(t-1)}$$

(2.17)

Here, index k is the iteration handle: generate the (k+1)th results from the kth iteration, $k \succ 0$. There are also various root solving algorithms available [15]. In parallel, these algorithms are also applicable for U-rational model root solving based on (2.10).

It should be noted that in simulation studies, MATLAB codes, such as *roots*, can be used to find accurate roots of the U-model equations.

2.3 INVERSION OF U-STATE SPACE MODELS

For simplicity, consider the SISO U-state space model (2.14). Inversion is a multi-layer root solving procedure involving a backstepping routine whenever $x_1(t+1)$ is known; each line of the equation iteratively uses the Newton-Raphson algorithm to obtain $x_1(t+1)$ $x_2(t+1)$ $\cdots$ in backstepping order.

3 U-MODEL-BASED CONTROL SYSTEM DESIGN

A Chinese survey paper [16] has covered the major publications till 2012. Later, representative studies include “U-Block model technique” [8], “Control of total nonlinear systems” [9], “U-model enhanced Smith predict control for time delayed nonlinear processes” [11], and “U-neural networks enhanced control system design” [10].

This section further expands/formulates the U-control framework with updated results, including a newly introduced two parallel dynamic inversions in design, robust analysis, and a step by step procedure for U-control implementation.

3.1 U-CONTROL FRAMEWORK

Let G_p be a general dynamics in any expression of linear/nonlinear and polynomial/ state space models. Assumingly, the plant has the mostly claimed properties as those claimed in the other representative works [17]. Accordingly,

1) The model inverse G_p^{-1} exists
2) Lipschitz continuity satisfied, G_p and its inverse G_p^{-1} are diffeomorphic and globally uniformly Lipschitz in R^n ; that is,

   $$\begin{aligned}&\|G(x_1)-G(x_2)\| \le \gamma_1 G\|x_1-x_2\|, \quad \forall x_1, x_2 \in \mathrm{R}^n\\ &\|G^{-1}(x_1)-G^{-1}(x_2)\| \le \gamma_2 G^{-1}\|x_1-x_2\|, \quad \forall x_1, x_2 \in \mathrm{R}^n\end{aligned}$$ where x_1, x_2 are the states while G_p in expression of state space equation, γ_1, γ_2 are the Lipschitz coefficients.

For simplicity, but not losing generality, take consideration of a SISO (input $u \in \mathrm{R}^1$ and output $y \in \mathrm{R}^1$) U-model based control system, U-control system in short, which is constructed within an autonomous linear feedback control framework with a triplet bracketed of:

$$\sum = \begin{pmatrix} F_{lfbc} & G_{c1} & G_{ip} \end{pmatrix} \qquad (3.1)$$

where F_{lfbc} is a linear feedback loop with functions, linear virtual controller $G_{c1}: y \to u$ and virtual unit plant $G_{ip} = 1: u \to y$.

This U-control system structure proposes a model-independent control procedure, because the designs of $G_{c1}: y \to u$ and $G_{ip} = 1: u \to y$ are independent. These two independent designs are explained below.

For design of the virtual linear controller $G_{c1}: y \to u$, referring to Figure 3.1, it gives

$$G_{c1} = \frac{G}{1-G} = (1-G)^{-1} G \qquad (3.2)$$

where G is a specified closed loop transfer function with proper dynamic/static responses.

For design of the virtual unit plant $G_{ip} = 1 : u \to y$, designing/formulating the plant inverse G_P^{-1} gives

$$G_{ip} = G_P^{-1} G_p = 1 : u \to y \qquad (3.3)$$

Remark 1: Regarding to the merit of the design prototype, the established U-control system framework (3.1) has two independent inversion designs, 1) linear controller $G_{c1} : y \to u$ without involving any plant model structures, therefore it is also named as linear invariant controller [9], 2) virtual plant unitisation $G_{ip} = G_P^{-1} G_p = 1 : u \to y$ applicable to almost all smooth dynamics models (NB, hard nonlinear dynamic models could be sorted out along similar route in the subsequent studies). Therefore the two designs are separately independent and connected within a linear feedback control loop.

Remark 2: Regarding to the efficiency of the U-control system design, linear controller $G_{c1} : y \to u$ is once off design irrespective to plant model types and parameters. Plant inverter G_P^{-1} is formable for polynomial and state space equations in U-model and numerically solvable for the roots to achieve $G_{ip} = G_P^{-1} G_p = 1 : u \to y$. Consequently, the is reduced to the determination of the plant inverse G_P^{-1} once the linear controller designed. Consequently, the design procedure is that once off $G_{c1} : y \to u$ design and $G_P^{-1} \Rightarrow G_{ip} = G_P^{-1} G_p = 1 : u \to y$ follow up design to keep the same closed loop performance while plant model is changed.

Remark 3: Regarding to the inversion involved in control system design. This is a must for any type of control system design. U-control provides concise structure and less computational effort for its two inversions (one is the inversion of specified linear closed loop transfer function and the other is the inversion of plant U-model). This aspect can be explained through an inverse function Ψ^{-1}, for U-control systems, it is a split into two separate functions of $G_{c1} = \Psi_1^{-1}(G)$ (linear dynamic inversion) and $G_p^{-1} = \Psi_2^{-1}(G_p)$ (U-model based root solving). For the other popular control system design approaches, it is at least a function of $\Psi^{-1}(G_{c1}, G_p)$, which is a common formulation in classical linear feedback control system design. It should be notes that it is more complex in designing control systems with nonlinear plant models.

Remark 4: In regarding to the relationship in control system design between the U-control and the other major approaches, U-control is a supplement to the approaches and taking away the need for the plant structures in controller design and clearly specifies the closed loop dynamic/static performances. It should be noted that taking the transient performance into consideration of designing nonlinear control systems has received significant consideration, and analysing their performance through linear system approaches is a key research domain [18]. U-control is therefore a promising procedure.

In some sense, those, using the other approaches, well designed control systems could take U-control as a plug-in box to expand to control different types of plants.

Remark 5: As U-control is fundamentally based on the assumption $G_P^{-1}G_p = 1$, it is critical to consider the robustness of the resulting control system in the case of uncertainty, which is very common in practical systems. Surely two types of approaches are the candidates by adding additional robust control loop and/or adaptive loop.

3.2 DESIGN PROCEDURE

With reference to the aforementioned description and the block diagram in Figure 3.1. Here is a step by step design procedure listed.

1) Establish a stable linear feedback control system structured in Figure 3.1, Assign G for the whole system transfer function in the closed loop setup. Specify G by means of damping ratio, undamped natural frequency, and steady state error and/or the other performance indices (such as poles and zeros, and frequency response).
2) Let the plant model be a constant unit or the virtual pant $G_{ip} = 1: u \to y$ have been achieved. To determine a linear invariant controller G_{c1} by taking inverse of the closed loop transfer function G using (3.2). Accordingly, the desired system output is equivalently determined by the output v of the controller G_{c1} .
3) Convert plant model into U-model realisation $\sum_{\text{U-model}}$ with reference to the formulations presented in Section 2.
4) To achieve $G_{ip} = 1: u \to y$ to guarantee the desired output $y_{desired}(t) = v(t)$, determine the controller output by solving an equation $v(t) - \sum_{\text{U-model}} = 0$, that is, $u(t) \in v(t) - \sum_{\text{U-model}} = 0$
5) Locate/connect the blocks in Figure 3.2.

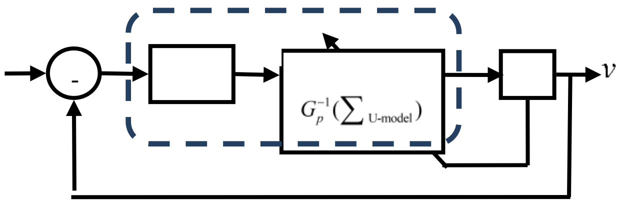


Figure 3.1 U-control framework

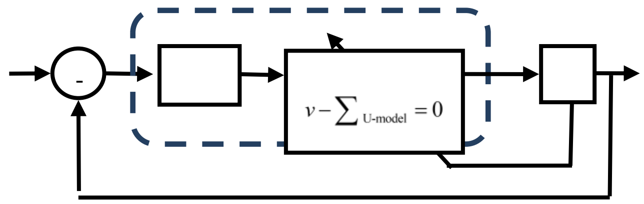


Figure 3.2 U-control implementation

3.3 U-MODEL BASED ADAPTIVE CONTROL

This was first studied in recent publications [9, 19]. Figure 3.3 shows a double looped (feedback control and adaptation) diagram that adds an adaptation role in dealing with uncertainties and disturbance by online updating model parameters. Interested readers can find the details in the aforementioned reference. Compared with the classic adaptive control scheme, Adaptive U-control does not request controller design in each updating step; it only updates the plant model while the controller is fixed. Here only the framework is explained briefly and the detailed expansions will be reported in the future publications.

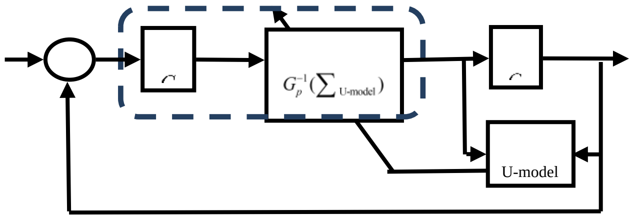


Figure 3.3. U-model based adaptive control

3.4 ROBUSTNESS ANALYSIS OF U-CONTROL

This section presents the robustness analysis of U-control based on discrete-time H_∞ using linear matrix inequalities (LMI) technique. Consider the state space equation in terms of multi-layer U-realisation (2.14) with an external disturbance vector $W(t)=\left[w_1(t) \quad \cdots \quad w_n(t)\right]^T$ as

$$
\begin{cases}
x_1(t+1) = \sum_{j=0}^{M_1} \psi_{1j}(t) U_{1j}(\mathrm{x}_2(t)) + w_1(t) \\
\vdots \\
x_{n-1}(t+1) = \sum_{j=0}^{M_{n-1}} \psi_{(n-1)j}(t) U_{(n-1)j}(\mathrm{x}_n(t)) + w_{n-1}(t) \\
x_n(t+1) = \sum_{j=0}^{M_n} \psi_{nj}(t) U_{nj}(u(t)) + w_n(t) \\
y(t) = h(\mathrm{x}(t))
\end{cases}
\tag{3.4}
$$

Remark 1: Assume that the elements of the external disturbance vector are bounded, i.e., $|w_i(t)| < d_i, \forall i = 1 \cdots n$, where d_i is a positive constant.

To provide the robustness analysis, take one single line state of $x_n(t+1)$ from state-space equation (3.4) at first, and then extend the analysis to the other state variables $x_i(t+1)$. Accordingly, take out

$$
T_{wx}: \begin{cases}
x_n(t+1) = \sum_{j=0}^{M_n} \left\{ \psi_{nj}(t) U_{nj}(\mathrm{x}_n(t)) + w_{nj}(t) \right\} \\
y(t) = h(x(t)).
\end{cases}
\tag{3.5}
$$

The control objective is to minimize the effect of the external disturbance w_n on the state vector x_n . This study takes the discrete-time H_∞ robust control technique into consideration, which the robust control condition is

$$
\|T_{wx}\| < \beta \;\Rightarrow\; \sup \frac{\|x_n\|_{L_2}}{\|w_n\|_{L_2}} < \beta
\tag{3.6A}
$$

where β is a known constant defining the upper boundary of H_∞ performance index. Eq. (3.6A) can be rewritten as

$$
\|x_n\|_{L_2}^2 < \beta^2 \|w_n\|_{L_2}^2
\tag{3.6B}
$$

or equivalently

$$
\beta^{-1} \|x_n\|_{L_2}^2 - \beta \|w_n\|_{L_2}^2 < 0
\tag{3.6C}
$$

with

$$
\begin{cases}
\|x_n\|_{L_2}^2 = \sum_{t=0}^{\infty} x_n^T(t) x_n(t) \\
\|w_n\|_{L_2}^2 = \sum_{t=0}^{\infty} w_n^T(t) w_n(t)
\end{cases}
\tag{3.7}
$$

From the above formulations, it gives

$$
\Rightarrow \sum_{t=0}^{\infty} \left\{ \beta^{-1} x_n^T(t) x_n(t) - \beta w_n^T(t) w_n(t) \right\} < 0
\tag{3.8}
$$

Construct the positive-definite Lyapunov function with

$$V(x_n(t)) = x_n^T(t)Qx_n(t) > 0 \tag{3.9}$$

where $Q > 0$. Suppose the gradient of the Lyapunov function ($\nabla V(x_n(t))$) being satisfied in the following inequality,

$$\nabla V(x_n(t)) + \beta^{-1}x_n^T(t)x_n(t) - \beta w_n^T(t)w_n(t) < 0 \tag{3.10}$$

In order to prove condition of (3.10), take summation ($\sum$) of all terms as

$$\sum_{t=0}^{\infty}\nabla V(x_n(t)) + \sum_{t=0}^{\infty}\left\{\beta^{-1}\,\mathrm{x}_n^T(t)x_n(t) - \beta w_n^T(t)w_n(t)\right\} < 0 \tag{3.11}$$

Since the first term of (3.11) is positive, then the second term is always negative, that is,

$$\sum_{t=0}^{\infty}\left\{\beta^{-1}x_n^T(t)x_n(t) - \beta w_n^T(t)w_n(t)\right\} < 0 \tag{3.12}$$

which is the same as condition (3.8)). Then inequality (3.10) is a correct assumption.

Determine the gradient of the Lyapunov function by

$$\nabla V(x_n(t)) = x_n^T(t+1)Qx_n(t+1) - x_n^T(t)Qx_n^T(t) \tag{3.13}$$

By substituting (3.13) into (3.10), it gives

$$x_n^T(t+1)Qx_n(t+1) - x_n^T(t)Qx_n(t) + \beta^{-1}x_n^T(t)x_n(t) - \beta w_n^T(t)w_n(t) < 0 \tag{3.14}$$

Now, substituting $x_n(t+1)$ from (3.4) into (3.14), it gives

$$\begin{aligned}
&\sum_{j=0}^{M_n}\left\{\left(\psi_{nj}(t)U_{nj}(x_n(t)) + w_{nj}(t)\right)^T Q\left(\psi_{nj}(t)U_{nj}(x_n(t)) + w_{nj}(t)\right)\right\} \\
&-\sum_{j=0}^{M_n}\left\{\begin{array}{l}\left(\psi_{nj}(t-1)U_{nj}(x_n(t-1)) + w_{nj}(t-1)\right)^T Q \\ \left(\psi_{nj}(t-1)U_{nj}(x_n(t-1)) + w_{nj}(t-1)\right)\end{array}\right\} \\
&+\sum_{j=0}^{M_n}\left\{\begin{array}{l}\beta^{-1}\left(\psi_{nj}(t-1)U_{nj}(x_n(t-1)) + w_{nj}(t-1)\right)^T \\ \left(\psi_{nj}(t-1)U_{nj}(x_n(t-1)) + w_{nj}(t-1)\right)\end{array}\right\} \\
&-\sum_{j=0}^{M_n}\left\{\beta\left(w_{nj}^T(t)w_{nj}(t)\right)\right\} < 0
\end{aligned} \tag{3.15}$$

In what follows, for simplicity, shorten the following notations as

$$\begin{cases} \psi_{nj}(t) \triangleq \psi_t \qquad \psi_{nj}(t-1) \triangleq \psi_{t-1} \qquad U_{nj}(x_n(t)) \triangleq U_t \\ U_{nj}(x_n(t-1)) \triangleq U_{t-1} \qquad w_{nj}(\mathrm{t}) \triangleq w_t \qquad w_{nj}(\mathrm{t}-1) \triangleq w_{t-1}. \end{cases}$$

Then (3.15) is expressed as

$$\begin{aligned} &\sum \begin{Bmatrix} U_t{}^T \psi_t{}^T Q \psi_t U_t + U_t{}^T \psi_t{}^T Q w_t \\ + w_t{}^T \mathrm{Q} \psi_t U_t + w_t{}^T \mathrm{Q w}_t \end{Bmatrix} \\ &- \sum \begin{Bmatrix} U_{t-1}{}^T \psi_{t-1}{}^T Q \psi_{t-1} U_{t-1} + U_{t-1}{}^T \psi_{t-1}{}^T Q w_{t-1} \\ + w_{t-1}{}^T Q \psi_{t-1} U_{t-1} + w_{t-1}{}^T \mathrm{Q w}_{t-1} \end{Bmatrix} \\ &+ \sum \beta^{-1} \begin{Bmatrix} U_{t-1}{}^T \psi_{t-1}{}^T \psi_{t-1} U_{t-1} + U_{t-1}{}^T \psi_{t-1}{}^T w_{t-1} \\ + w_{t-1}{}^T \psi_{t-1} U_{t-1} + w_{t-1}{}^T w_{t-1} \end{Bmatrix} \\ &- \sum \left\{ \beta (w_t{}^T w_t) \right\} < 0 \end{aligned} \tag{3.16}$$

Further, it can be expressed in terms of quadratic form,

$$\begin{aligned} &\sum \left(U_t{}^T \psi_t{}^T \{Q\} \psi_t U_t \right) + \sum \left(U_t{}^T \psi_t{}^T \{Q\} w_t \right) + \sum \left(w_t{}^T \{\mathrm{Q}\} \psi_t U_t \right) \\ &+ \sum \left(w_t{}^T \{\mathrm{Q} - \beta I\} \mathrm{w}_t \right) + \sum \left(U_{t-1}{}^T \psi_{t-1}{}^T \left\{ \beta^{-1} I - Q \right\} \psi_{t-1} U_{t-1} \right) \\ &+ \sum \left(U_{t-1}{}^T \psi_{t-1}{}^T \left\{ \beta^{-1} I - Q \right\} w_{t-1} \right) + \sum \left(w_{t-1}{}^T \left\{ \beta^{-1} \mathrm{I} - Q \right\} \psi_{t-1} U_{t-1} \right) \\ &- \sum \left(w_{t-1}{}^T \left\{ \beta^{-1} I - Q \right\} w_{t-1} \right) < 0 \end{aligned} \tag{3.17}$$

and then matrix form of

$$\begin{bmatrix} \sum U_t \psi_t \\ \sum w_t \\ \sum U_{t-1} \psi_{t-1} \\ \sum w_{t-1} \end{bmatrix}^T H \begin{bmatrix} \sum U_t \psi_t \\ \sum w_t \\ \sum U_{t-1} \psi_{t-1} \\ \sum w_{t-1} \end{bmatrix} < 0 \tag{3.18}$$

with

$$H = \begin{bmatrix} Q & Q & 0 & 0 \\ Q & Q - \beta I & 0 & 0 \\ 0 & 0 & \beta^{-1} I - Q & \beta^{-1} I - Q \\ 0 & 0 & \beta^{-1} I - Q & \beta^{-1} I - Q \end{bmatrix} < 0 \tag{3.19}$$

Now, applying the Schur complement [20] on (3.19), it gives

$$\begin{cases} F_{22} < 0 \\ F_{11} - F_{12} \left(F_{22} \right)^{-1} F_{21} < 0 \end{cases} \tag{3.20}$$

where

$$F_{11}=\begin{bmatrix} Q & Q \\ Q & Q-\beta I \end{bmatrix},\ F_{12}=F_{21}=\begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix},$$
$$F_{22}=\begin{bmatrix} \beta^{-1}I-Q & \beta^{-1}I-Q \\ \beta^{-1}I-Q & \beta^{-1}I-Q \end{bmatrix}$$

Then condition of (3.20) can be simplified as:

$$\begin{cases} \begin{bmatrix} \beta^{-1}I-Q & \beta^{-1}I-Q \\ \beta^{-1}I-Q & \beta^{-1}I-Q \end{bmatrix} < 0 \\ \begin{bmatrix} Q & Q \\ Q & Q-\beta I \end{bmatrix} < 0 \end{cases} \quad (3.21)$$

Defining a new variable $\gamma := \beta^{-1}$ in the first inequality of (3.21), it changes to

$$LMI(Q,\gamma):\ \begin{bmatrix} \gamma I-Q & \gamma I-Q \\ \gamma I-Q & \gamma I-Q \end{bmatrix} < 0\ \Rightarrow\ \gamma I-Q<0 \quad (3.22)$$

where the optimal values Q^* and γ^* can be calculated via Matlab LMI toolbox. The optimal value of β is correspondingly given by $\beta^*=(\gamma^*)^{-1}$. Applying Schur complement on the second inequality of (3.21), it gives

$$\begin{cases} Q<0 \\ Q-\beta I-Q(Q)^{-1}Q<0\ \rightarrow\ -\beta I<0 \end{cases} \quad (3.23)$$

which yields $-\gamma I<0$. Then, from (3.23), it gives

$$LMI(Q,\gamma):\ \begin{bmatrix} Q & 0 \\ 0 & -\gamma I \end{bmatrix} < 0\ \Rightarrow\ Q^*,\gamma^* \quad (3.24)$$

where the existence of optimal solutions for Q^* and γ^* means the robustness of the U-model system versus external disturbances.

The robustness analysis for the remainder of equations of state-space model (3.4) can be proved similar to the above-presented procedure.

4 SIMULATION EXAMPLES

This simulation demonstration selected three plant models: SISO Hammerstein model, SISO nonlinear state space model, and an extended total nonlinear model. In the control system design, it formulated a commonly used pole placement controller for the three examples. The main purposes for designing the simulation tests of the U-control are as follows:

1) To demonstrate the principle of model-independent design in U-control.
2) To demonstrate a once-off design of the linear controller in accordance of a closed loop performance specification irrespective to the plant model structure to change or different models.

3) To demonstrate the workability and conciseness/simplicity of U-control, particularly in the design of nonlinear control systems.
4) To demonstrate that U-control can supplement/enhance classic pole placement control.

From previous sections, the design is divided into two parallel blocks: 1) designing the linear invariant control G_{c1} (thus $v(t)$) by reversing the specified closed loop transfer function and 2) determining the control input $u(t-1)$ by reversing the plant U-model equation.

For familiarisation of different notations used in U-control, this simulation section takes in $y_{t-j} = y(t-j), j \in Z^{\geq}$, $u_{t-j} = u(t-j), j \in Z^{+}$, $v_{t-j} = v(t-j), j \in Z^{+}$, $x_j(t+1) = x_{j(t+1)}, j \in Z^{+}$, and $\psi_j(t) = \psi_j, j \in Z^{\geq}$

Design invariant control G_{c1}

In a popular approch, the conventional pole placement control [21] assigns the closed loop characteristic equation in terms of Z transform:

$$\begin{aligned} A(z) &= z^2 + a_1 z + a_2 \\ &= z^2 - 1.3205z + 0.4966 \end{aligned} \tag{4.1}$$

Equivalently the poles are located at 0.6603 ± i0.2463 within the unit circle (stable), a typical decayed oscillatory response with damping ratio of 0.7 and unit undamped natural frequency, this is a commonly used dynamic response index set.

Assign the numerator polynomial in the desired closed loop transfer function as

$$B(z) = bz \tag{4.2}$$

where the constant b is determined by steady state error requirement to a given reference input. Accordingly in this case study, to make the steady state follow a given step reference input without error, it sets up:

$$b = A(z)\big|_{z=1} = 1 - 1.3205 + 0.4966 = 0.1761 \tag{4.3}$$

Thereby the resultant transfer function is specified as

$$\frac{Y(z)}{R(z)} = G(z) = \frac{0.1761z^{-1}}{1 - 1.3205z^{-1} + 0.4966z^{-2}} \tag{4.4}$$

It should be noted, while the condition $G_{ip} = 1 : u \to y$ satisfied, it gives

$$\frac{V(z)}{R(z)} = G(z) = \frac{0.1761z^{-1}}{1 - 1.3205z^{-1} + 0.4966z^{-2}} \tag{4.5}$$

where $V(z)$ is the Z transform of the controller G_{c1} output as shown in Figure 3.1.

To determine the linear invariant controller G_{c1}, temporarily, let the plant $G_p = 1$ or $G_{ip} = G_p^{-1} G_p = 1 : u \to y$. Then take inverse of the transfer function G to yield:

$$\begin{aligned} G_{c1} &= \frac{G(z)}{1 - G(z)} = \frac{bz^{-1}}{1 + (a_1 - c)z^{-1} + a_2 z^{-2}} \\ &= \frac{0.1761z^{-1}}{1 - 1.4966z^{-1} + 0.4966z^{-2}} \end{aligned} \tag{4.6}$$

The rest of the control system design will formulate the specific plant inverse G_p^{-1} in form of U-model for each selected example, which will be implemented in each related sub-section.

4.1 HAMMERSTEIN MODEL: A SISO NONLINEAR POLYNOMIAL [7]

The Hammerstein style model, a static (memoryless) nonlinear block is cascaded with a linear differential equation (dynamic), is a good representative of various nonlinear dynamic plants/processes. Its control has been widely studied with model-based approaches [22]. The simulation example selected [7] is as follows:

$$\begin{aligned} y_t &= 0.5y_{t-1} + x_{t-1} + 0.1x_{t-2} \\ x_t &= 1 + u_{t-1} - u_{t-1}^2 + 0.2u_{t-1}^3 \end{aligned} \quad (4.7)$$

where $\{y_t, u_t, x_t\}$ are the plant output, input, and intermediate variable for the static nonlinear component output respectively.

As explained above, the first step in U-control system design is generic to determine the linear invariant controller G_{c1}, that is, independent of the plant model and universally designed (as was done in the beginning of this section). The second step of the design is specifically working out the controller output u_{t-1} by inverting the plant model to find its U-model roots. Accordingly, to realise a U-model for the controller output, it uses the absorbing rule to convert the Hammerstein model into the following U-expression:

$$y_t = \psi_0 + \psi_1 u_{t-1} + \psi_2 u_{t-1}^2 + \psi_3 u_{t-1}^3 \quad (4.8)$$

where

$$\begin{aligned} \psi_0 &= 0.5y_{t-1} + 1 + 0.3x_{t-2} \qquad \psi_1 = 1 \\ \psi_2 &= -1 \qquad\qquad \psi_3 = 0.2 \end{aligned} \quad (4.9)$$

Then replace the output y_t with the virtual controller output v_t (that is, the desired output). Subsequently, it determines one of the roots by solving (4.8) as the controller output. This gives the following formula:

$$u_{t-1} \in roots\left(v_t - \psi_0 + \psi_1 u_{t-1} + \psi_2 u_{t-1}^2 + \psi_3 u_{t-1}^3 = 0\right) \quad (4.10)$$

Figure 4.1 illustrates the simulation results.

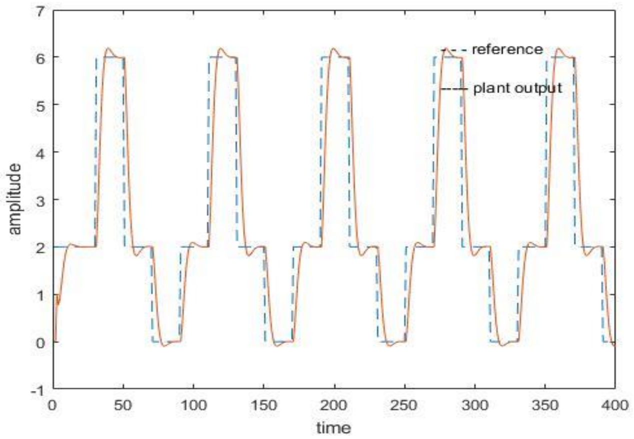


Figure 4.1(a) Plant output

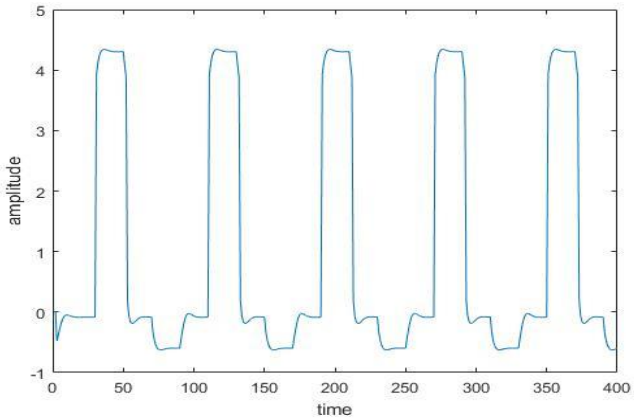


Figure 4.1(b) Control input

4.2 NONLINEAR STATE SPACE MODEL

The control of strict-feedback nonlinear systems is a widely studied, challenging topic [23]. Many leading publications have used neural network model-based approaches to approximate the model set as a pointwise linear model set to alternatively design equivalent linear control systems [24]. The simulation example for the state-space model is as follows:

$$\begin{cases} x_{1t} = x_{2(t-1)} + 0.1x_{1(t-1)}x_{2(t-1)} \\ x_{2t} = -0.1x_{1(t-1)} - 0.7x_{2(t-1)} + u_{t-1} \\ y_t = x_{1t} \end{cases} \quad (4.11)$$

where $\{y_t, u_t, x_t\}$ denote the plant output, input, and $x(t)$ is a state vector respectively. This represents a second order nonlinear dynamic plant.

Again, in the second step of U-control system design, it requires to work out the specific controller output u_{t-1} by inverting the plant U-model. The realised multi-layer U-model is expressed as follows:

$$\begin{aligned} x_{1t} &= \psi_{11} x_{2(t-1)} \\ x_{2t} &= \psi_{20} + \psi_{21} u_{t-1} \\ y_t &= x_{1t} \end{aligned} \qquad (4.12)$$

where $\psi_{11} = 1 + 0.1 x_{1t}$, $\psi_{20} = -0.1 x_{1(t-1)} - 0.7 x_{2(t-1)}$, and $\psi_{21} = 1$.

This is a two layer U-model structure. Accordingly, using backstepping routine with initial $x_{1t} = v_t$ works out the controller output u_{t-1} by inverting each line of the equations, as specified in step 4 in U-control design procedure. The simulation results are shown in Figure 4.2.

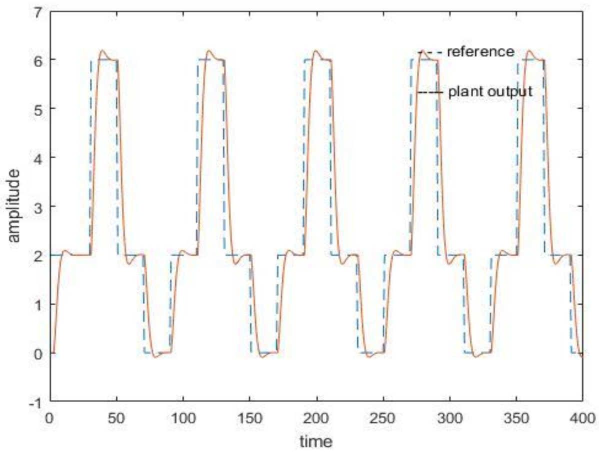


Figure 4.2(a) Plant output

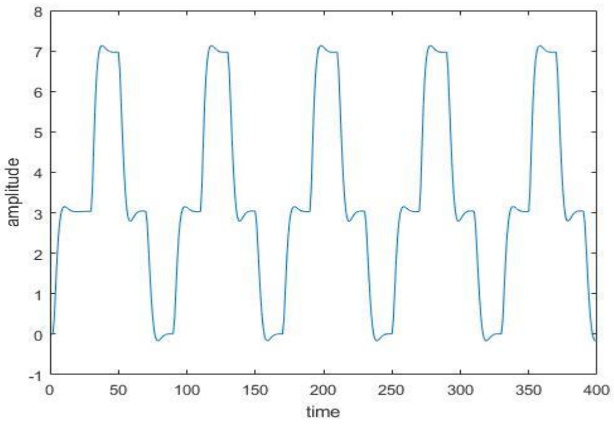


Figure 4.2(b) Control input

4.3 EXTENDED TOTAL NONLINEAR MODEL [9]

The control of nonlinear rational systems, which are modelled as ratios of two nonlinear polynomials, is even more challenging. Until a recent analytical U-model-based approach [9], these models were previously taken as examples of complex systems in neuro-control system design. The difficulty is that rational model sets are subject to total nonlinearity (both in the parameters/identification and in input/control) [13]. The selected simulation example [9] with dynamics (time delay) and transcendental nonlinearities was as follows:

$$y(t)=\frac{0.5y(t-1)+\sin(u(t-1))+u(t-1)}{1+\exp(-y^2(t-1))} \qquad (4.13)$$

where $\{y_t, u_t\}$ are the plant output and input respectively. Once again by applying the absorbing rule, it wields the following U-rational model:

$$\begin{aligned} & y_t\left(1+\exp(-y_{t-1}^2)\right)=0.5y_{t-1}+\sin(u_{t-1})+u_{t-1} \Rightarrow \\ & \psi_{0d}\,y(t)=\psi_{0n}+\psi_{1n}\sin(u_{t-1})+\psi_{2n}u_{t-1} \end{aligned} \qquad (4.14)$$

With the same linear invariant controller G_{c1} used as before, replacing the output y_t of (4.13) with the desired output v_t of (4.6) gives the following:

$$\psi_{0d} v_t = \psi_{0n} + \psi_{1n} \sin(u_{t-1}) + \psi_{2n} u_{t-1} \quad (4.15)$$

Subsequently, the control input u_{t-1} is obtained by the following:

$$\psi_{0d} v_t - \psi_{0n} - \psi_{1n} \sin(u_{t-1}) - \psi_{2n} u_{t-1} = 0 \quad (4.16)$$

Figure 4.3 illustrates the simulation results. Again, the bench test confirms the performance of the U-control.

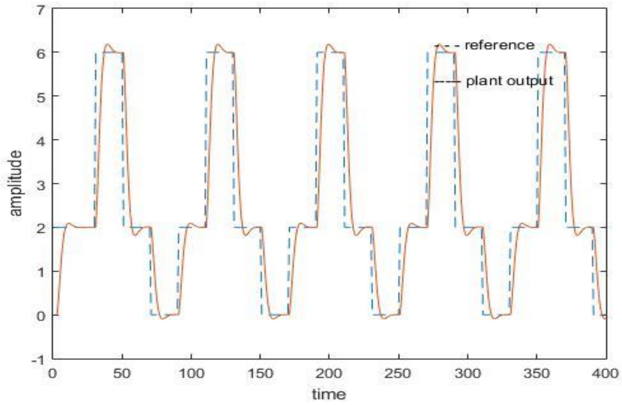


Figure 4.3(a) Plant output

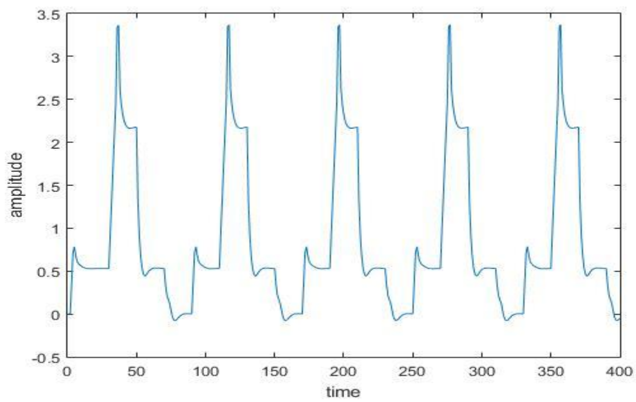


Figure 4.3(b) Control input

V. CONCLUSION

U-control has been featured in several publications. This tutorial has been presented to summarise and expand on the essential insights, formulations, and simulated case studies. We hope that this self-contained study can achieve the following purposes:

1) Explain/demonstrate the principle of model-independent design in U-control.
2) Explain/demonstrate a universal design for multiple plant model structures.
3) Explain/demonstrate U-control workability and effectiveness/efficiency, particularly dealing with nonlinear plant control.
4) Explain/demonstrate U-control as a supplement to classic control system design frameworks.

In terms of research techniques, compared with the two most popular control system design framework, model based and model free, this model-independent design effectively relieves the complexity involved in inverting the controller and plant together. The problem of inversion is reduced to inverting the plant model only, which means this framework results in an invariant controller that is universally applicable to the classic model sets and features no repetition if the plant model changes. The most critical issue with this design framework is its robustness because it relies on having $G_p^{-1}G_p = 1$. Accordingly, robust U-control is a central topic for research and applications. Additional demonstrations of its use in real cases will help to prepare it for wider application.

In research methodology, U-control is simple/concise and uses basic tools such as poles and zeros for analysing/designing linear system stability, transient responses (damping ratios and undamped natural frequencies), and the small gain theorem for robustness analysis. All of these are fundamental in postgraduate courses. However, U-control effectively combines them to provide solutions for challenging research problems. It hopes this technique will be user friendly for industrial engineers working with ad hoc applications and easy-to-use for academics developing further enhancements of the method.

REFERENCES

1. Kim K, Kim M, Kim D, Lee D. Modeling and velocity-field control of autonomous excavator with main control valve. Automatica. 2019;104:67-81.
2. Mobayen S, Ma J. Robust finite-time composite nonlinear feedback control for synchronization of uncertain chaotic systems with nonlinearity and time-delay. Chaos, Solitons & Fractals. 2018;114:46-54.
3. Oliveira TR, Rodrigues VHP, Fridman L. Generalized Model Reference Adaptive Control by means of Global HOSM Differentiators. IEEE Transactions on Automatic Control. 2019;64(5):2053-60.
4. Ogata K, Yang Y. Modern control engineering: Prentice-Hall; 2002.
5. Xu J-X, Tan Y. Linear and nonlinear iterative learning control: Springer; 2003.
6. Fliess M, Join C. Model-free control. International Journal of Control. 2013;86(12):2228-52.
7. Zhu Q, Guo L. A pole placement controller for non-linear dynamic plants. Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering. 2002;216(6):467-76.
8. Zhu Q, Zhao D, Zhang J. A general U-block model-based design procedure for nonlinear polynomial control systems. International Journal of Systems Science. 2016;47(14):3465-75.
9. Zhu Q, Liu L, Zhang W, Li S. Control of complex nonlinear dynamic rational systems. Complexity. 2018;2018(Article ID 8953035):12 pages.
10. Zhu Q, Zhang W, Zhang J, Sun B. U-neural network-enhanced control of nonlinear dynamic systems. Neurocomputing. 2019;352:12-21.
11. Geng X, Zhu Q, Liu T, Na J. U-model based predictive control for nonlinear processes with input delay. Journal of Process Control. 2019;75:156-70.
12. Billings SA. Nonlinear system identification: NARMAX methods in the time, frequency, and spatio-temporal domains: John Wiley & Sons; 2013.
13. Zhu Q, Wang Y, Zhao D, Li S, Billings SA. Review of rational (total) nonlinear dynamic system modelling, identification, and control. International journal of systems science. 2015;46(12):2122-33.
14. Gerald CF. Applied numerical analysis: Pearson Education India; 2004.
15. Chong EK, Zak SH. An introduction to optimization: John Wiley & Sons; 2013.
16. Xu F-X, Zhu Q-M, Zhao D, Li S-Y. U-model based design methods for nonlinear control systems a survey of the development in the 1st decade. Control and Decision. 2013;28(7):961-71.
17. Isidori A. Nonlinear control systems: Springer Science & Business Media; 2013.
18. Chen BM, Lee TH, Peng K, Venkataramanan V. Composite nonlinear feedback control for linear systems with input saturation: Theory and an application. IEEE Transactions on automatic control. 2003;48(3):427-39.
19. Du W, Wu X, Zhu Q. Direct design of a U-model-based generalized predictive controller for a class of non-linear (polynomial) dynamic plants. Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering. 2012;226(1):27-42.
20. Boyd S, El Ghaoui L, Feron E, Balakrishnan V. Linear matrix inequalities in system and control theory: Siam; 1994.

21. Åström KJ, Wittenmark B. Adaptive control (2nd ed): Addison Wesley, Reading; 1995.
22. Fruzzetti K, Palazoğlu A, McDonald K. Nolinear model predictive control using Hammerstein models. Journal of process control. 1997;7(1):31-41.
23. Lv M, Wang Y, Baldi S, Liu Z, Wang Z. A DSC method for strict-feedback nonlinear systems with possibly unbounded control gain functions. Neurocomputing. 2018;275:1383-92.
24. Ge SS, Yang C, Dai S-L, Jiao Z, Lee TH. Robust adaptive control of a class of nonlinear strict-feedback discrete-time systems with exact output tracking. Automatica. 2009;45(11):2537-45.

U-model-based Model Reference Adaptive Control of Nonlinear Systems

Ji Qiu [1], Quanmin Zhu, Mokhtar Nibouche, Yufeng Yao

Department of Engineering Design and Mathematics, University of the West of England, Bristol, United Kingdom. BS16 1QY

Abstract: This paper proposes a design method of model reference adaptive control for uncertain nonlinear dynamic systems using a U-model-based framework. The design is taken in two separate routines, i.e., classic model reference adaptive control routine and nonlinear plant model solver by U-model-based framework. Model reference adaptive control is designed based on the MIT normalised algorithms and Lyapunov rules are employed to select a specified reference model. The controller output is achieved by deriving the plant inverse in terms of the roots of U-model equations. Different from traditional adaptive methods, the proposed controller simplifies the model reference adaptive control structure and the virtual controller parameter estimation is set up via U-model framework. The simulation results are conducted to investigate the efficiency of the proposed method. From these simulations, it can be verified that the system response of the proposed control scheme for nonlinear dynamic plant model obtains the same as the classic/standard model reference adaptive control design for linear dynamic plant model.

[1] Corresponding Author, Address: Coldharbour Ln, Stoke Gifford, Bristol BS16 1QY, E-mail address: Ji2.qiu@live.uwe.ac.uk, url: https://scholar.google.com/citations?user=4UweLEcAAAAJ&hl=en,Tel.:+44 7784143762.

Keywords: Model reference adaptive control, U-model framework, U-control, virtual plant model, adaptive control.

1 Introduction

1.1.Background and motivation

In the past few decades, nonlinear dynamic systems controller design becomes focal point of control process in modern industry. The efficiency and quality of the controllers directly affect the profits, which contributes to the requirement of maximally reduction on controllers' complexity. However, the adaptive controller system, which aims to modify the behaviour in response to the variations in the dynamics of the process, is already a complex nonlinear system [1]. Therefore, designing a concise and efficient adaptive controller system for nonlinear dynamic model was motivated as a fervent challenge.

The adaptive controller designs for nonlinear dynamic systems have been proposed and discussed in different adaptive control theory. Zhao proposed smooth adaptive internal model control based on model to simplify the identification of time-varying parameters in presence of bounded external disturbances [2]. Liu applied sliding-mode observer for MIMO uncertain neutral stochastic systems [3] and some researchers applied sliding mode model reference adaptive control (MRAC) to deal with nonlinear dynamic systems [4–7]. MIT based MRAC and modified adaptive control (MAC) are considered for pressure regulation of hypersonic wind tunnel in [8].

MRAC is widely used in digital adaptive control dynamic systems design with online parameter estimation and adjustment. It can be applied to a nonlinear aircraft model with unknown structural damage [9] and a quadrotor UAV [10]. Nonlinear Hydraulic Actuator is designed by adaptive PID and MRAC switch controller in [11]. Besides, MRAC also be applied for nonlinear switched systems under asynchronous switching between subsystems and adaptive controller [12] and human-robot interaction [13]. Meanwhile, the method of neural network combined with MRAC is considered for solving the nonlinear system [14,15]. Fuzzy logic controller based MRAC is introduced in [16]. Direct and indirect MRAC is proposed in [17] for multivariable piecewise affine systems. Modified MRAC is considered for inverted pendulum compared with MRAC [18].

MRAC is inherently nonlinear so the structure could be completely concise if it is analysed and designed through linear technique or plant. Up to now, the MRAC algorithms for nonlinear dynamic systems obtain numerous data identification or adjustment calculations, which cause strict requirement of chips resources and time.

Enlightened by [3, 19–22], U-model can be deployed to reduce the complexity of MRAC applying to nonlinear dynamic plant model. U-model is a plant oriental structure apply for including but not limited to most smooth nonlinear dynamic systems. With the root solving algorithms, nonlinear dynamic plant model can be directly applied to MRAC and the reference model could be designed as a linear model.

To address this issue, this paper will propose a U-model root solver for the actual nonlinear dynamic plant model, and a virtual plant model to substitute plant model in MRAC. The virtual plant model is designed as same as the plant model in the classic/standard MRAC controller system, and the virtual parameter (gain) could be time-varying estimated comparing to a reference model. The actual nonlinear dynamic plant model is overlooked by applying the root solving algorithms of U-model.

1.2.Contribution

- U-model based model reference adaptive control is investigated as a supplement/alternative approach in application of adaptive control for nonlinear systems.
- Scope of application is checked directly for the smooth nonlinear dynamic models, i.e. the Hammerstein model.
- The proposed scheme combined with MIT normalized algorithms and Lyapunov model reference adaptive control are investigated.

1.3.Paper organization

The rest of the paper is organised as follow: in Section 2, U-model structure, MIT normalised algorithms and Lyapunov rules are presented. Moreover, the whole controller design of U-model-based MRAC is also proposed in Section 2. The simulation and experimental results are demonstrated in Section 3 and conclusions are given in Section 4.

2. Problem statement and preliminaries

In this paper, we consider the SISO smooth nonlinear systems described by a form of NARMAX as follows [22]:

$$y(t) = f\left[y(t-1),\ldots,y(t-n),u(t-1),\ldots,u(t-n),\ldots,e(t),\ldots,e(t-n)\right] \tag{8}$$

where it is a discrete-time polynomial model and $f[*]$ is a smooth nonlinear function. $y(t)$ and $u(t)$ are systems output and input. $e(t)$ denotes disturbance, noise, uncertain dynamics, etc.

Therefore, the U-model stochastics characteristics are defined as:

$$\begin{cases} U(t) = \sum_{j}^{M} \alpha_j u^j(t-1) + e(t) \\ \alpha_j(t) = \alpha_j\left(\Xi, y(t-i), u(t-i-1), e(t-i)\right) \\ y(t) = U(t), \quad i = 1, 2, ..., n \end{cases} \tag{9}$$

where $U(t)$ denotes a pseudo variable, M is the degree of the control input $u(t-1)$, $\alpha_j(t)$ denotes time-varying parameter including some model's parameter constants Ξ, the past time input $u(t-2),...,u(t-i-1)$, the past time output $y(t-1),...,y(t-i)$ and error signal $e(t-1),...,e(t-i)$. For example, with such U-model framework, a NARMAX model

$$\begin{aligned} y(t) = 1.1 + 0.6y^2(t-1) + 0.3u(t-1)e(t-1) - 0.4y(t-1)u^2(t-1) + \\ 0.6y(t-1)u^3(t-1)e(t-2) + e(t-1) + e(t) \end{aligned} \tag{10}$$

can be treated as a pure power series of input $u(t-1)$:

$$\begin{aligned} &U(t) = \alpha_0(t) + \alpha_1(t)u(t-1) + \alpha_2(t)u^2(t-1) + \alpha_3(t)u^3(t-1) + e(t) \\ &y(t) = U(t) \end{aligned} \tag{11}$$

where

$$\begin{aligned} \alpha_0(t) &= 1.1 + 0.6y^2(t-1) + e(t-1) \\ \alpha_1(t) &= 0.3e(t-1) \\ \alpha_2(t) &= 0.4y(t-1) \\ \alpha_3(t) &= 0.6y(t-1)e(t-2) \end{aligned} \tag{12}$$

To design the controller, the control input should be obtained by root-solving algorithms, such as:

$$u_{k+1}(t-1) = u_k(t-1) - \frac{\Phi[U_k(t-1)] - U(t)}{d\Phi[u(t-1)]/du(t-1)}$$

$$= u_k(t-1) - \frac{\sum_{j=0}^{M} \alpha_j(t) u_k^j(t-1) - U(t)}{d\left[\sum_{j=0}^{M} \alpha_j(t) u^j(t-1)\right] / du(t-1)} \tag{13}$$

For linear plant models, the control input is

$$u(t-1) = \frac{U(t) - \alpha_0(t)}{\alpha_1(t)} \tag{14}$$

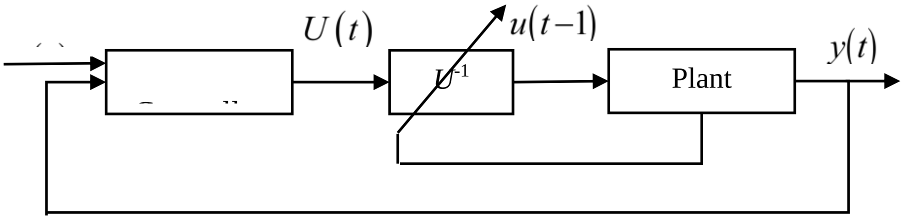


Figure 1 *Block diagram of U-model framework control system*

Figure shows the whole U-model based nonlinear/linear control system [20]. U^{-1} is a root solver such as Newton-Raphson algorithm for obtaining the roots, real controller output $u(t-1)$. $w(t)$ is the desired reference input and $U(t)$ is the pseudo controller output.

3. U-model-based MRAC

The actual nonlinear plant model G is applied root solver U^{-1}. In order to apply the linear direct MRAC controller design, a virtual plant model G_v is constructed as the combination of the root solver U^{-1} and the actual nonlinear plant model G is considered to be a constant 1 mathematically. The whole structure is shown as follow:

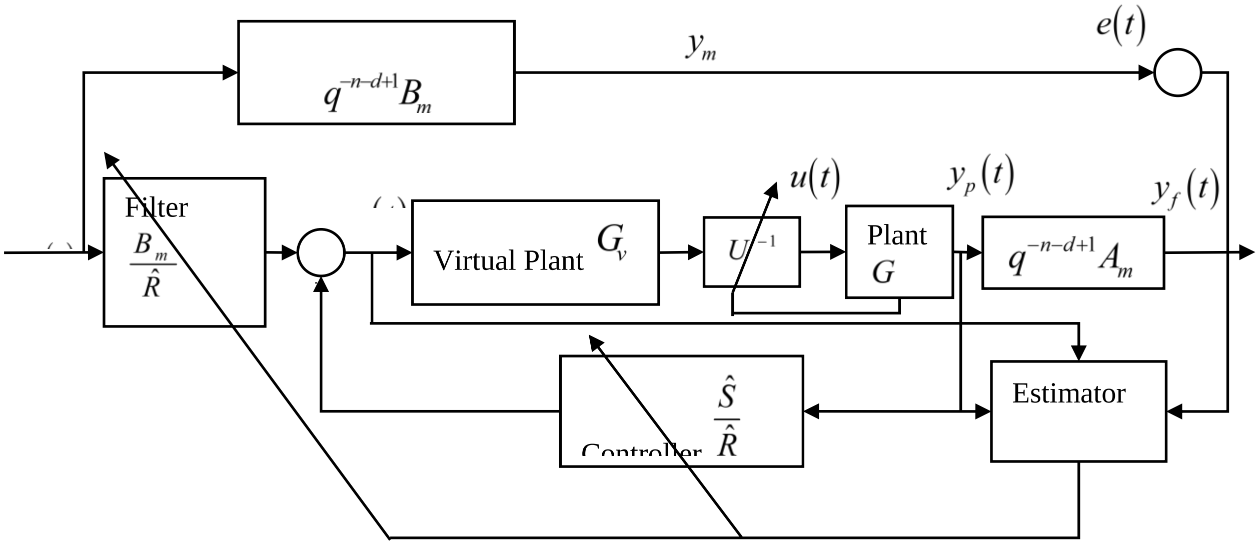


Figure 2 *U-model based MRAC control systems*

As shown in Figure , the virtual plant model G_v is $\frac{B}{A}$, which mathematically same as plant model in **Error! Reference source not found.**. The actual plant model in Figure 4 can be smooth nonlinear dynamic system.

This is the online estimation and control law for a class of smooth nonlinear dynamic systems with U-model based MRAC structure.

Apply pole placement to figure out discrete time plant model. Let the plant model written in terms of the forward shift operator q as

$$\boldsymbol{A}(\boldsymbol{q})y_p(t)=\boldsymbol{B}(\boldsymbol{q})u(t) \tag{15}$$

where y_p and u are output and input of the plant model. $\boldsymbol{A}$ and $\boldsymbol{B}$ are polynomials of degree n and m. They are coprime and $\boldsymbol{B}$ is minimum phase. Defined them as

$$\boldsymbol{A}(\boldsymbol{q})=\boldsymbol{q}^n+a_1\boldsymbol{q}^{n-1}+...+a_n \tag{16}$$

and

$$\boldsymbol{B}(\boldsymbol{q}) = \boldsymbol{q}^m + b_1\boldsymbol{q}^{m-1} + \ldots + b_m \tag{17}$$

where $m \prec n$ and $b_0 \neq 0$.

A general linear controller is described as

$$\boldsymbol{R}(\boldsymbol{q})u(t) = \boldsymbol{T}(\boldsymbol{q})r(t) - \boldsymbol{S}(\boldsymbol{q})y_p(t) \tag{18}$$

To determine the controller,

$$u(t) = \frac{\boldsymbol{T}(\boldsymbol{q})}{\boldsymbol{R}(\boldsymbol{q})}r(t) - \frac{\boldsymbol{S}(\boldsymbol{q})}{\boldsymbol{R}(\boldsymbol{q})}y_p(t) \tag{19}$$

where $r(t)$ is the reference input signal. Then the plant output could be written as

$$y_p(t) = \frac{\boldsymbol{B}(\boldsymbol{q})\boldsymbol{T}(\boldsymbol{q})}{\boldsymbol{A}(\boldsymbol{q})\boldsymbol{R}(\boldsymbol{q}) + \boldsymbol{B}(\boldsymbol{q})\boldsymbol{S}(\boldsymbol{q})}r(t) \tag{20}$$

The output and the input of the reference model can be described by

$$y_m(t) = \frac{\boldsymbol{B_m}(\boldsymbol{q})}{\boldsymbol{A_m}(\boldsymbol{q})}r(t) \tag{21}$$

where A_m is monic and stable, $\deg \boldsymbol{A_m}(\boldsymbol{q}) - \deg \boldsymbol{B_m}(\boldsymbol{q}) = d$, $d > 0$.

To force the output of plant model the same as the output of reference model,

$$\frac{\boldsymbol{B}(\boldsymbol{q})\boldsymbol{T}(\boldsymbol{q})}{\boldsymbol{A}(\boldsymbol{q})\boldsymbol{R}(\boldsymbol{q}) + \boldsymbol{B}(\boldsymbol{q})\boldsymbol{S}(\boldsymbol{q})} = \frac{\boldsymbol{B_m}(\boldsymbol{q})}{\boldsymbol{A_m}(\boldsymbol{q})} \tag{22}$$

which can be easily found as

$$\frac{\boldsymbol{T}(\boldsymbol{q})}{\boldsymbol{A}(\boldsymbol{q})\boldsymbol{R}(\boldsymbol{q})+\boldsymbol{B}(\boldsymbol{q})\boldsymbol{S}(\boldsymbol{q})}=\frac{\boldsymbol{B_m}(\boldsymbol{q})}{\boldsymbol{A_m}(\boldsymbol{q})\boldsymbol{B}(\boldsymbol{q})} \tag{23}$$

The roots of the close loop characteristic polynomial $\boldsymbol{A_m}(\boldsymbol{q})\boldsymbol{B}(\boldsymbol{q})$ are stable as the roots of $\boldsymbol{A_m}(\boldsymbol{q})$ and the roots of $\boldsymbol{B}(\boldsymbol{q})$ are stable. Define the close loop characteristic polynomials as

$$\boldsymbol{P}(\boldsymbol{q}) = \boldsymbol{A_m}(\boldsymbol{q})\boldsymbol{B}(\boldsymbol{q}) = b_0\boldsymbol{q}^{(n_m+m)} + p_1\boldsymbol{q}^{(n_m+m)-1} + \ldots + p_{n_m+m} \tag{24}$$

where $n_m = \deg \boldsymbol{A_m}(\boldsymbol{q})$.

To meet the requirements of (4.2.10),

$$\boldsymbol{T}(\boldsymbol{q}) = \boldsymbol{B_m}(\boldsymbol{q}) \tag{25}$$

$$\boldsymbol{A}(\boldsymbol{q})\boldsymbol{R}(\boldsymbol{q})+\boldsymbol{B}(\boldsymbol{q})\boldsymbol{S}(\boldsymbol{q}) = \boldsymbol{A_m}(\boldsymbol{q})\boldsymbol{B}(\boldsymbol{q}) \tag{26}$$

Define $n_{\boldsymbol{R}} = \deg \boldsymbol{R}(\boldsymbol{q})$, $n_{\boldsymbol{S}} = \deg \boldsymbol{S}(\boldsymbol{q})$, $n_e = n + n_{\boldsymbol{R}} + 1$, and $n_u = n_{\boldsymbol{R}} + n_{\boldsymbol{S}} + 2$. (4.2.13) can be written as

$$M\begin{bmatrix} \mathscr{C}(\boldsymbol{R}) \\ \mathscr{C}(\boldsymbol{S}) \end{bmatrix} = \mathscr{C}(\boldsymbol{P}) \tag{27}$$

where $M \in R^{n_e \times n_u}$ is the Sylvester matrix

$$M = \begin{bmatrix} 1 & 0 & & 0_{(n-1)\times 1} & 0_{d\times 1} & 0_{(d+1)\times 1} & & 0_{(n+d-1)\times 1} \\ a_1 & 1 & & 1 & b_0 & b_0 & & b_0 \\ a_2 & a_1 & & a_1 & b_1 & b_1 & & b_1 \\ \ldots & \ldots & & a_2 & \ldots & \ldots & \ldots & b_2 \\ a_n & a_n & \ldots & \ldots & b_m & b_m & & \ldots \\ 0_{(n-1)\times 1} & 0_{(n-2)\times 1} & & a_n & 0_{(n-1)\times 1} & 0_{(n-2)\times 1} & & b_m \end{bmatrix} \tag{28}$$

The coefficients of $\boldsymbol{R}(\boldsymbol{q})$, $\boldsymbol{S}(\boldsymbol{q})$ and $\boldsymbol{P}(\boldsymbol{q})$ are vectors

$$\mathscr{C}(\boldsymbol{R})=\begin{bmatrix} r_0 \\ r_1 \\ \dots \\ r_{n_{\boldsymbol{R}}} \end{bmatrix},\mathscr{C}(\boldsymbol{S})=\begin{bmatrix} s_0 \\ s_1 \\ \dots \\ s_{n_S} \end{bmatrix},\mathscr{C}(\boldsymbol{P})=\begin{bmatrix} b_0 \\ p_1 \\ \dots \\ p_{n_{\boldsymbol{m}}+m} \end{bmatrix} \tag{29}$$

As proved in [23], if $n_{\boldsymbol{R}} \geq n_S$, the control law (4.2.6) is causal. Assume that $\deg \boldsymbol{S} = n-1$ and $\deg \boldsymbol{A_m} = 2n-m-1$, so $\deg \boldsymbol{P} = 2n-1$. Assumed that $\deg \boldsymbol{R} = n-1$ and $M \in R^{2n\times 2n}$ to gain a minimum degree causal controller. Then

$$\begin{aligned} \boldsymbol{R} &= r_0\boldsymbol{q}^{n-1} + r_1\boldsymbol{q}^{n-2} + \dots + r_{n-1},\ \ r_0 \neq 0 \\ \boldsymbol{S} &= s_0\boldsymbol{q}^{n-1} + s_1\boldsymbol{q}^{n-2} + \dots + s_{n-1},\ \ s_0 \neq 0 \end{aligned} \tag{30}$$

From (4.2.13) and (4.2.17), we can know that $r_0 = b_0$.

The filtered output signal can be defined for a linear estimation model

$$y_f(t) = \boldsymbol{q}^{-n-d+1}\boldsymbol{A_m}y(t) \tag{31}$$

From (4.2.2) it becomes

$$y_f(t) = \frac{\boldsymbol{q}^{-n-d+1}\boldsymbol{A_m}\boldsymbol{B}}{\boldsymbol{A}}u(t) \tag{32}$$

To match the condition (4.2.13), $y_f(t)$ should be

$$\begin{aligned} y_f(t+d) &= \frac{\boldsymbol{q}^{-n+1}(\boldsymbol{AR}+\boldsymbol{BS})}{\boldsymbol{A}}u(t) \\ &= \left(\boldsymbol{R}+\boldsymbol{S}\frac{\boldsymbol{B}}{\boldsymbol{A}}\right)u(t-n+1) \\ &= \boldsymbol{R}u(t-n+1)+\boldsymbol{S}y(t-n+1) \end{aligned} \tag{33}$$

Define the parameter vector θ and the regressed matrix $\varphi(t)$ as

$$\theta = \begin{bmatrix} r_1 \\ \cdots \\ r_{n-1} \\ s_0 \\ \cdots \\ s_{n-1} \end{bmatrix} \in R^{2n-1} \tag{34}$$

and

$$\varphi(t) = \begin{bmatrix} u(t-1) \\ \cdots \\ u(t-n+1) \\ y(t) \\ \cdots \\ y(t-n+1) \end{bmatrix} \in R^{2n-1} \tag{35}$$

When $r_0 = b_0$, (4.2.20) can be obtained as the linear identification model

$$y_f(t+d) = b_0 u(t) + \varphi^T(t)\theta \tag{36}$$

From (4.2.20) and (4.2.23), the model matching control law (4.2.6) can be gained as

$$u(t) = -\frac{1}{b_0}\left[\varphi^T(t)\theta - \boldsymbol{q}^{-n+1}\boldsymbol{B_m}u_c(t)\right] \tag{37}$$

With the filtered plant model (4.2.23) and the matching control law (4.2.24), direct adaptive control can be apply with discrete time plant model.

However, when the plant model is unknown, the controller parameters $\boldsymbol{R}$ and $\boldsymbol{S}$ cannot be obtained. So the controller parameters $\boldsymbol{R}$ and $\boldsymbol{S}$ should be estimates by polynomials $\hat{\boldsymbol{R}}(t)$ and $\hat{\boldsymbol{S}}(t)$ in q, which is

$$u(t)=\frac{\boldsymbol{B_m}}{\hat{\boldsymbol{R}}(t)}r(t)-\frac{\hat{\boldsymbol{S}}(t)}{\hat{\boldsymbol{R}}(t)}y(t) \tag{38}$$

The corresponding close loop system is

$$y(t)=\frac{\boldsymbol{BB_m}}{\boldsymbol{A}\hat{\boldsymbol{R}}(t)+\boldsymbol{B}\hat{\boldsymbol{S}}(t)}r(t) \tag{39}$$

Define the parameter error

$$\bar{\theta}(t)=\hat{\theta}(t)-\theta \tag{40}$$

where $\hat{\theta}(t)$ is the estimate of q at time t. The filtered output error signal is then defined as

$$e_f(t)=y_f(t)-\boldsymbol{q}^{-n-d+1}\boldsymbol{B_m}r(t) \tag{41}$$

By the parameter error, the filter output can be written as

$$\begin{aligned} y_f(t+d)&=\boldsymbol{q}^{-n+1}\frac{\boldsymbol{A_m BB_m}}{\boldsymbol{A}\hat{\boldsymbol{R}}(t)+\boldsymbol{B}\hat{\boldsymbol{S}}(t)}r(t)=\boldsymbol{q}^{-n+1}\frac{\boldsymbol{AR}+\boldsymbol{BS}}{\boldsymbol{A}\hat{\boldsymbol{R}}(t)+\boldsymbol{B}\hat{\boldsymbol{S}}(t)}\boldsymbol{B_m}r(t)\\ &=\boldsymbol{q}^{-n+1}\frac{\boldsymbol{R}+\frac{\boldsymbol{BS}}{\boldsymbol{A}}}{\hat{\boldsymbol{R}}(t)+\frac{\boldsymbol{B}\hat{\boldsymbol{S}}(t)}{\boldsymbol{A}}}\boldsymbol{B_m}r(t)=\boldsymbol{q}^{-n+1}\frac{\boldsymbol{R}u(t)+\boldsymbol{S}y(t)}{\hat{\boldsymbol{R}}(t)u(t)+\hat{\boldsymbol{S}}(t)y(t)}\boldsymbol{B_m}r(t)\\ &=\frac{b_0u(t)+\varphi^T(t)\theta}{\boldsymbol{q}^{n-1}\left[b_0u(t)+\varphi^T(t)\hat{\theta}(t)\right]}\boldsymbol{B_m}r(t) \end{aligned} \tag{42}$$

Compared (4.2.23) and (4.2.29),

$$b_0u(t)+\varphi^T(t)\hat{\theta}(t)=\boldsymbol{q}^{-n+1}\boldsymbol{B_m}r(t) \tag{43}$$

Therefore, the filter output error signal is than

$$
\begin{aligned}
e_f(t+d) &= y_f(t+d) - \boldsymbol{q}^{-n+1}\boldsymbol{B_m} r(t) \\
&= b_0 u(t) + \varphi^T(t)\theta - b_0 u(t) - \varphi^T(t)\hat{\theta}(t) \\
&= -\varphi^T(t)\bar{\theta}(t)
\end{aligned}
\tag{44}
$$

The model matching error dynamics can be expressed in the n th order fraction form as Figure [24]:

$$
\begin{aligned}
\boldsymbol{A}\xi(t-n) &= u(t) \\
y(t) &= \boldsymbol{B}\xi(t-n)
\end{aligned}
\tag{45}
$$

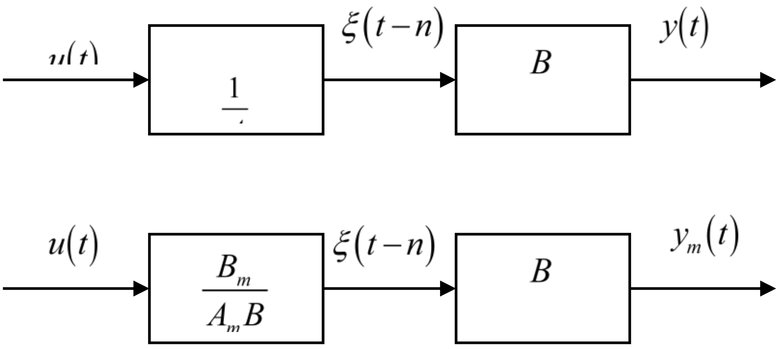


Figure 3 Fraction forms of the plant and the reference model

Then, the filtered output can be

$$
y_f(t+d) = \boldsymbol{q}^{-n+1}\boldsymbol{A_m} y(t) = \boldsymbol{q}^{-n+1}\boldsymbol{A_m B}\xi(t-n) = \boldsymbol{q}^{-2n+1}\boldsymbol{P}\xi(t) \tag{46}
$$

Similarly, the reference model can be expressed in the $2n-1$th order non-minimal fraction form as

$$
\begin{aligned}
\boldsymbol{A_m B}\xi_m(t-n) &= \boldsymbol{B_m} r(t) \\
y_m(t) &= \boldsymbol{B}\xi_m(t-n)
\end{aligned}
\tag{47}
$$

So that it can be expressed as

$$
\boldsymbol{q}^{-n+1}\boldsymbol{B_m} r(t) = \boldsymbol{q}^{-n+1}\boldsymbol{A}_m\boldsymbol{B}\xi_{\boldsymbol{m}}(t-n) = \boldsymbol{q}^{-n+1}\boldsymbol{P}\xi_{\boldsymbol{m}}(t) \tag{48}
$$

Define

$$\xi_e(t) = \xi(t) - \xi_m(t) \tag{49}$$

From (4.2.33) and (4.2.35), the d -step ahead filtered output error can be expressed by

$$\begin{aligned} e_f(t+d) &= y_f(t+d) - \boldsymbol{q}^{-n+1}\boldsymbol{B_m} r(t) \\ &= \boldsymbol{q}^{-2n+1}\boldsymbol{P}\left[\xi(t) - \xi_m(t)\right] \\ &= \boldsymbol{q}^{-2n+1}\boldsymbol{P}\xi_e(t) \end{aligned} \tag{50}$$

Define the plant, reference model, and model matching error states as

$$\begin{aligned} x(t) &= \begin{bmatrix} \xi(t-1) \\ \cdots \\ \xi(t-2n+1) \end{bmatrix} \in R^{2n-1} \\ x_m(t) &= \begin{bmatrix} \xi_m(t-1) \\ \cdots \\ \xi_m(t-2n+1) \end{bmatrix} \in R^{2n-1} \\ x_e(t) &= x(t) - x_m(t) = \begin{bmatrix} \xi_e(t-1) \\ \cdots \\ \xi_e(t-2n+1) \end{bmatrix} \in R^{2n-1} \end{aligned} \tag{51}$$

Moreover,

$$\begin{aligned} \xi_e(t) &= \boldsymbol{q}\xi_e(t-1) \\ &= -\boldsymbol{q}\left[\frac{1}{b_0}\boldsymbol{q}^{-2n+1}\boldsymbol{P} - 1\right]\xi_e(t-1) + \frac{1}{b_0}\boldsymbol{q}^{-2n+1}\boldsymbol{P}\xi_e(t) \\ &= -\boldsymbol{q}\left[\frac{1}{b_0}\boldsymbol{q}^{-2n+1}\boldsymbol{P} - 1\right]\xi_e(t-1) + \frac{1}{b_0}e_f(t+d) \end{aligned} \tag{52}$$

the state equation of x_e in the controllable canonical form is now then

$$x_e(t+1) = Ax_e(t) + \frac{1}{b_0}Be_f(t+d), \quad k \ge 0 \tag{53}$$

where

$$A = \begin{bmatrix} -p_0/b_0 & \cdots & -p_{2n-1}/b_0 \\ & & 0 \\ I_{(2n-2)\times(2n-2)} & & \cdots \\ & & 0 \end{bmatrix}, \ B = \begin{bmatrix} 1 \\ 0 \\ \cdots \\ 0 \end{bmatrix} \tag{54}$$

The model matching dynamics becomes

$$x_e(t+1) = Ax_e(t) - \frac{1}{b_0} B\varphi^T(t)\bar{\theta}(t), \ t \geq 0 \tag{55}$$

2.2.1 Normalised Algorithms

Normalised algorithms is used with MIT rule to develop the control law by less sensitive of the changes in the amplitude of the reference input [25].

Let the discrete-time linear time-invariant plant model be as

$$\frac{y_p(t)}{u(t)} = \frac{k_p \boldsymbol{B_p}(\boldsymbol{q})}{\boldsymbol{A_p}(\boldsymbol{q})}, \quad t = 0,1,2,\ldots \tag{56}$$

where $u(t)$ and $y_p(t)$ denote the input and output of the plant model, and k_p is unknown constant gain.

The reference model is written as

$$\frac{y_m(t)}{r(t)} = W_m(\boldsymbol{q}) = \frac{1}{P_m(\boldsymbol{q})} \tag{57}$$

where r is the reference input and it is assumed to be uniformly bounded.

The input $u(t)$ is chosen as

$$u(t) = \boldsymbol{\theta}^{*T}\boldsymbol{\omega} \tag{58}$$

where

$$\begin{aligned}\boldsymbol{\theta}^{*T} &= \left[\boldsymbol{\theta}_1^{*T}, \boldsymbol{\theta}_2^{*T}, \theta_3^*, \theta_4^*\right], \\ \boldsymbol{\omega} &= \begin{bmatrix}\boldsymbol{\omega}_1^T \\ \boldsymbol{\omega}_2^T \\ y \\ r\end{bmatrix}\end{aligned} \tag{59}$$

and

$$\begin{aligned}\boldsymbol{\omega}_1 &= \frac{\boldsymbol{\alpha}(\boldsymbol{q})}{\Lambda(\boldsymbol{q})}u, \\ \boldsymbol{\omega}_2 &= \frac{\boldsymbol{\alpha}(\boldsymbol{q})}{\Lambda(\boldsymbol{q})}y_p\end{aligned} \tag{60}$$

where

$$\boldsymbol{\alpha}(\boldsymbol{q}) = \begin{bmatrix}\boldsymbol{q}^{n-2} \\ \cdots \\ \boldsymbol{q} \\ 1\end{bmatrix} \tag{61}$$

and $\Lambda(\boldsymbol{q})$ is arbitrary Hurwitz polynomial. The eigenvalue of $\Lambda(\boldsymbol{q})$ is in $|\boldsymbol{q}| \le \sqrt{\delta}, \delta \in (0,1]$.

Applying the matching equations

$$\begin{aligned}&\theta_4^* = k_p^{-1} \\ &\boldsymbol{\theta}_1^{*T}\boldsymbol{\alpha}(\boldsymbol{q})\boldsymbol{A_p}(\boldsymbol{q}) + \left(\boldsymbol{\theta}_2^{*T}\boldsymbol{\alpha}(\boldsymbol{q}) + \boldsymbol{\theta}_3^{*T}\Lambda(\boldsymbol{q})\right)k_p\boldsymbol{B_p}(\boldsymbol{q}) = \\ &\qquad\qquad \Lambda(\boldsymbol{q})\left(\boldsymbol{A_p}(\boldsymbol{q}) - \boldsymbol{\theta}_4^{*T}P_m(\boldsymbol{q})k_p\boldsymbol{B_p}(\boldsymbol{q})\right)\end{aligned} \tag{62}$$

we can find that $y_p = y_m$ is achived. Using (62), $r(t) = P_m(\boldsymbol{q}) y_m(t)$, and $\varepsilon = y_p - y_m$, the existence of $\boldsymbol{\theta}^*$ can be guaranteed as

$$\varepsilon = W_m(\boldsymbol{q}) \frac{1}{\theta_4^*}\left(u(t) - \boldsymbol{\theta}^{*T}\boldsymbol{\omega}\right) \tag{63}$$

Let the certain equivalence adaptive control law [25] as

$$u = \boldsymbol{\theta}^T \boldsymbol{\omega} \tag{64}$$

where

$$\boldsymbol{\theta} = \begin{bmatrix} \boldsymbol{\theta}_1^T \\ \boldsymbol{\theta}_2^T \\ \theta_3 \\ \theta_4 \end{bmatrix} \tag{65}$$

is the estimate of θ^*. As θ_4^* is constant, (63) can be rewritten as

$$\varpi = W_m(\boldsymbol{q}) u = \boldsymbol{\theta}^{*T} \boldsymbol{\phi}_p \tag{66}$$

where $\boldsymbol{\phi}_p = W_m(\boldsymbol{q}) \boldsymbol{\omega}_p$, $\boldsymbol{\omega}_p = \begin{bmatrix} \omega_1^T \\ \omega_2^T \\ y \\ W_m^{-1}(\boldsymbol{q}) y \end{bmatrix}$.

Let the estimate $\varpi = \boldsymbol{\theta}^T \boldsymbol{\phi}_p$, the normalised estimation error can be

$$e = \frac{\varpi - \hat{\varpi}}{m^2} = -\frac{\bar{\boldsymbol{\theta}}^T \boldsymbol{\phi}_p}{m^2}, \quad m = 1 + \boldsymbol{\phi}_p^T \boldsymbol{\phi}_p \tag{67}$$

where $\bar{\boldsymbol{\theta}} = \boldsymbol{\theta} - \boldsymbol{\theta}^*$. When $t = 0, 1, 2, \ldots$, the adaptive law becomes

$$\begin{aligned}
&\boldsymbol{\theta}(t+1)=\boldsymbol{\theta}^{p}(t+1)+\Delta(t+1),\ \boldsymbol{\theta}(0)=\boldsymbol{\theta}_0\\
&\boldsymbol{\theta}^{p}(t+1)=\boldsymbol{\theta}(t)+\Gamma\phi_p(t)e(t)\\
&\Delta(t+1)=\begin{cases}0 & \theta_4^p(t+1)\operatorname{sgn}(k_p)\geq c_0\\ \frac{\tau_1}{\tau_2}\left(c_0\operatorname{sgn}(k_p)-\theta_4^p(t+1)\right) & \text{otherwise}\end{cases}
\end{aligned} \tag{68}$$

where $\boldsymbol{\theta}^p=\begin{bmatrix}\boldsymbol{\theta}_1^{pT}\\ \boldsymbol{\theta}_2^{pT}\\ \theta_3^p\\ \theta_4^p\end{bmatrix}$, $\Gamma=diag\{\lambda_1,...,\lambda_{2n}\}$ is a gain matrix with $0<\lambda_i<2, i=1,...,2n$, $\boldsymbol{\theta}(0)$ is

an initial estimate of $\boldsymbol{\theta}^*$. $c_0=\frac{1}{k_p^0}>0$. τ_1 is the last column of Γ and τ_2 is the last element of τ_1.

2.2.2 Lyapunov-MRAC

Lyapunov-MRAC is a method adjusting the controller gain to make the errors between plant model and reference model approaching to zero, which is similar to normalised MIT algorithms. However, when the adaptive gain γ is an excessive value, or performance of y_m is poor, MIT normalised MRAC may obtain an unstable close loop system [1]. Lyapunov-MRAC is introduced to improve this phenomenon.

Let the plant model written in terms of the forward shift operator $\boldsymbol{q}$ as

$$\frac{y_p(t)}{u(t)}=\frac{\boldsymbol{B}_p(\boldsymbol{q})}{\boldsymbol{A}_p(\boldsymbol{q})} \tag{69}$$

Derive a recursive least squares update law for the parameter vectors $\hat{\theta}(t)$ of a controller that asymptotically drives $e_f(t+d)$ to zero. The retrospective cost function decided the performance of $\hat{\theta}$ by assessing the present value of $\hat{\theta}(t)$ in terms of the past behaviour of the linear identification model over the interval $d \le i \le t$. It can be determined as

$$J\left(\hat{\theta}(t),t\right)=\sum_{i=d}^{t}E^{2}\left(\hat{\theta}(t),i\right),\ t\ge d \tag{70}$$

The retrospective error is then defined as

$$E\left(\hat{\theta}(t),i\right)=y_f(i)-b_0u(i-d)-\varphi^T(i-d)\hat{\theta}(t),\ t\ge d \tag{71}$$

Define

$$\begin{aligned}
E\left(\hat{\theta}(t),t\right)&=\begin{bmatrix}E\left(\hat{\theta}(t),d\right)\\ \dots \\ E\left(\hat{\theta}(t),\mathrm{t}\right)\end{bmatrix}\in R^{t-d+1}\\
Y(t)&=\begin{bmatrix}y_f(d)-b_0u(0)\\ \dots \\ y_f(t)-b_0u(t-d)\end{bmatrix}\in R^{t-d+1}\\
\Phi(t)&=\begin{bmatrix}\varphi(0)\\ \dots \\ \varphi(t-d)\end{bmatrix}\in R^{(t-d+1)\times(2n-1)}
\end{aligned} \tag{72}$$

Therefore,

$$E\left(\hat{\theta}(t),t\right)=Y(t)-\Phi(t)\hat{\theta}(t)\in R^{t-d+1} \tag{73}$$

The cost function can be written by the notations as

$$\begin{aligned}
J\left(\hat{\theta}(t),t\right)&=E^T\left(\hat{\theta}(t),t\right)E\left(\hat{\theta}(t),t\right)\\
&=\left[Y(t)-\Phi(t)\hat{\theta}(t)\right]^T\left[Y(t)-\Phi(t)\hat{\theta}(t)\right]
\end{aligned} \tag{74}$$

The recursive least squares estimate for $\hat{\theta}(t)$ is

$$P(t)=P(t-1)-\frac{P(t-1)\varphi(t-d)\varphi^T(t-d)P(t-1)}{1+\varphi^T(t-d)P(t-1)\varphi(t-d)},\ P(0)>0$$
$$\hat{\theta}(t)=\hat{\theta}(t-1)+P(t)\varphi(t-d)\left[y_f(t)-b_0u(t-d)-\varphi^T(t-d)\hat{\theta}(t-1)\right] \quad (75)$$

Where the adaptive control law [26] is

$$u(t)=-\frac{1}{b_0}\left[\varphi^T(t)\hat{\theta}(t)-\boldsymbol{q}^{-n+1}\boldsymbol{B_m}r(t)\right] \quad (76)$$

4. Simulation results

Consider a nonlinear dynamic plant model G expressed by a Hammerstein model [20] as the follow

$$y(t)=0.5y(t-1)+x(t-1)+0.1x(t-2)$$
$$x(t)=1+u(t)-u^2(t)+0.2u^3(t) \quad (77)$$

Convert the plant model into U-model expression as:

$$y(t)=\lambda_0(t)+\lambda_1(t)u(t-1)+\lambda_2(t)u^2(t-1)+\lambda_3(t)u^3(t-1) \quad (78)$$

where

$$\lambda_0(t)=0.5y(t-1)+1+0.3x(t-2)$$
$$\lambda_1(t)=1$$
$$\lambda_2(t)=-1 \quad (79)$$
$$\lambda_3(t)=0.2$$

With the U-model expression, Newton-Raphson algorithm is applied to find out the root of the nonlinear system.

Assume that the virtual model is

$$G_v(z)=\frac{k_v}{z^2-1.3205z+0.4966},\quad k_v \text{ is unknown} \tag{80}$$

In MATLAB, k_v is set as 1 for initialization. For comparison, standard MRAC with MIT normalised algorithm and Lyapunov rules will be present by applying the virtual model (4.4.4) as plant model G and the whole controller scheme shown in Figure 2.

The reference model is

$$G_m(z)=\frac{k_m}{z^2-1.3205z+0.4966},\ k_m=0.1761 \tag{81}$$

3.1 MIT normalised MRAC

Let the adaptive gain $\gamma=0.1$, $\alpha=0.01$, and $\beta=2$. y_r is a square wave signal. The amplitude $r=1$. The results shown in Figure and Figure compared the U-model based MIT normalised MRAC control system applied for Hammerstein model and standard MIT normalised MRAC control system applied for virtual model without root solver U^{-1} nor Hammerstein model.

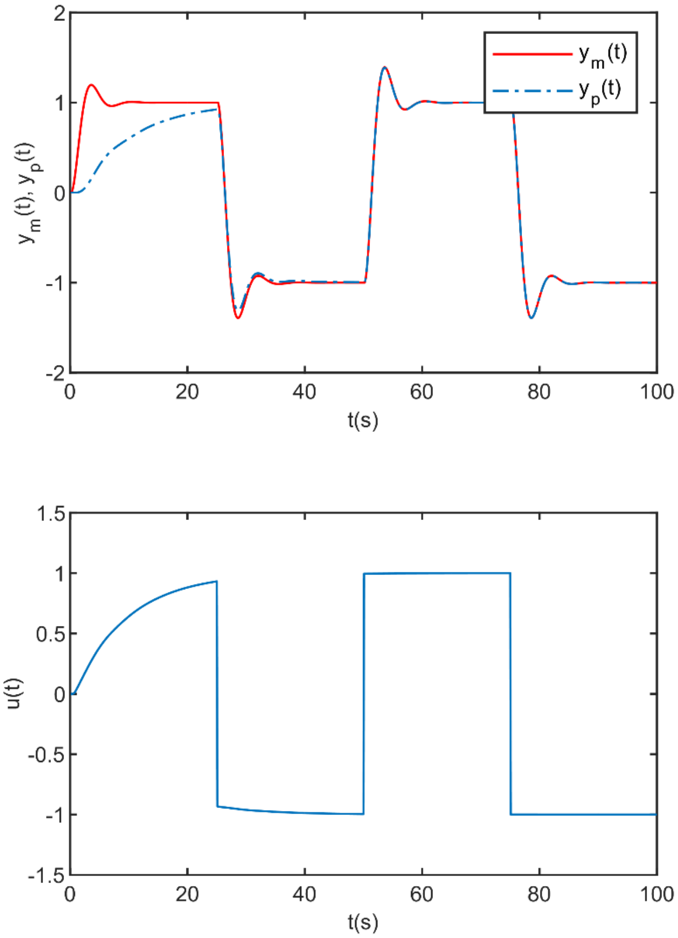


Figure 4 *Standard MIT normalised MRAC control system*

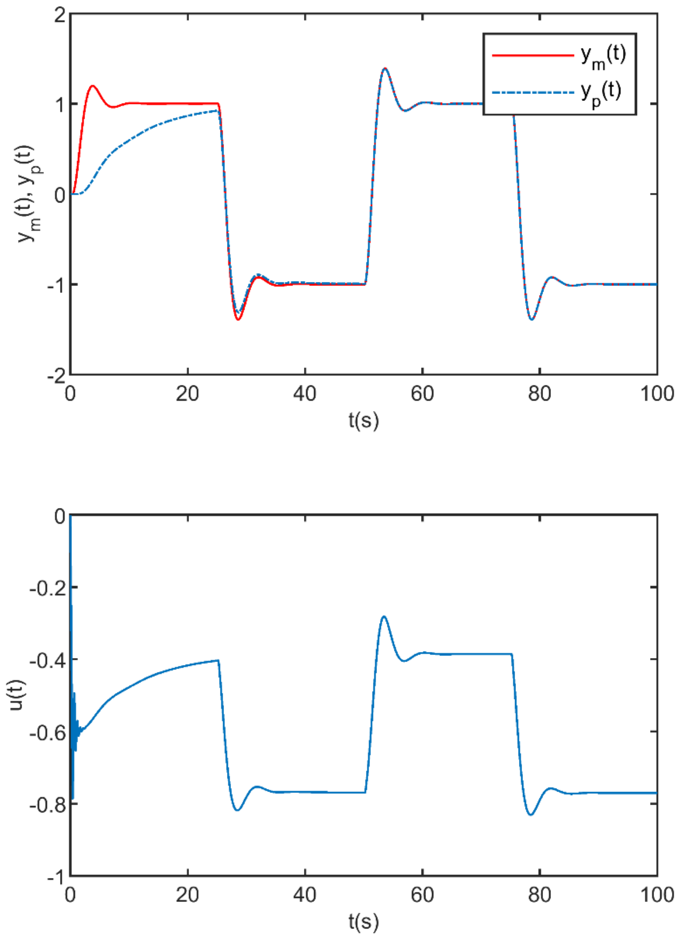


Figure 5 *U-model based MIT normalised MRAC control system*

The plant model of U-model based Lyapunov-MRAC control system is a nonlinear dynamic system (70) and the plant model of standard Lyapunov-MRAC control system is a linear model (73). However, both control system designs maintain the consistent system responses and the same reference outputs. Owing to the linear plant model in standard Lyapunov-MRAC is actually identical with the virtual plant model in U-model based Lyapunov-MRAC control system, the combination output of U-model root solver and actual nonlinear dynamic model can be verified as 1. Accordingly, the nonlinear dynamic system can be applied to obtain the same system response as a MRAC control design for linear plant model.

3.2 Lyapunov-MRAC

Let the adaptive gain $\gamma = 0.1$. y_r is a square wave signal. The amplitude $r = 2$. The results shows in Figure and Figure . When the amplitude rise to $r = 4$, the results is shown in Figure and Figure .

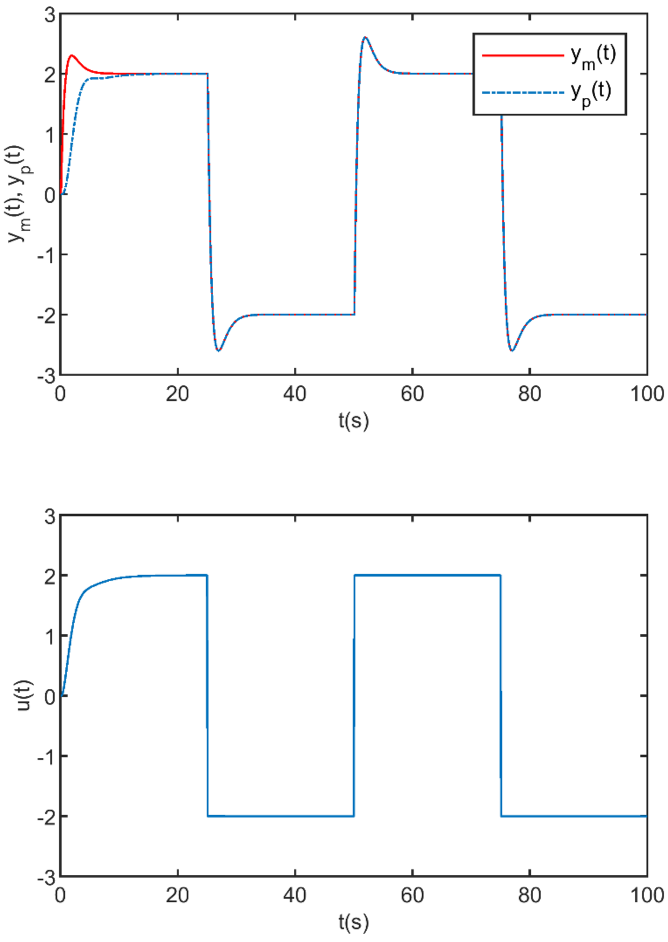


Figure 6 *Standard Lyapunov-MRAC control system when the amplitude r=2*

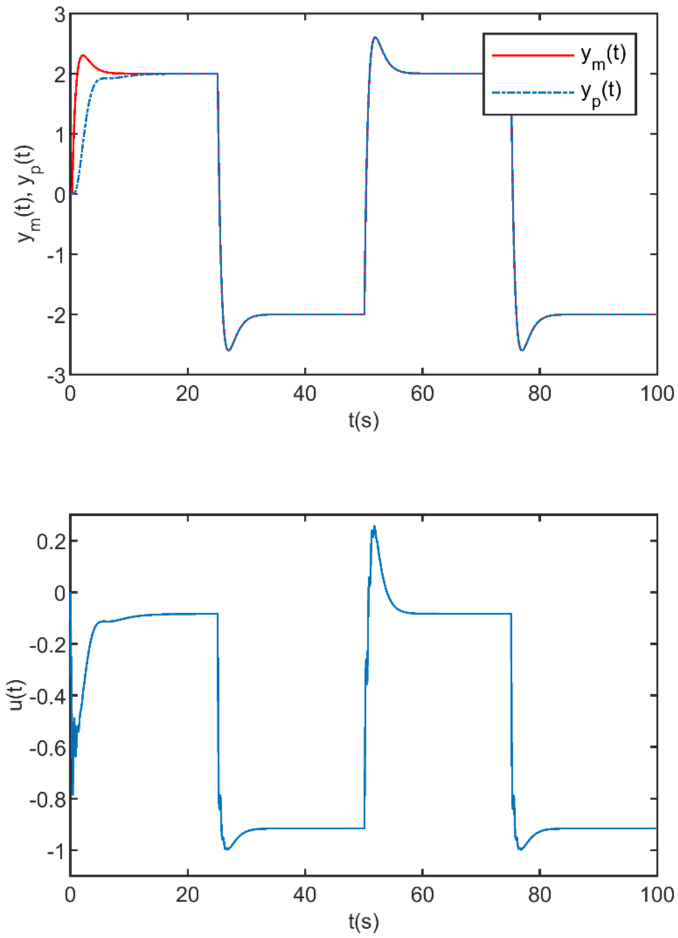


Figure 7 U-model system when the amplitude r=2

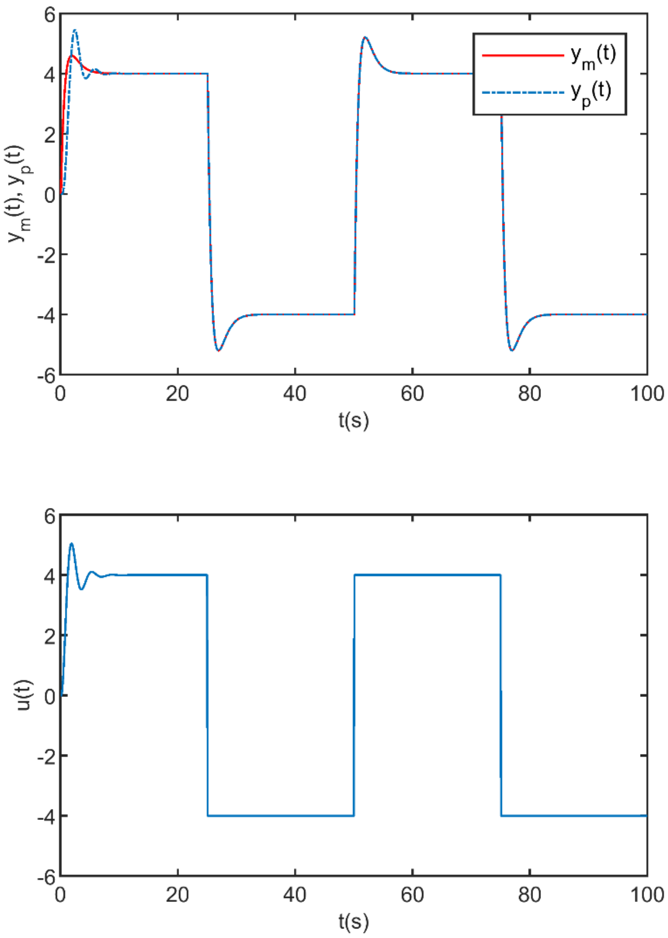
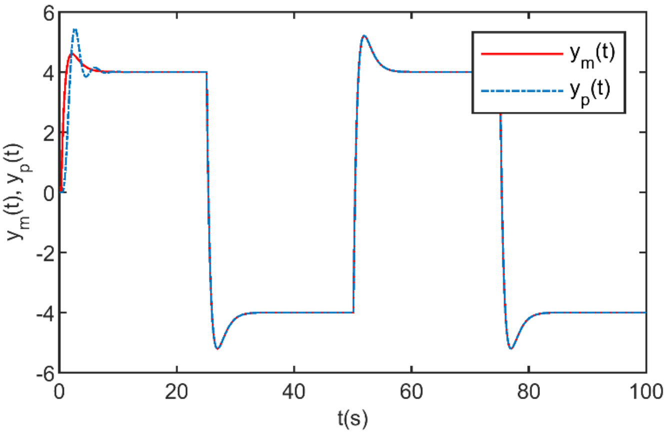


Figure 8 *Standard Lyapunov-MRAC control system when the amplitude r=4*



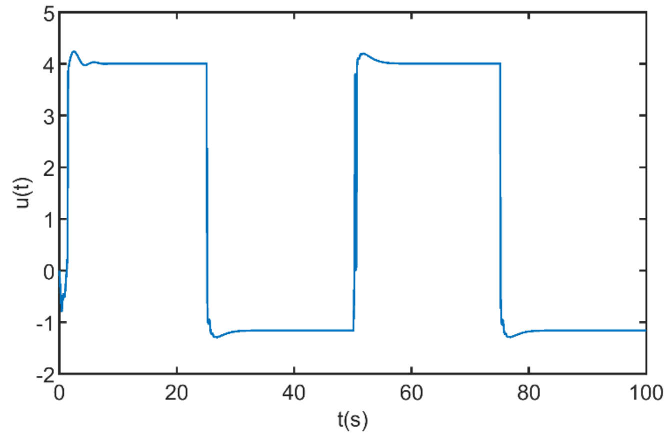


Figure 9 *U-model based Lyapunov-MRAC control system when the amplitude* $r = 4$

Similarly, both standard MIT normalised MRAC control system and U-model based MIT normalised MRAC control system design maintain the coherent system responses and the same reference outputs whether the amplitude changed or not. It shows that the mutative conditions will not influent the intimate relationship between standard Lyapunov-MRAC control system and the U-model based Lyapunov-MRAC control system. Moreover, it can be deduced that U-model based control system is compatible with other control system designs, not intending to improve the performance.

5. Conclusions

This paper proposed a new method to directly apply smooth nonlinear dynamic plant models to MRAC with MIT normalised algorithm and Lyapunov rules. From the results, it is clear to clarify that U-model based MRAC control system maintain the same system response as standard MRAC control system. Particularly, the plant model of U-model based MRAC control system is a nonlinear dynamic model and the plant model of standard MRAC control system is a linear dynamic model. Furthermore, U-model based MRAC control system directly apply the linear controller design for nonlinear dynamic model, which reduce the complexity of system design when other MRACs for nonlinear model need to design whole controller system depended on the plant model. Additionally, U-model based MRAC control system is not antithetic to other MRACs methods. Accordingly, it is possible to develop more U-model based linear adaptive controller systems for nonlinear dynamic plant models. Further studies could be conducted on improvement of comprehensive controller design, such as accuracy, stability, robustness, etc.

Acknowledgements

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

References

[1] K.J. Åström, B. Wittenmark, Adaptive control, Courier Corporation, 2013.

[2] L. Zhao, J. Wang, W. Zhang, Smooth Adaptive Internal Model Control Based on Model for Nonlinear Systems with Dynamic Uncertainties, Math. Probl. Eng. (2016).

[3] Z. Liu, L. Zhao, Q. Zhu, C. Gao, Design of sliding-mode observer for a class of uncertain neutral stochastic systems, Int. J. Syst. Sci. 48 (2017) 1380–1394. doi:10.1080/00207721.2016.1258597.

[4] B. Mirkin, P.O. Gutman, Y. Shtessel, Coordinated decentralized sliding mode MRAC with control cost optimization for a class of nonlinear systems, J. Franklin Inst. 349 (2012) 1364–1379.

[5] B. Mirkin, P.-O. Gutman, Y. Shtessel, C. Edwards, Continuous decentralized MRAC with sliding mode of nonlinear delayed dynamic systems, IFAC World Congr., Milano, Italy. (2011).

[6] B. Mirkin, P.-O. Gutman, Y. Shtessel, Decentralized continuous MRAC with local asymptotic sliding modes of nonlinear delayed interconnected systems, J. Franklin Inst. 351 (2014) 2076–2088.

[7] M. Ganesan, D. Ezhilarasi, B. Jijo, Hybrid model reference adaptive second order sliding mode controller for automatic train operation, IET Control Theory Appl. 11 (2017) 1222–1233. doi:10.1049/iet-cta.2016.1275.

[8] S.H. Rajani, B.M. Krishna, U. Nair, Adaptive and Modified Adaptive Control for Pressure Regulation in a Hypersonic Wind Tunnel, Int. J. Model. Identif. Control. 29 (2018) 78–87.

[9] J. Guo, G. Tao, A discrete-time multivariable MRAC scheme applied to a nonlinear aircraft model with structural damage, Automatica. 53 (2015) 43–52.

[10] M. Mohammadi, A.M. Shahri, Adaptive nonlinear stabilization control for a quadrotor UAV: Theory, simulation and experimentation, J. Intell. Robot. Syst. 72 (2013) 105–122.

[11] X. Zuo, J. Liu, X. Wang, H. Liang, Adaptive PID and Model Reference Adaptive Control Switch Controller for Nonlinear Hydraulic Actuator, Math. Probl. Eng. 2017 (2017).

[12] J. Xie, J. Zhao, Model reference adaptive control for nonlinear switched systems under asynchronous switching, (2017) 3–22. doi:10.1002/acs.2666.

[13] M. Sharifi, S. Behzadipour, G. Vossoughi, Nonlinear model reference adaptive impedance control for human–robot interactions, Control Eng. Pract. 32 (2014) 9–27.

[14] R. Prakash, R. Anita, Neuro-PI controller based model reference adaptive control for nonlinear systems, Int. J. Eng. Sci. Technol. 3 (2011) 44–60.

[15] O.F. Lutfy, Wavelet neural network model reference adaptive control trained by a modified artificial

immune algorithm to control nonlinear systems, Arab. J. Sci. Eng. 39 (2014) 4737–4751.

[16] R. Prakash, R. Anita, Modeling and simulation of fuzzy logic controller-based model reference adaptive controller, Int. J. Innov. Comput. Inf. Control. 8 (2012) 2533–2550.

[17] S. Kersting, M. Buss, Direct and Indirect Model Reference Adaptive Control for Multivariable Piecewise Affine Systems, IEEE Trans. Automat. Contr. 62 (2017) 5634–5649. doi:10.1109/TAC.2017.2690060.

[18] R.J. Pawar, B.J. Parvat, Design and Implementation of MRAC and Modified MRAC technique for Inverted Pendulum, in: Int. Conf. Pervasive Comput., 2015.

[19] X. Wu, L. Liu, Q. Zhu, W. Du, B. Wang, J. Zhang, U-model-based adaptive control for a class of stochastic non-linear dynamic plants with unknown parameters, Int. J. Model. Identif. Control. 13 (2011) 135–143.

[20] Q. Zhu, L.Z. Guo, A pole placement controller for non-linear dynamic plants, Proc. I MECH E Part I J. Syst. 216 (2002) 467–476. doi:10.1243/095965102321042227.

[21] Q. Zhu, D. Zhao, J. Zhang, A general U-block model-based design procedure for nonlinear polynomial control systems, Int. J. Syst. Sci. 47 (2015) 791–804. doi:10.1080/00207721.2015.1086930.

[22] S.A. Billings, Nonlinear system identification: NARMAX methods in the time, frequency, and spatio-temporal domains, John Wiley & Sons, 2013.

[23] S. Akhtar, D.S. Bernstein, Lyapunov-stable discrete-time model reference adaptive control, Int. J. Adapt. Control Signal Process. 19 (2005) 745–767.

[24] R.H. Middleton, G.C. Goodwin, Digital Control and Estimation: A Unified Approach (Prentice Hall Information and System Sciences Series), Prentice Hall Englewood Cliffs, NJ, 1990.

[25] J.L. Li, X.J. Xie, Discrete-time direct model reference adaptive control: a systematic approach, Acta Autom. Sin. 33 (2007) 1048–1052.

[26] L. Hsu, M.C. Minhoto Teixeira, R.R. Costa, E. Assunção, Lyapunov design of multivariable MRAC via generalized passivation, Asian J. Control. 17 (2015) 1484–1497.