The Islamic University – Gaza

Deanery of Graduate Studies

Faculty of Engineering

Electrical Engineering

الجامعة الإسلامية – غزة

عمادة الدراسات العليا

كلية الهندسة

الهندسة الكهربائية

Comparative Study for Controller Design of Time-delay Systems

Wesam H. Sakallah

Advisor

Dr. Hatem A. Elaydi

"This Thesis is Submitted in Partial Fulfillment of the Requirements for the Degree of Master of Science in *Electrical Engineering* "

1430هـ – 2009م

Dedication

To my parents, my brothers, my sisters, my wife, and my lovely kids

Haider, Alia, and Anas

Wesam Haider Sakallah

Acknowledgments

First and foremost, all praise is due to Allah, the Almighty, who gave me the opportunity, strength, and patience to carry out this work.

I wish to express my deepest gratitude to my advisor, Dr. Hatem Elaydi, for his professional assistance, support, advice and guidance throughout my thesis, and to my discussion committee, Dr. Basil Hamed and Dr. Maher Sabra for their acceptance to discuss my thesis.

I would also like to extend my gratitude to my family for providing all the preconditions necessary to complete my studies, also for keeping me in their prayers.

Abstract

Comparative Study for Controller Design of Time-delay Systems

By

Wesam H. Sakallah

Time delays are usually unavoidable in many mechanical and electrical systems. The presence of delay typically imposes strict limitations on achievable feedback performance in both continuous and discrete systems. The presence of the delay complicates the design process as it makes continuous systems to be infinite dimensional and it significantly increases the dimensions in discrete systems. Most of classical methods used controller design cannot be used with delayed systems.

In this study, the delay will be modeled using different approaches such as Pad'e approximation and Smith Predictor in continuous system and modified z-transform in discreet systems. In this study, the delays are assumed to be constant and known. The delays in the system are lumped in the plant model.

This study will show the design of stable and optimal controller for time-delay systems using algebraic Riccati equation solutions and PID control. This study will also present comparison between these controllers.

ملخص البحث

دراسة مقارنة لتصميم المتحكمات ذات التأخير الزمني

إعداد

وسام حيدر ساق الله

التأخير الزمني عادة لا يمكن تجنبه في العديد من الانظمة الميكانيكية والكهربية, كما أن وجود التأخير الزمني في الأنظمة التماثلية أو الرقمية يفرض محددات صارمة على كفاءة هذه الانظمة ويعقد عملية تحليلها وتصميمها.

وجود التاخير الزمني في الأنظمة التماثلية يؤدي إلى زيادة أبعاد مصفوفة النظام إلى مالانهاية كما أنه يعمل على زيادة أبعاد الأنظمة الرقمية. معظم الطرق التقليدية التي تستخدم في عملية تحليل الأنظمة لا يمكن استخدامها في حالة وجود التأخير الزمني في أنظمة التحكم.

في هذه الدراسة, سيتم التعبير عن التأخير الزمني بعدد من الطرق, مثل طريقة بادي التقريبية وطريقة سميث في حالة الأنظمة التماثلية وطريقة زد المعدلة في حالة الانظمة الرقمية. افترضت بأن التأخير الزمني الموجود في النظام ثابت ومعروف وهو عبارة عن جزء من تركيبة النظام. معادلة الريكاتي و المتحكم بي آي دي سيتم استخدامهم لايجاد متحكم مستقر بواسطة ايجاد الحل الامثل ومقارنة أداء النظام عند استخدام كل طريقة على حدى.

Table of Contents

List of Tables

List of Figures

CHAPTER 1

Introduction

1.1 Introduction

Control theory and control engineering deal with various problems that can be encountered in electrical and mechanical systems. The settings of the control problem are outputs, inputs, disturbances and uncertainty.

1. **Outputs**: these are the system variables that are requested to stay within prescribed range. For example, the water level in a tank or temperature in a room, etc. the output variables changes according to certain independent variables which are the inputs.
2. **Inputs:** are the variables that have controlled and direct impact on the output. For example, voltage applied to the motor terminals.
3. **Disturbances:** are the variables that can affect the system output in uncontrolled and unpredicted way. For example, the effect of changing weather conditions on certain plant.
4. **Uncertainty** in control systems arises from the fact that perfect modeling of any physical process is impossible to realize [1]. Modeling of any system is done usually in certain conditions (operating point) which are subject to change with time.

These considerations suggest the following general representation of the plant or system to be controlled.

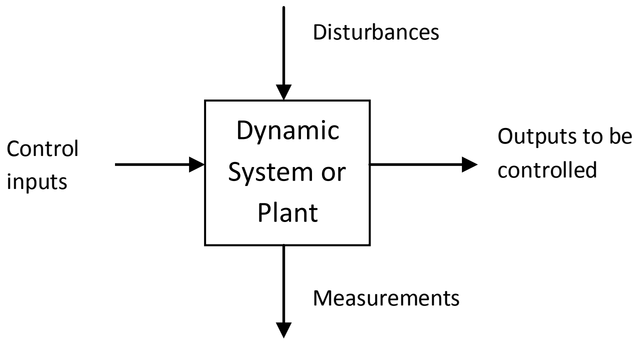


Figure 1.1 A general Plant

Feedback is a necessity in the control process. The input of the system is modified according to the feedback value of the outputs. The controlled system with feedback (closed-loop system) is shown in Figure 1.2.

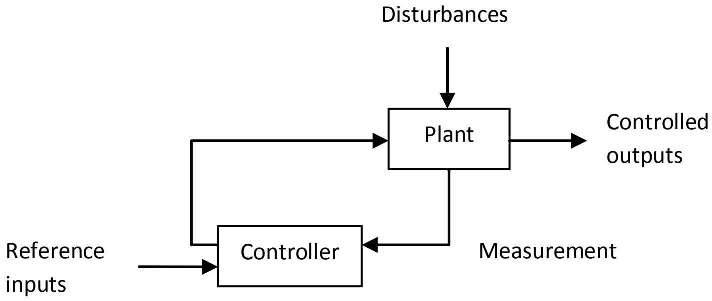


Figure 1.2 A feedback control

1.1.1 Time Delay Systems

Systems with delays can be usually encountered in the real world. When the system involves propagation and transmission of information or material, the delay is certain to occur. The presence of delays complicates the system analysis and the control design [2].

Time delay is defined as the required time between applying change in the input and notices its effect on the system output. The delay time is variable and is dependent on the complexity of the system. The longer the delay, the more complicated the system analysis and design.

Very long delay can be disastrous because it may lead to confuse the delayed system as one without delay. This happens when the effect of the delay cannot be noticed in right time.

In system without delays, the response to any error in the output can be encountered directly by applying a change in the input. If the error has not been immediately reduced or eliminated, more measures need to be taken. However, in a time-delay system, direct effect on the error should not be expected. The effect of applying any change in the input will occur at the output after an inherited delay [3]. So it is very important to study the delay so not to overreact and worsening the error instead of improving it.

The design of feedback control system in presence of the delay becomes more challenging. If the delay exists in the measurement channel, the controller receives the information in wrong time. The other case happens when the delay exists in the actuation channel. In this case, the efficiency of the system is reduced as an effect of applying the control signal in the wrong time (i.e. after the delay) [4].

In the continuous-time system case, the delay is expressed as an infinite dimensions e^{-sh}. As a result, many conventional design methods cannot be applied with presence of delay. In the discrete-time system case, finite dimensional z^{-h} can be considered as part of the system. However, the term z^{-h}, increases the problem dimension. In both cases, it can be seen that the delay in either continuous- or discreet- time systems increase the complexity significantly.

Also, time delays increase the phase lag which leads instability of the control system at relatively lower controller gain. As a result, it put constrains on the performance of the control process.

When the delay exists in the internal state, in addition to the previous pitfalls associated with the delay, it also changes dynamic behavior of the system considerably from the behavior of delay-free systems [5].

If the delay exists in the feedback loop, it can be considered as disturbances. This might result in significant and immediate change in the slope of the system step-response. Hence, the response of system with feedback delay is usually not smooth [6].

1.1.2 Characteristic Equations for Delay Systems

For a given delay element with a delay $h \geq 0$, the system equation may take the form $y(t) = u(t - h)$ (1.1)

Hence, the transfer function of a delay element is given by e^{-sh} and the time delay can be represented in Figure 1.3.

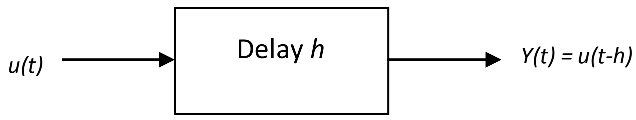


Figure 1.3 Delay representation

In a dynamic feedback system where delay is present, the system equation may take the form

$$\dot{y}(t) + ay(t - h) = u(t) \tag{1.2}$$

The block diagram representation of (1.2) is depicted in Figure 1.4.

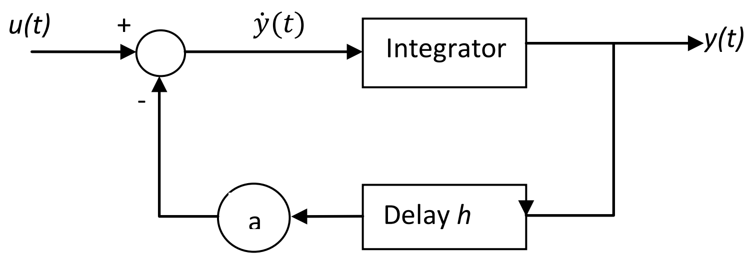


Figure 1.4 A feedback system with delay

If there is a delay in the input, the system equation may take the form

$$\dot{y}(t) + ay(t) = u(t-h) \tag{1.3}$$

The block diagram is depicted in Figure 1.5 or, if the delay is within the loop, the system equation becomes

$$\dot{y}(t) = -ay(t-h) + u(t-h) \tag{1.4}$$

And the block diagram is depicted in Figure 1.6.

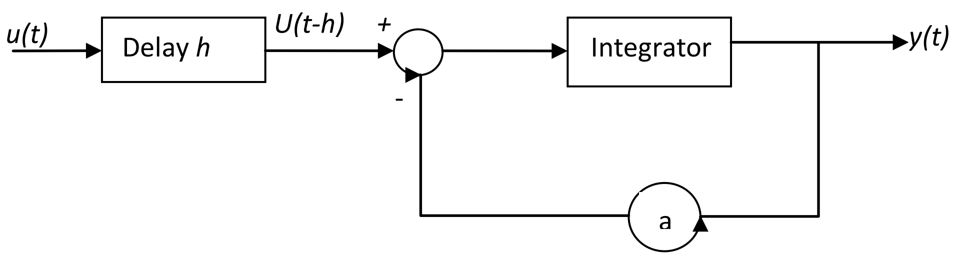


Figure 1.5 Input delay

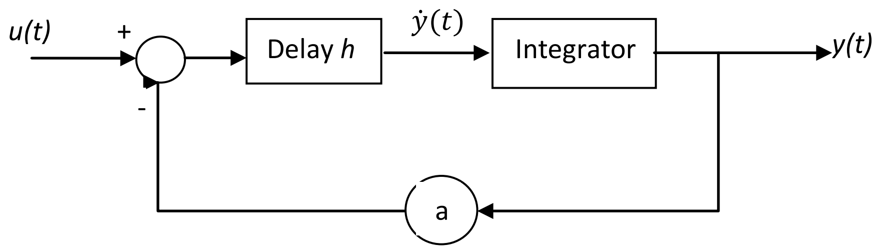


Figure 1.6 Delay within the loop

A higher-order system with multiple delays may be represented by the equation

$$\ddot{y}(t) + a_1\dot{y}(t-h_1) + a_0 y(t-h_0) = u(t) \tag{1.5}$$

The corresponding block diagram is depicted in Figure 1.7[7]. The system (1.5) can be represented in state variable form by introducing.

$$y(t) = x_1(t)\,, \qquad \dot{y}(t) = x_2(t)$$

And writing

$$\begin{pmatrix}\dot{x}_1(t)\\ \dot{x}_2(t)\end{pmatrix} = \begin{pmatrix}0 & 1\\ 0 & 0\end{pmatrix}\begin{pmatrix}x_1(t)\\ x_2(t)\end{pmatrix} + \begin{pmatrix}0 & 0\\ -a_0 & 0\end{pmatrix}\begin{pmatrix}x_1(t-h_0)\\ x_2(t-h_0)\end{pmatrix}$$
$$+\begin{pmatrix}0 & 0\\ 0 & -a_1\end{pmatrix}\begin{pmatrix}x_1(t-h_1)\\ x_2(t-h_1)\end{pmatrix} + \begin{pmatrix}0\\ 1\end{pmatrix} u(t) \qquad (1.6)$$

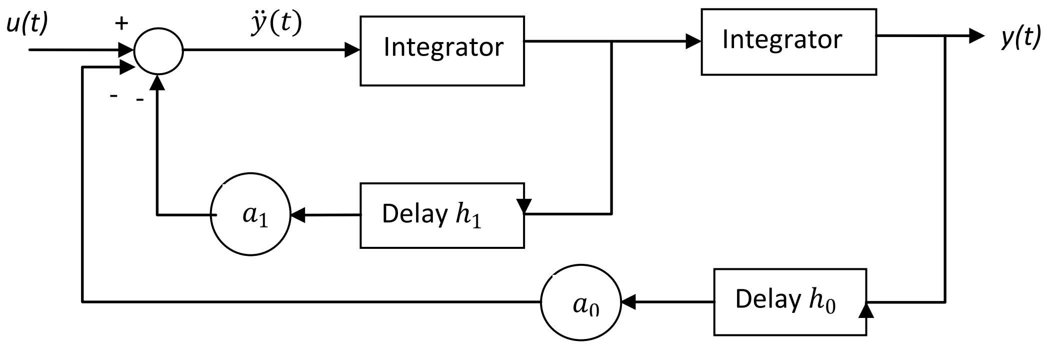


Figure 1.7 Multiple delays

1.1.3 Properties of Time-Delay Systems

In this section, some properties of linear time-delay systems, i.e. systems of the form $\mathrm{G(s)} = \mathrm{G_0(s)e^{-sh}}$ (continuous-time) or $\mathrm{G(z)} = \mathrm{G_0(z)z^{-d}}$ (discrete- time) are stated. Noting that $\mathrm{G_0(s)}$ and $\mathrm{G_0(z)}$ are transfer functions without time delay.

We have the following properties for such systems:

- A pure time-delay e^{-sh} is a linear system.
- A continuous time-delay system is of infinite dimension since an infinite number of values are needed to describe the state of the system at each point of time.
- A continuous time-delay system in state space form can be described by system of differential equations.
- The transfer function $G(s)$ of a continuous time-delay system is not a rational function of s.
- A discrete time-delay system can be described by a system of pure difference equations when the sampling time is constant and the delay is multiple integer of the sampling time.

- Alternatively, if the sampling period is not constant, then a discrete time-delay system cannot be described by pure difference equations. Thus, mixed of differential and difference equations are needed.
- The transfer function G(z) of a discrete time-delay system is a rational function of *z*. G(z) has a finite number of poles, which is consistent with the systems finite dimensionality.

1.2 Scope of my thesis

In this thesis, delays will be lumped into a single delay in the feedback loop, representing delay in control action or delayed measurements. Even for this simple case, the stability problem from the synthesis point of view is complex and challenging, as we shall see later.

First of all, in continuous time-delay system, the delay is modeled by using both of Pad'e and smith predictor approaches. In discrete time-delay systems, a modified *z*-transform will be presented to model constant delays, which are expressed as non-integer multiples of the sampling periods.

In each of the previous cases, the design of stable and optimal controller for time-delay systems will be presented.

1.3 Literature review

Systems with delays are very common. Examples of time-delayed systems are communication networks, chemical processes, bio-systems, and so on. The presence of delays complicates the control design of the system. However, there are different approaches to model the delay such as Smith Predictor and Pad’e approximation methods.

Although Smith Predictor was firstly introduced in late 1950s, it is still fundamental and basic tool for modeling systems with time-delay [8]. What makes Smith predictor so special is that it predicts outputs against time-delays. The achieved systems after prediction can be treated as delay-free systems (i.e. conventional design methods can be used) [8]. However, Smith predictor can be applied only to stable systems. Modified Smith can be applied to unstable systems with certain complex approximations [8]. After modeling the delay, classical PID control can be used [9].

In state-space models, state predictor is used which is similar to modified smith predictor, but it can predict also future states of the systems under consideration [2,10].

Costas Kravaris and Raymond A. Wright presented deadtime compensation for nonlinear processes in (1989) [11]. This paper developed a novel approach for deadtime compensation for nonlinear processes. The approach structure consisted of linearzing state feedback of a nonlinear system and developing smith-predictor to be used in state-space to deal with systems with delay. To compensate for the dead-time linearized system, an open-loop state observer and a linear external controller have been added.

Hsiao-Ping Huang, et al., presented a modified smith predictor at low frequencies with an approximate inverse of dead time in (1990) [12]. Analysis and simulation results showed that the compensator had better disturbance rejection performance than the original Smith predictor.

J. J. Hench, et al., presented dampening controllers via a Riccati equation approach in (1998) [13]. The algorithm presented in this paper did not only introduce stable solution for the system but restricted the poles of the closed-pole system within predefined region in the left half plane. This had an effect of dampening the closed-loop system. This was accomplished by solving a damped algebraic Riccati equation and a degenerate Riccati equation. The solution to these equations was computed using numerically robust algorithms. Riccati can be expressed in format of periodic Hamiltonian system. This periodic Hamiltonian system induced two damped Riccati equations with two different solutions (symmetric and skew symmetric solutions). These two solutions were valid. They produced different closed-loop eigenvalues and different controller gain. This increased the design flexibility by providing an alternative solution.

S.I. Niculescu, Erik I. Verriest., presented a Riccati equation approach to solve delay-independent stability of linear neutral systems: in (1998)[14]. This paper focused on the problem of asymptotic stability when the system has delay in the state of linear neutral systems. Sufficient conditions were given to ensure of the existence of symmetric and positive definite solutions of a continuous Riccati algebraic matrix equation coupled with a discrete Lyapunov equation.

In [15], J. Syder, et al., compared predictive compensation strategies with PID. TA first-order system with delay was assumed to evaluate performance and robustness of predictive and PID compensation strategies. It was demonstrated that for a strong dominant delay, the predictive controllers had better performance than PID based controllers. In case of less dominant delays, some of the PID controller gave comparable or even better performance than the predictive controllers. In non-dominant delay system, PID controller with filtered derivative gave better results than the predictive methods.

In 2003, N. Abe and K.Yamanaka presented the structure of Smith predictor control which was equivalent to Internal Model Control (IMC) in the sense that the delayed behavior of the plant was removed modeling the plant [16].
The disturbance of the input channel can have a very long harmful effect when the system has slow modes. This can be avoided by adding disturbance compensator in the feedback path in the Smith predictor control. The integral error increases in the time delay period (as the the output of the plant does not being affected from the input). This results on increasing the windup phenomena. To solve this problem, Self conditioning anti-windup PI controller was proposed, which includes saturation model in PI controller. The saturation input reduce the integral error and therefore the extreme overshoot response is controlled.

1.4 Statement of the problem

In this thesis, two types of controller for time delayed system are developed: One of them is based on pole placement method, and the other on PID controller.
In continuous time-delayed system, both of Pad’e approximation and Smith predictor techniques are used to model the delay. In Pad'e method, the delay is modeled as a rational transfer function. In Smith predictor, the delay is shifted outside the feedback loop and the system may be considered as a delay free system with certain constraints. In discrete time-delay system, modified *z*-transform will be used to model a constant delay.

The solution of Algebraic Riccati Equation (ARE) and the gains of PID controller are obtained in each case to find the stabilizing solution which is used to obtain the optimal controller of the system and comparison between control performances of these methods are discussed.

1.5 Objectives

The main objective of this research is to design a stabilizing controllers of time delay system by solving an ARE and tuning PID controller. The project deals with the following main points.

- design of Time delay systems,
- solving algebraic Riccati equation,
- tuning PID controller,
- using modified z-transform,
- using pad'e approximation method,
- using smith predictor method,
- satisfying the stability using MATLAB optimization toolbox.

1.6 Organization of Report

In chapter 2, Smith predictor and Pad’e approximation methods which are used to model the delay in continuous systems are presented. Moreover, the modified z-transform which is used to model the delay in discrete systems is introduced.

Chapter 3 deals with the Algebraic Riccati Equation (ARE) approach and PID method which are used to design stabilizing controller to compensate dead-time systems. Theoretical backgrounds for these methods are discussed.

The simulation results are discussed in chapter 4. Modeling the delay in continuous systems has been conducted using both Pad’e approximation and Smith predictor methods. For each case, the controller has been designed using ARE approach and PID method. In digital systems case, the delay is modeled directly to a rational function using modified z-transform. The system has been also designed using ARE approach and PID method. The conclusion and final work are presented in chapter 5.

CHAPTER 2

Delay Modeling

Delay is unavoidable in many control systems. Most of the classical methods that analyze the control system such as root locus and nyquist criterion, cannot deal with delay. Moreover, systems with delay have infinite dimensions which make it impossible to express the system in state space. Hence, there is a need to model the delay which has been done by different approaches. In this chapter, the most famous methods represented are Pad’e approximation and Smith predictor methods in continuous system case and modified z-transform in discrete system case.

2.1 Pad'e approximation method

The Pad'e approximation approximates a pure time delay by a rational transfer function which simplifies the analysis and design of time-delay system. The approximation enables the delay system to be treated as delay-free system [1].

The Pad'e approximation for the term e^{-sh} is given by

$$e^{-sh} \cong \frac{N_r(sh)}{D_r(sh)} \tag{2.1}$$

Where,

$$N_r(sh) = \sum_{k=0}^{r} \frac{(2r-k)!}{k!\,(r-k)!}(-sh)^k \tag{2.2a}$$

$$D_r(sh) = \sum_{k=0}^{r} \frac{(2r-k)!}{k!\,(r-k)!}(sh)^k \tag{2.2b}$$

and r represents the order of the approximation. For example, the first-order Pad'e approximation (r =1) of the time-delay term is

$$e^{-sh} \cong \frac{1-\frac{h}{2}s}{1+\frac{h}{2}s} \tag{2.3}$$

And the second-order Pad'e approximation (r =2) is

$$e^{-sh} \cong \frac{1-\frac{h}{2}s+\frac{h^2}{12}s^2}{1+\frac{h}{2}s+\frac{h^2}{12}s^2} \tag{2.4}$$

2.2 Smith Predictor method

The Smith predictor is probably the most famous method for the control of systems with time delays [3].

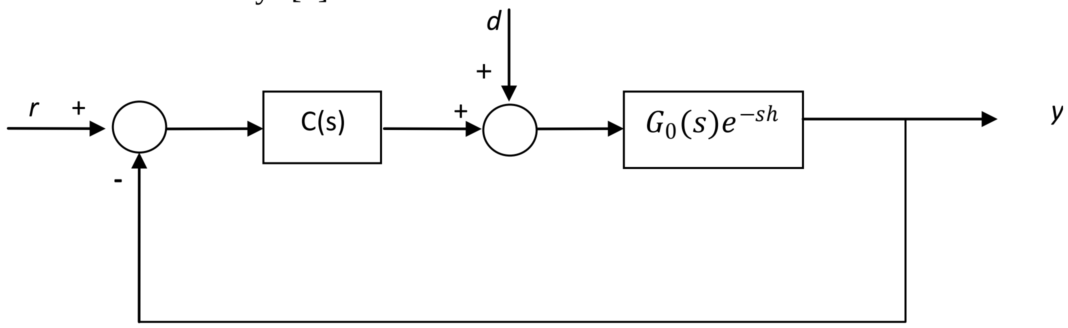


Figure 2.1 A feedback control system with a time delay

A feedback control system with a time delay is shown in Figure 2.1, where $C(s)$ is the controller; $G_0(s)e^{-sh}$ is the plant with a time delay h , where all zeros and poles of $G_0(s)$ are in the left half plane; d is the disturbance. In this case, the transfer function of the closed-loop system with the output $y(s)$ and input $r(s)$ can be formulated as,

$$\frac{y(s)}{r(s)} = \frac{C(s)G_0(s)e^{-sh}}{1+C(s)G_0(s)e^{-sh}} \tag{2.5}$$

From equation 2.5, it is very clear that the location of the closed-loop poles directly related to the time delay h. As result, the stability of the system can be affected by the amount the delay. At certain delay value, the poles might be shifted into the right-half plane; hence making the system unstable [5].

The Smith predictor cancels the effect of the delay by adding output of dead-time free part (corrective signal) to the measured output signal. This result in prediction of what the output would have been if there was no delay.

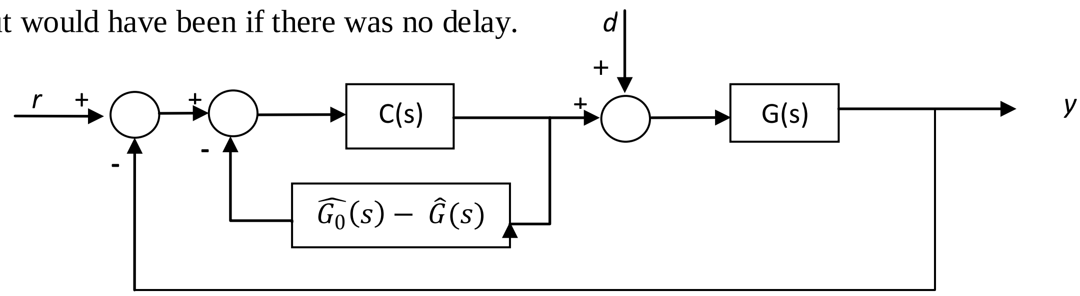


Figure 2.2 Smith Predictor

The classical configuration of a system containing a Smith predictor is depicted in Figure 2.2, where $G(s) = G_0(s)e^{-sh}$, $\widehat{G_0}(s) and\ \hat{G}(s)$ are nominal models of $G_0(s)$ and $G(s)$, respectively. The block $C(s)$ combined with the block $\widehat{G_0}(s) - \hat{G}(s)$ is called the "Smith predictor". When $G_0(s)$ is stable and the disturbance is assumed to be $d = 0$, the transfer function of the closed-loop system relating the output control $u(s)$ to the error signal $e(s)$ is

$$\frac{u(s)}{e(s)} = C_e(s)\, G(s) = \frac{C(s)G(s)}{1 + C(s)[\widehat{G_0}(s) - \hat{G}(s)]} \tag{2.6}$$

Where $C_e(s) = \frac{C(s)}{1+C(s)[\widehat{G_0}(s)-\hat{G}(s)]}$ is the equivalent controller.

If we assume the perfect model matching, i.e.,$\hat{G}(s) = G(s)$, the closed-loop transfer function becomes

$$\frac{y(s)}{r(s)} = \frac{\frac{C(s)G(s)}{1 + C(s)[\widehat{G_0}(s) - \hat{G}(s)]}}{1 + \frac{C(s)G(s)}{1 + C(s)[\widehat{G_0}(s) - \hat{G}(s)]}}$$

$$\frac{y(s)}{r(s)} = \frac{C(s)G(s)}{1 + C(s)\left[\widehat{G_0}(s) - \hat{G}(s)\right] + C(s)G(s)}$$

$$\frac{y(s)}{r(s)} = \frac{C(s)G(s)}{1 + C(s)G_0(s)} = \frac{C(s)G_0(s)}{1 + C(s)G_0(s)}e^{-sh} \tag{2.7}$$

Now from equation 2.7, an equivalent model of the system is depicted in Figure 2.3. It is clear that the delay part is shifted outside the feedback loop. So, the controller design (C(s) depends only on the delay free-part $G_0(s)$ of the plant. It is apparent that the previous constrains on the controller gain do not explicitly exist. This does not mean that the controller gain can take any value. The delay restrict the resultant bandwidth within certain range and therefore the gain cannot be excessively high[17]. At any case, the controller gain is to be used to compromise between the robustness and the speed of the system [5].

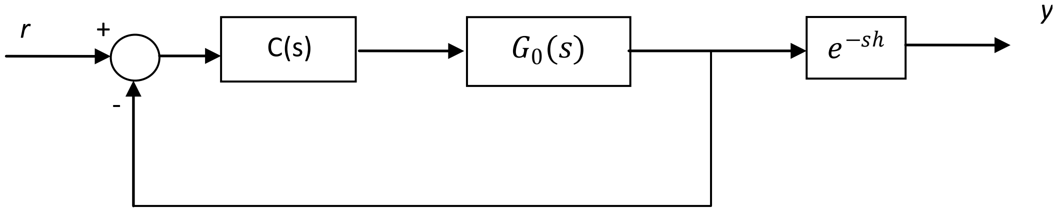


Figure 2.3 SP-based control system: Nominal case and *d = 0*

The main idea of smith predictor is dependent on realizing a perfect model matching between the model and its nominal version. By this modeling and using equation 2.7, it is clear that the stability is no longer related to the time delay as the delay has been removed from the denominator.

An alternative implementation of the Smith predictor is shown in Figure 2.4. Since this configuration makes the design of the Smith predictor more suitable for realization.

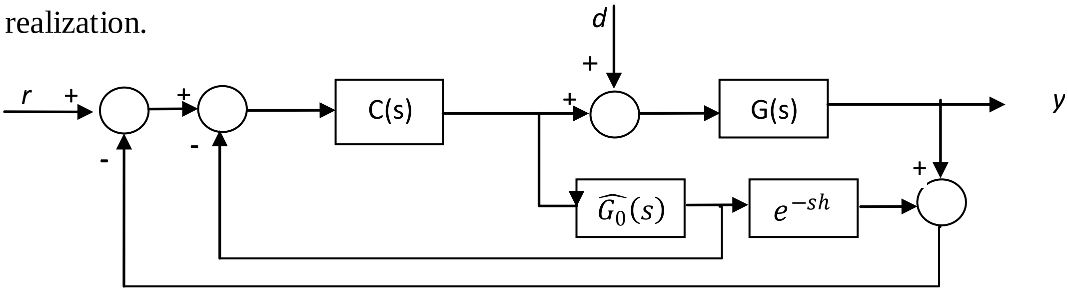


Figure 2.4 An alternative Smith Predictor implementation

Extension of the Smith predictor idea to state space

The compensation of the delay in state space is examined in this section. This does not mean that an alternative or different version of Smith predictor will be used. But rather, the approach is to discuss how the smith predictor can be used with state space model.

Consider a linear process with dead-time of the form

$$\dot{x} = Ax + bu(t-h) \quad (2.8a)$$

$$y = cx \quad (2.8b)$$

If the process is dead-time-free $(h = 0)$ and is subject to the static state feedback $u = v - Kx$, the closed loop transfer function is given by

$$\frac{y(s)}{v(s)} = \frac{CAdj(sI-A)b}{\det(sI-A) + K\ Adj(sI-A)b} \quad (2.9)$$

where $\det(sI - A)$ and $CAdj(sI - A)b$ have all roots in the open left-half plane, and a block diagram is shown in Figure 2.5.

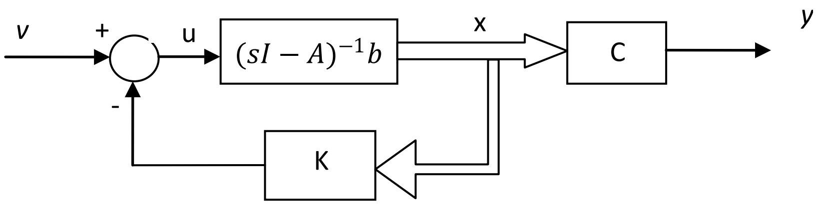


Figure 2.5 State feedback for linear systems without dead-time

In case of system with delay ($h \neq 0$), the same idea of classical smith predictor can be followed in state space. By simulating the difference between the delayed and non-delayed states, a corrective signal is achieved. The corrective signal is added to the state measurements in order to predict what the states would have been if there is no delay (See Figure 2.6).

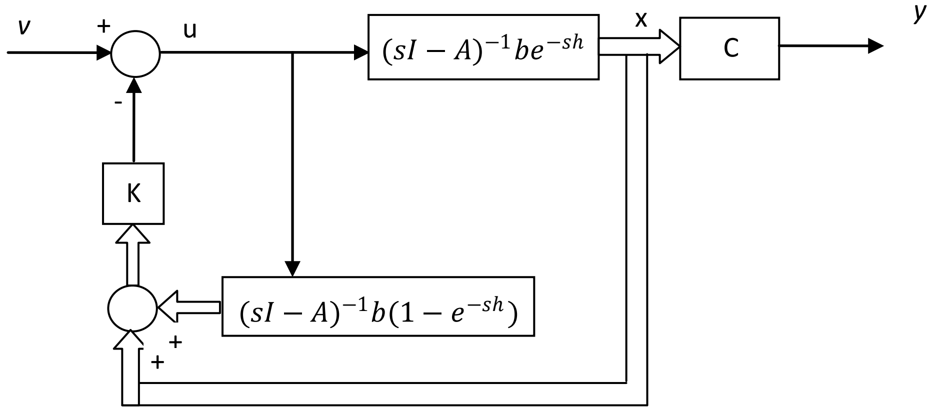


Figure 2.6 Smith Predictor structure in state space for linear systems

The closed-loop transfer function becomes

$$\frac{y(s)}{v(s)} = \frac{c\ Adj(sI - A)b}{\det(sI - A) + K\ Adj(sI - A)b}\ e^{-sh} \tag{2.10}$$

Comparing equations (2.9), and (2.10), the only difference between system with and without delay is the factor e^{-sh}. This factor cannot be eliminated as it would require a non- causal state feedback.

However, as the delay factor e^{-sh} is moved from the denominator, it is possible to use any pole placement formula for delay-free systems to select the closed-loop poles of equation 2.10. Therefore, by using the Smith predictor, the pole placement can be used with delayed systems [11].

2.2 *Z*-transform method

Usually, the calculation of the *z* transfer function is performed using either transform tables or calculus. For this, either the impulse response *g(t)* in analytical form or the *s* transfer function *G(s)* in partial-fraction form is required. If there is a zero-order hold, *G(s)/s* has to be taken into account and special transform tables were developed for this case.

For small sampling periods different discretizing methods of linear differential equations are available. All of them approximate the time-differential with more or less accuracy.

2.3.1 Approximate calculation of *z* transfer function from s transfer function

The continuous-time model of a linear system is expressed usually in frequency domain by using the Laplace transform which is associated with the complex variable 's'. For discrete-time system, z-transform is usually required to deal with system in the frequency domain. The variables *s* and *z* are related to each other in some respects as $z = e^{-sT_s}$, where T_s is the sampling time in seconds. Conversion from continuous-time systems to discreet-time system can be done by different approaches of approximating the relation between s and z

$$z = e^{-sT_S} \cong 1 + sT_S \text{ (Forward difference method)} \tag{2.11}$$

$$z = e^{-sT_s} \cong \frac{1}{1 - sT_s} \text{(Backward difference method)} \tag{2.12}$$

The trapezoidal method for numerical integration leads to the approximation:

$$z = e^{-sT_s} \cong \frac{1 + sT_s/2}{1 - sT_s/2} \text{(Trapezoidal integration method)} \tag{2.13}$$

The approximation given by Equation (2.13) is often called *Tustin's approximation* or *bilinear transformation*. Using the methods above, the approximate *z* transfer function *G(z)* is obtained by simply replacing the argument *s* in *G(s)*, where:

$$s = \frac{z-1}{T_s} \text{(Forward difference or Euler's method)}, \tag{2.14}$$

$$s = \frac{z-1}{zT_s} \text{(Backward difference method)}, \tag{2.15}$$

$$s = \frac{2}{T_s}\frac{z-1}{z+1} \text{(Tustin's approximation method)} , \tag{2.16}$$

Although, these approximations can be easily implemented, they have some drawbacks. For example in the forward difference (equation 2.14), there is possibility of mapping a stable continuous-time system to an unstable discreet time system. This problem can be overcome when the backward difference is used. However, in this case, an unstable discreet-time system can be modelled into a stable continuous-time system.

Using equation 2.16, a stable continuous-time system is always converted to a stable discreet time system and unstable continuous-time system is always converted to an unstable discreet time system [18]. Therefore the Tustin's is preferable method to use for conversion from *s* to *z*, though it is calculation is a little complicated, relatively.

2.3.2 Discrete-time models with dead-time

In case of continuous-time systems the dead-time appears in their mathematical model as a time shift in the output variable. Using the Laplace transform formalism, the presence of the dead-time is characterised by the presence of the term e^{-sh}, where h represents the dead-time value.

The simplest dead-time system can be described by:

$$y(t) = u(t-h) \tag{2.17}$$

or by the corresponding transfer function,

$$D(s) = \frac{Y(s)}{U(s)} = e^{-sh} \tag{2.18}$$

If the dead-time is an integer multiple of the sampling period,

$$\Delta = \frac{h}{T_s} = 1, 2, 3, \dots. \tag{2.19}$$

then according to the shifting theorem results

$$D(z) = \frac{Y(z)}{U(z)} = z^{-\Delta} \tag{2.20}$$

If we generalise the result for a discrete-time system given by the *z* transfer function *G(z)* that is preceded or followed by a dead-time element, the *z* transfer function will be:

$$DG(z) = \frac{Y(z)}{U(z)} = z^{-\Delta}G(z) \tag{2.21}$$

The discrete-time systems that include dead-time have the same type of mathematical model expressed by *z* transfer function as other dynamic elements. Contrary to models of continuous-time systems, dead-time elements can rather easily be included in models of discrete-time systems.

If the dead-time is not an integer multiple of the sampling period but it is a rational multiple of it, then the discrete-time models with such time-delays can be handled with the aid of so called *modified z transform*.

2.3.3 Modified *z*-transform

The modified z-transforms can be used to evaluate a non-integer dead-time by delaying the function by few sampling intervals and subsequently delaying it by a fraction of the sampling interval [19]. This approach was developed so that the value of a function in between the sampling intervals could be evaluated correctly. It evaluates the function from the k^{th} sample to $(k\text{-}1)^{th}$.

The modified *z*-transform can be developed by considering a time function $y(t)$ that is delayed by an amount $h = \Delta\, T_s$, $0 < \Delta \leq 1$, that is, by considering

$$y_0(\mathrm{t}) = y(t-h) = \mathrm{y}(t - \Delta\, T_s) \qquad (2.22)$$

The ordinary *z*-transform of the delayed time function is

$$\mathrm{z}[\mathrm{y}_0(\mathrm{t})] = \sum_{k=0}^{\infty} \mathrm{y}(\mathrm{kT_s} - \mathrm{h})\mathrm{z}^{-k} = \sum_{k=0}^{\infty} \mathrm{y}(\mathrm{kT_s} - \Delta\, \mathrm{T_s})\mathrm{z}^{-k} \qquad (2.23)$$

$1 -$ If $\Delta = \text{integer}$

Then, $\mathrm{z}[\mathrm{y}_0(\mathrm{t})] = \mathrm{z}^{-\Delta}.\, \mathrm{y}(\mathrm{z})$

$$\mathrm{y}(\mathrm{s}) = \mathrm{G}(\mathrm{s}).\mathrm{X}^*(\mathrm{s}) \quad \rightarrow \mathrm{Y}(\mathrm{z}) = \mathrm{G}(\mathrm{z}).\mathrm{X}(\mathrm{z})$$

Then, $\frac{\mathrm{Y}_0(\mathrm{z})}{\mathrm{X}(\mathrm{z})} = \mathrm{z}^{-\Delta}.\, \mathrm{G}(\mathrm{z})$ (2.24)

$2 -$ If Δ is a fraction, or $\Delta = 1 - \mathrm{m}$. $0 < \mathrm{m} < 1$

Then,

$$\mathrm{z_m}[\mathrm{y}_0(\mathrm{t})] = \sum_{k=0}^{\infty} \mathrm{y}(\mathrm{kT_s} + (\mathrm{m}-1)\,\mathrm{T_s})\mathrm{z}^{-k}$$

$$= \sum_{k=0}^{\infty} \mathrm{y}\big((\mathrm{k}+\mathrm{m})\mathrm{T_s} - \mathrm{T_s}\big)\mathrm{z}^{-k}$$

$$= \mathrm{z}^{-1}.\sum_{k=0}^{\infty} \mathrm{y}\big((\mathrm{k}+\mathrm{m})\mathrm{T_s}\big)\mathrm{z}^{-k} = \mathrm{y}(\mathrm{z},\mathrm{m}) = \mathrm{z_m}\{\mathrm{y}(\mathrm{t})\} \qquad (2.25)$$

$3 -$ If $(\mathrm{n}-1) < \Delta < \mathrm{n}$, or $\Delta = \mathrm{n} - \mathrm{m}$, m is integer

then, $\mathrm{y}_\Delta(\mathrm{t}) = \mathrm{y}[(\mathrm{t} + \mathrm{mT_s}) - \mathrm{nT_s}]$

thus, $\mathrm{z}[\mathrm{y}_\Delta(\mathrm{t})] = \mathrm{y}(\mathrm{z},\mathrm{m}) = \mathrm{z}^{-\mathrm{n}} \sum_{k=0}^{\infty} \mathrm{y}(\mathrm{kT_s} + \mathrm{mT_s})\, \mathrm{z}^{-k}$

CHAPTER 3

Controller Design

The main aim of this thesis is to design stabilizing controller to compensate delayed systems using Algebraic Riccati Equation (ARE) approach and PID method. In case of ARE, optimal state feedback gain (K) can be achieved. In PID, the optimal proportional, integral and derivative gains are realized. PID method is simpler than ARE but it gives less accurate results. In this chapter, theoretical background for both methods is discussed.

3.1 Pole placement method (Algebraic Riccati Equation (ARE))

Solution of algebraic Riccati equation (ARE) is required in many important control techniques, including H_∞, H_2(LQG), and the Youla Q-parameterization [17].

The ARE is given by,

$$PA - PRP + Q = 0 \qquad (3.1)$$

Where A, Q, R, $\in R^{nxn}$, where $Q = Q^T$ and $R = R^T$.

The above ARE can also be written in the matrix form

$$\begin{bmatrix} -P & I \end{bmatrix} \begin{bmatrix} A & -R \\ -Q & -A^T \end{bmatrix} \begin{bmatrix} I \\ P \end{bmatrix} = 0 \qquad (3.2)$$

The Riccati Equation is a matrix generalization of the standard quadratic equation. In the 1×1 case, it becomes

$$ap + pa - prp + q = -\mathrm{rp}^2 + 2ap + q = 0$$

This scalar quadratic equation has two solutions p:

$$p = \frac{-2\mathrm{a} \pm \sqrt{4\mathrm{a}^2 + 4\mathrm{rq}}}{-2\mathrm{r}} = \frac{\mathrm{a} \pm \sqrt{\mathrm{a}^2 + \mathrm{rq}}}{\mathrm{r}}$$

Note that $-rp = \mathrm{a} - \left(\mathrm{a} \pm \sqrt{\mathrm{a}^2 + \mathrm{rq}}\right) = \pm\sqrt{\mathrm{a}^2 + \mathrm{rq}}$; there are two possible values of $a - rp$, both symmetric about the origin.

Although there are $\binom{2n}{n} = \frac{(2n)!}{n!n!}$ solutions to the ARE with the $n \times n$ matrix case, there is only one useful solution for control purposes. The eigenvalues of $A - RP$ will correspond to poles of the closed-loop system. We are interested only in stable closed-loop systems. There is only one of the systems which has $\frac{(2n)!}{n!n!}$ possible solutions that make $A - RP$ stable by placing all its eigenvalues and hence the poles in the left-half plane [20]. In the 1 × 1 case above, $-rp = \pm\sqrt{\mathrm{a}^2 + \mathrm{rq}}$, so only one solution of p can satisfy $a - rp < 0$.

Every ARE $A^T P + PA - PRP + Q = 0$ has an associated Hamiltonian matrix

$$H = \begin{bmatrix} A & -R \\ -Q & -A^T \end{bmatrix} \in R^{2nx\,2n} \tag{3.3}$$

We will use the notation P = *Ric (H)* to denote the one solution to the ARE which makes $A - RP$ stable.

It is impossible to have full set of stable eigenvalues when the system is marginally stable [20]. Therefore, choosing A, R, and Q should not cause some of the eigenvalues of $A - RP$ to be purely imaginary (and on the boundary of stability) [20]. And let it include only those choices for H which has stabilizing solutions P satisfying:

$$(i)\ P = P^T \tag{3.4a}$$

$$(ii)\ A^T P + PA - PRP + Q = 0 \qquad and \tag{3.4b}$$

$$(iii)\ Re\big(\lambda_i(A - RP)\big) < 0 \qquad (A - RP\ )stable \tag{3.4c}$$

Lemma 3.1. [21] The ARE (3.1) or (3.2) has a stabilizing solution *P* only if *H* does not have eigenvalues on the $j\omega$-axis.

Proof.

As a matter of fact, Equation (3.2) is part of the following similarity transformation applied to *H* with $\begin{bmatrix} I & 0 \\ P & I \end{bmatrix}$

$$\begin{bmatrix} I & 0 \\ -P & I \end{bmatrix} \begin{bmatrix} A & -R \\ -Q & -A^T \end{bmatrix} \begin{bmatrix} I & 0 \\ P & I \end{bmatrix} = \begin{bmatrix} A - RP & -R \\ 0 & -(A - RP)^T \end{bmatrix}$$

where the (2, 1)-block is set to 0. If $A - RP$ is stable then $-(A - RP)^T$ is antistable. In other words, H does not have eigenvalues on the jω-axis. However, it is not sufficient for the ARE to have a stabilizing solution if H does not have eigenvalues on the jω-axis. A stronger condition is needed.

Lemma 3.2. [21] Suppose H has no imaginary eigenvalues and R is either positive semi-definite or negative semi-definite. Then a stabilizing solution P exists if and only if (A, R) is stabilizable. Furthermore, P is real, symmetric and unique.

Proof.

Only the uniqueness is shown here. But, the proof of this lemma can be found in [21].

Let $P_1 \; and \; P_2$ be solutions of (3.1) such that $A - RP_1 \; and \; A - RP_2$ are stable. Then,

$$A^T P_i + P_i A - P_i R P_i + Q = 0 \qquad \text{(i} = 1,2)$$

Subtract one from the other, then

$$(P_1 - P_2)(A - RP_1) + (A - RP_2)^T (P_1 - P_2) = 0$$

Since $(A - RP_1) \; and \; (A - RP_2)$ are all stable, there is $P_1 - P_2 = 0$.

3.1.1 Properties of Hamiltonian Matrices

A Hamiltonian matrix H has the structure

$$H = \begin{bmatrix} A & -R \\ -Q & -A^T \end{bmatrix} \in R^{2nx\,2n}$$

Where A, Q, R, $P \in R^{nxn}$, where $Q = Q^T \; and \; R = R^T$. Matrices with this special structure have propereties that can be exploited to solve the associated algebraic Riccati equation $A^T P + PA - PRP + Q = 0$. Recall that for P to be valid solution P=Ric (H), it must not only satisfy the ARE, but must also be symmetric $P = P^T$ and must make all the eigenvalues of $A - RP$ stable (real parts less than zero).

Fact 3.1 [20] $\lambda \in C$ is an eigenvalue of H if and only if $-\lambda$ is also an eigenvalue of H.

Proof : *Introduce the matrix*

$$J = \begin{bmatrix} 0 & -I \\ I & 0 \end{bmatrix} \tag{3.5}$$

Note that $J^{-1} = -J$ since

$$JJ^{-1} = \begin{bmatrix} 0 & -I \\ I & 0 \end{bmatrix} \begin{bmatrix} 0 & I \\ -I & 0 \end{bmatrix} = \begin{bmatrix} I & 0 \\ 0 & I \end{bmatrix}$$

Now perform a similarity transformation on H. (Recall that similarity transformation doesn't alter the eigenvalues of a matrix).

$$J^{-1} H J = \begin{bmatrix} 0 & I \\ -I & 0 \end{bmatrix} \begin{bmatrix} A & -R \\ -Q & -A^T \end{bmatrix} \begin{bmatrix} 0 & -I \\ I & 0 \end{bmatrix}$$

$$= \begin{bmatrix} 0 & I \\ -I & 0 \end{bmatrix} \begin{bmatrix} -R & -A \\ -A^T & Q \end{bmatrix}$$

$$J^{-1}HJ = \begin{bmatrix} -A^T & Q \\ R & A \end{bmatrix} = -H^T \qquad (3.6)$$

⇨ $the\ eigenvalues\ of\ H\ and - H^T\ are\ the\ same$

⇨ $the\ eigenvalues\ of\ H\ and - H\ are\ the\ same$

⇨ $\lambda\ is\ an\ eigenvalue\ of\ H\ if\ and\ only\ if - \lambda\ is\ an\ eigenvalue\ of\ H$

Fact 3.2 [20] If Range $\left(\begin{bmatrix} I \\ P \end{bmatrix}\right)$ = { any n-dimensional eigenspace of H}, Where P $\in R^{nxn}$, then,

$$(i)\ A^TP + PA - PRP + Q = 0 \qquad (3.7a)$$

$$(ii)\ H\begin{bmatrix} I \\ P \end{bmatrix} = \begin{bmatrix} I \\ P \end{bmatrix}(A - RP) \qquad (3.7b)$$

Proof : The range of the matrix $\begin{bmatrix} I \\ P \end{bmatrix}$, denote by Range $\left(\begin{bmatrix} I \\ P \end{bmatrix}\right)$, is defined to be the set of vectors $y \in R^{2n}$ such that $y = \begin{bmatrix} I \\ P \end{bmatrix}v$ for some vector v $\in R^n$; that is the range is the span of the columns of $\begin{bmatrix} I \\ P \end{bmatrix}$. If the columns of $\begin{bmatrix} I \\ P \end{bmatrix}$ span an eigenspace of H, this means there exists a matrix Z $\in R^{nxn}$ such that $H\begin{bmatrix} I \\ P \end{bmatrix} = \begin{bmatrix} I \\ P \end{bmatrix}Z$. in other words, $Range\left(H\begin{bmatrix} I \\ P \end{bmatrix}\right)$ = Range $\left(\begin{bmatrix} I \\ P \end{bmatrix}\right)$. Introduce the transformation matrix T and its inverse:

$$T = \begin{bmatrix} I & 0 \\ P & I \end{bmatrix} \qquad and \qquad T^{-1} = \begin{bmatrix} I & 0 \\ -P & I \end{bmatrix}$$

By Assumption, we have $H\begin{bmatrix} I \\ P \end{bmatrix} = \begin{bmatrix} I \\ P \end{bmatrix}Z$ for some Z$\in R^{nxn}$. Therefore, let multiplying by T^{-1} yields

$$T^{-1}H\begin{bmatrix} I \\ P \end{bmatrix} = (T^{-1}HT)\left(T^{-1}\begin{bmatrix} I \\ P \end{bmatrix}\right) = T^{-1}\begin{bmatrix} I \\ P \end{bmatrix}Z$$

$$But \quad T^{-1}\begin{bmatrix} I \\ P \end{bmatrix} = \begin{bmatrix} I & 0 \\ -P & I \end{bmatrix}\begin{bmatrix} I \\ P \end{bmatrix} = \begin{bmatrix} I \\ 0 \end{bmatrix}$$

$$And\ so \quad T^{-1}\begin{bmatrix} I \\ P \end{bmatrix}Z = \begin{bmatrix} Z \\ 0 \end{bmatrix}$$

Evaluate the full similarity transformation on H yields

$$T^{-1}HT = \begin{bmatrix} I & 0 \\ -P & I \end{bmatrix}\begin{bmatrix} A & -R \\ -Q & -A^T \end{bmatrix}\begin{bmatrix} I & 0 \\ P & I \end{bmatrix}$$

$$= \begin{bmatrix} I & 0 \\ -P & I \end{bmatrix} \begin{bmatrix} A - RP & -R \\ -Q - A^T P & -A^T \end{bmatrix}$$

$$= \begin{bmatrix} A - RP & -R \\ -(A^T P + AP - PRP + Q) & -(A - RP)^T \end{bmatrix} \quad (3.8)$$

Finally, substitute this into the original equation:

$$(T^{-1} HT)\left(T^{-1} \begin{bmatrix} I \\ P \end{bmatrix}\right) = \begin{bmatrix} A - RP & -R \\ -(A^T P + AP - PRP + Q) & -(A - RP)^T \end{bmatrix} \begin{bmatrix} I \\ 0 \end{bmatrix}$$

$$= \begin{bmatrix} A - RP \\ -(A^T P + AP - PRP + Q) \end{bmatrix} = T^{-1} \begin{bmatrix} I \\ P \end{bmatrix} Z = \begin{bmatrix} Z \\ 0 \end{bmatrix}$$

Which implies that,

$$(i)\ A^T P + PA - PRP + Q = 0 \quad (3.9a)$$

$$(ii) Z = A - RP \ \rightarrow \quad H \begin{bmatrix} I \\ P \end{bmatrix} = \begin{bmatrix} I \\ P \end{bmatrix} (A - RP) \quad (3.9b)$$

This means that the eigenvalues of the eigenspace spanned by the columns of $\begin{bmatrix} I \\ P \end{bmatrix}$ are the eigenvalues of $A - RP$. Furthermore, note that since the ARE is satisfied

$$T^{-1} HT = \begin{bmatrix} A - RP & -R \\ 0 & -(A - RP)^T \end{bmatrix}$$

This is a block-triangular matrix, and its eigenvalues are the eigenvalues of its block-diagonal components. Therefore, the eigenvalues of H are simply the eigenvalues λ_i of $A - RP$ and the negatives $-\lambda_i$ of these eigenvalues [20].

In order to solve Ric(H), all we need to do is find a basis $\begin{bmatrix} I \\ P \end{bmatrix}$ for the stable eigenspace of H(denoted by $P_{-}(H)$); then $A - R$P will be stable as we require. If H has no eigenvalues on the imaginary axis; then there must exist n stable eigenvalues $Re(\lambda_i) < 0$ and n unstable eigenvalues $Re(\lambda_i) > 0$.

Suppose we find a basis $\begin{bmatrix} P_1 \\ P_2 \end{bmatrix}$ for the stable eigenspace $P_{-}(H)$, where $P_1, P_2 \in C^{nxn}$.

This means that the columns of the matrix span the eigenspace. Then $\begin{bmatrix} P_1 \\ P_2 \end{bmatrix} P_3$ is also a basis for $P_{-}(H)$ if $P_3 \in C^{nxn}$ and P_3 is nonsingular. Let $P_3 = P_1^{-1}$. Then

$\begin{bmatrix} P_1 \\ P_2 \end{bmatrix} P_1^{-1} = \begin{bmatrix} I \\ P_2 P_1^{-1} \end{bmatrix}$ is a basis of $P_{-}(H)$

$\rightarrow \quad P = P_2 P_1^{-1} = Ric(H)$

Even though P_1 *and* P_2 may be complex, $P = P_2 P_1^{-1}$ may be real and symmetric as long as H is in dom Ric.

3.1.2 Solving the ARE

3.1.2a Potter's Method

A basis for $P_{-}(H)$ is found in straightforward manner by computing the eigenvector y_i for the n stable eigenvalues λ_i, then stack and partition the n eigenvectors into a matrix $[y_1, y_2, \dots\dots, y_n] = \begin{bmatrix} P_1 \\ P_2 \end{bmatrix} \in C^{2nxn}$

By the definition of eigenvalues and eigenvectors, then,

$$H \begin{bmatrix} P_1 \\ P_2 \end{bmatrix} = \begin{bmatrix} P_1 \\ P_2 \end{bmatrix} \Lambda$$

Where $\Lambda = diag(\lambda_1, \dots\dots, \lambda_n)$. The matrix $P = Ric(H) = P_2 P_1^{-1}$ solves the ARE.

3.1.2b Laub's Method

The eigenvalues and the eigenvectors can be found by using the Schur method which is done by numerically decomposition of H [20].

Assume $H = UMU^* \; where \; U^*U = I$, where H is the Hamiltonian matrix, and where M is upper triangular. The diagonal elements of M are the eigenvalues of H. Schur decomposition does not give a unique solution as the eigenvalues can be arranged in any order along the diagonal of M.

Assume here that the eigenvalues are ordered so that the n stable eigenvalues are the first n elements on the diagonal of M. Then

$$H = \begin{bmatrix} U_{11} & U_{12} \\ U_{21} & U_{22} \end{bmatrix} \begin{bmatrix} M_{11} & M_{12} \\ 0 & M_{22} \end{bmatrix} \begin{bmatrix} U_{11}^* & U_{21}^* \\ U_{12}^* & U_{22}^* \end{bmatrix} \quad (3.10)$$

$Where \; Re(\lambda_i(M_{11})) < 0 \; and \; Re(\lambda_i(M_{22})) > 0,$

$$and \; where \; M_{11} \; and \; M_{22} are \; both \; triangular.$$

Then, $H \begin{bmatrix} U_{11} \\ U_{21} \end{bmatrix} = \begin{bmatrix} U_{11} & U_{12} \\ U_{21} & U_{22} \end{bmatrix} \begin{bmatrix} M_{11} & M_{12} \\ 0 & M_{22} \end{bmatrix} \begin{bmatrix} I \\ 0 \end{bmatrix} = \begin{bmatrix} U_{11} \\ U_{21} \end{bmatrix} M_{11}$ (3.11)

$\rightarrow \begin{bmatrix} U_{11} \\ U_{21} \end{bmatrix} \; is \; a \; basis \; for \; the \; stable \; eigenspace \; P_{-}(H)$ $(3.12a)$

$\rightarrow P = U_{21} U_{11}^{-1} = Ric\,(H)$ $(3.12b)$

Note: MATLAB is used Laub's Method for solving continuous-time algebraic Riccati equations and discrete-time algebraic Riccati equations (see appendix A)

3.1.3 Damping controller via a Riccati equation approach for State feedback with prescribed degree of stability

Consider the linear time-invariant controllable system

$$\dot{x}(t) = Ax(t) + Bu(t) \;; \qquad x(t_0) \triangleq x^0 \tag{3.13}$$

With linear control law of the form

$$u = -Kx \tag{3.14}$$

And the feedback gain matrix K may be selected to place the poles of the closed-loop system $\dot{x}(t) = (A - BK)x(t)$ at certain desired locations.

In the following subsection, transformations that convert the system into an infinite-time are introduced. This is done to ensure that the closed-loop poles lie to the left of the line $R(s) = -\alpha$ for a prescribed $\alpha > 0$.

Accordingly, Let us define

$$\hat{x}(t) = e^{-\propto t}\, x(t) \tag{3.15}$$

$$\hat{u}(t) = e^{-\propto t}\, u(t) \tag{3.16}$$

In terms of $\hat{x}(t)$ and $\hat{u}(t)$, eq. (3.13) is expressed as

$$\dot{x}(t) = e^{\propto t}\, \dot{\hat{x}}(t) + \propto\; e^{\propto t}\, \hat{x}(t)$$

$$and \;\; \dot{x}(t) = Ae^{\propto t}\, \hat{x}(t) + \; B\, e^{\propto t}\, \hat{u}(t)$$

$$\dot{\hat{x}}(t) = (A - \propto I)\hat{x}(t) + \; B\;\, \hat{u}(t) \tag{3.17}$$

It can be easily established that given the controllable pair $\{A, B\}$, the pair $\{A - \propto I, B\}$ will also be controllable. Similarly if the pair $\{A, C\}$is observable, the pair $\{A - \propto I, C\}$ will also be observable [22].

Therefore, there exists a feedback control law

$$\hat{u}(t) = -K_{\propto}\, \hat{x}(t) \tag{3.18}$$

Such that the resulting closed-loop system, given in the following equation, is asymptotically stable.

$$\dot{\hat{x}}(t) = (A - BK_{\propto} - \propto I)\hat{x}(t) \tag{3.19}$$

The feedback matrix

$$K_{\propto} = \; R^{-1}B^T P_{\propto} \tag{3.20}$$

Where $P_{\propto}$ is given by the solution of the algebraic Riccati equation

$$(A^T - \propto I)P_{\propto} + P_{\propto}(A - \propto I) - P_{\propto}BR^{-1}B^T P_{\propto} + Q = \; 0 \tag{3.21}$$

Using the transformations (eq. 3.15, 3.16), and from (3.18) and (3.20)

$$u(t) = \; -K_{\propto}\, x(t)$$

$$= \; R^{-1}B^T P_{\propto}\, x(t) \tag{3.22}$$

Thus the control law has constant feedback gains; the resulting closed-loop system will therefore be time invariant.

$$\dot{x}(t) = (A - BK_{\propto})x(t) \tag{3.23}$$

The poles of the system (3.19), given by the eigenvalues of $(A - BK_{\propto} - \propto I)$, have negative real parts. As a consequence, the poles of the system (3.23) have negative real part. This is because the eigenvalues of $(A - BK_{\propto})$ (in equation 3.23) which are less by $\propto$ than the eigenvalues of $(A - BK_{\propto} - \propto I)$ (in equation 3.19) definitely posses real parts less than $-\propto$.

3.1.4 Pole Assignment in a Specified Disk

The problem of assigning all poles of a closed-loop system in a specified disk by state feedback is considered for both continuous and discrete systems. A state feedback control law is determined by using a discrete Riccati equation. This kind of pole assignment problem is named *D*-pole assignment [23].

3.1.4a Pole assignment

In this section, the continuous system (equation 3.24a and 3.24b) and the discrete system, ((3.25a) and (3.25b)) are considered

$$\dot{x}(t) = Ax(t) + Bu(t) \tag{3.24a}$$

$$y(t) = Cx(t) \tag{3.24b}$$

$$x_{k+1} = Ax_k + Bu_k \tag{3.25a}$$

$$y_k = Cx_k \tag{3.25b}$$

where *u* is an *m*-dimensional input vector, *x* is an *n* dimensional state vector, and *y* is a *p*-dimensional output vector, and *A, B,* C are constant matrices of appropriate dimensions. It is also assumed that the pair *(A, B)* is controllable (or reachable).

The problem to be considered is to determine the state feedback

$$u = -Kx \tag{3.26}$$

$$u_k = -Kx_k \tag{3.27}$$

Such that all poles of the closed-loop system, i.e. the roots of equation (3.28), may be shown in Figure 3.l(a) for the continuous system and in Figure 3.l(b) for the discrete system.

$$\det\big(sI - (A - BK)\big) = 0 \tag{3.28}$$

The D-pole assignment problem is defined as the assignment of all the poles of the closed-loop system (i.e. the eigenvalues of the matrix) should be located in specified Disk D as shown in Figure 3.1.

Note that the following conditions are satisfied for both continuous and discrete systems.

Lemma *3.3:* [23] Consider the matrix equation

$$-\propto A^{*}P - \propto PA + A^{*}PA + (\propto^2 - r^2)P = -Q \tag{3.29}$$

where Q is an arbitrarily positive definite matrix, and * denotes the conjugate transpose of a matrix, and $\propto$, r are scalars. Then, the eigenvalues of matrix A are located within the specified disk as shown in Figure 3.1 if and only if there exists a positive definite solution $\boldsymbol{P}$ satisfying (3.29).

Proof*:* Let λ,v be an eigenvalue and right eigenvector of A, then

$$Av = \lambda v\,, \qquad v^{*}A^{*} = \lambda^{-}v^{*} \tag{3.30}$$

and substituting these expressions into (3.29) yields

$$-\propto (v^{*})^{-1}\lambda^{-}v^{*}\,P - \propto P\lambda + (v^{*})^{-1}\lambda^{-}v^{*}\,P\lambda + (\propto^2 - r^2)P = -Q$$

$$-\propto \lambda^{-}\,P - \propto P\lambda + \lambda^{-}\lambda P + (\propto^2 - r^2)P = -Q$$

$$\{-\propto (\lambda^{-} + \lambda) + |\lambda|^2 + (\propto^2 - r^2)\}P = -Q \tag{3.31}$$

Let $\lambda = x + j\,y$ and using it in (3.31), we obtain

$$\{-\propto (x - j\,y\; + x + j\,y\,) + (x^2 + y^2) + (\propto^2 - r^2)\}P = -Q$$

$$\{-2 \propto x + (\,x^2 + y^2) + (\propto^2 - r^2)\}P = -Q \tag{3.32}$$

$$\{(x - \alpha)^2 +\; y^2 - r^2\}P =\; -Q$$

Since Q is positive definite, the positivity of P yields

$$(x - \alpha)^2 +\; y^2 - r^2\} < 0 \tag{3.33}$$

which means the condition that all eigenvalues of the matrix A should be located in a specified disk with radius $\boldsymbol{r}$ and center at $\alpha + j0$ (Figure 3.1).

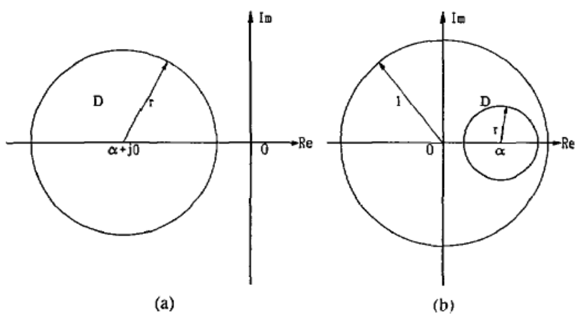


Figure 3.1(a) Disk D in the left half of the complex plane,

(b) Disk D in the unit disk with the center at the origin

On the contrary, let the eigenvalues of matrix A be inside a specified disk of Figure 3.1. That is, the eigenvalues of $\frac{(A-\alpha I)}{r}$ are within the unit disk with its center at the origin.

Consider the following matrix:

$$P_k = \sum_{i=0}^{k} \frac{1}{r^{2(i-1)}} \left((A-\alpha I)^*\right)^i Q (A-\alpha I)^i \tag{3.34}$$

Since $\boldsymbol{Q} > 0$, it is shown that the right term of (3.34) is positive definite for all $\boldsymbol{i}$. From (3.29) and (3.34), we can obtain

$$\frac{1}{r^2} (A-\alpha I)^* P_k (A-\alpha I) - P_k$$

$$= -\frac{1}{r^2} Q + \frac{1}{r^{2(k+2)}} \left((A-\alpha I)^*\right)^{k+1} Q (A-\alpha I)^{k+1} \tag{3.35}$$

And

$$P_{k+1} - P_k = \frac{1}{r^{2(k+2)}} \left((A-\alpha I)^*\right)^{k+1} Q (A-\alpha I)^{k+1} \tag{3.36}$$

By hypothesis of the matrix A, the following is yielded as $k \to \infty$

$$\left(\left\|\frac{A-\alpha I}{r}\right\|\right)^{k+1} \to 0$$

which means that the right-hand side of (3.36) and the second term of (3.35) become zero. Therefore, it proves that there exists a positive definite solution $\boldsymbol{P}$ such as $\lim_{k\to\infty} P_k = P$ which satisfies (3.29).

Remark: We may see from lemma 3.3 that a different expression for (3.29) can be given

$$\frac{(A-\alpha I)^T}{r} P \frac{(A-\alpha I)}{r} - P = -\frac{Q}{r^2}$$

and that the eigenvalues of the matrix $\frac{(A-\alpha I)}{r}$ are located within a unit disk with its center at the origin if and only if there exists a positive definite solution P that satisfies (3.29). In lemma 3.2, we considered only the case such that Q is positive definite. Also, in the case that $Q = H^T H$, the lemma is satisfied for the positive semidefinite Q as far as the pair (A, H) is observable.

Consider the state feedback

$$u = -Kx$$

then, from *Lemma 3.3*, we can have the conditions that the eigenvalues of the closed-loop matrix $(A - BK)$ should be located within a specified disk D.

Theorem 3.1: [23] Consider the following matrix equation:

$$-\propto (A-BK)^{*}P - \propto P(A-BK) + (A-BK)^{*}P(A-BK) + (\propto^{2} - r^{2})P = -Q \qquad (3.37)$$

where Q is positive definite. Then the eigenvalues of $(A - BK)$ are within a specified disk *D* as shown in Figure 3.1 if and only if there exists a positive definite solution *P* satisfying (3.37).

As a method to choose a state feedback law that satisfies our problems, We present the following theorem by using the discrete Riccati equation.

Theorem 3.2: [23] The state feedback law

$$u = (r^{2}R + B^{T}PB)^{-1}B^{T}P(A - \alpha I)x \qquad (3.38)$$

assigns all the closed-loop poles of a continuous system (3.24) or discrete system (3.25) in the disk D shown in Figure 3.1, where P is a positive definite symmetric solution of the Riccati equation

$$P = \frac{(A-\alpha I)^{T}}{r} P \frac{(A-\alpha I)}{r} + H^{T}H - \frac{(A-\alpha I)^{T}}{r} PB(r^{2}R + B^{T}PB)^{-1}B^{T}P\frac{(A-\alpha I)}{r} \qquad (3.39)$$

R is an arbitrarily positive definite matrix, and *H* is a matrix such that the pair *(A, H)* is observable.

***Proof**:* Substituting (3.38) into (3.39), we can rewrite (3.39) as the following:

$$-\propto (A-BK)^{T}P - \propto P(A-BK) + (A-BK)^{T}P(A-BK) + (\propto^{2} - r^{2})P = -r^{2}(K^{T}RK + H^{T}H) \qquad (3.39.1)$$

Where

$$K = (r^{2}R + B^{T}PB)^{-1}B^{T}P\,(A - \alpha I)$$

In (3.39.1), let

$$Q = r^{2}(K^{T}RK + H^{T}H)$$

and using the above *Lemma 3.3,* its *remarks* and *Theorem 3.1,* it proves that the poles of the closed-loop system, i.e., the eigenvalues of the matrix $(A - BK)$ are located within the specified disk *D*.

3.1.4b Maximal Disk with Prescribed δ and γ

It can be proved that when the radius (*r*) of the circle gets small, the robustness deteriorates [24]. Therefore, we need to maximize the radius of the disk in which the closed-loop poles are to be relocated. The following results provide the exact relationship between γ (stability margin), δ (damping ratio), *r* (radius of the disk) and α (center of the disk):

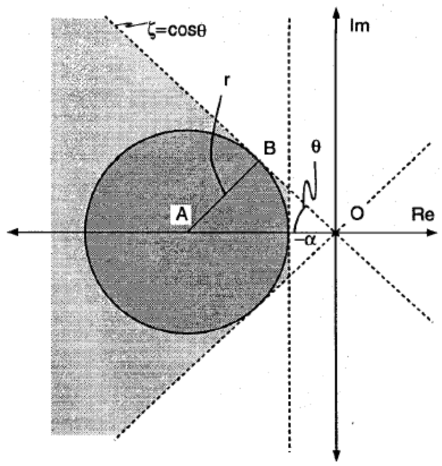


Figure 3.2: Desired disk for closed-loop poles

Given the stability margin γ and the damping ratio δ, the disk of maximum radius that satisfies both γ and δ has a radius

$$r = \gamma \frac{\cos(90 - \cos^{-1}\delta)}{1 - \cos(90 - \cos^{-1}\delta)} \tag{3.40}$$

And it is centered at $-\alpha = -(r + \gamma)$

To show the above result, it is clear from the right triangle *OBA* in Figure 3.2 that

$$\cos(90 - \theta) = \frac{AB}{OA} = \frac{r}{r + \gamma}$$

Where $\theta = \cos^{-1}\delta$, and $AB = r$. Then the proof follows trivially.

Remark: If only γ is specified, one may select arbitrarily large r so long as $\alpha = -(r + \gamma)$. Similarly, if only δ is specified, there is a complete freedom in selecting the radius so long as Eq. (3.40) is satisfied and the disk is centered at $-(r + \gamma)$.

3.2 PID Control method

The Proportional Integral Derivative (PID) controller is probably the most widely used controller in the process industry [25]. It gained its fame for its simplicity of having only three parameters which are proportional, integral and derivative. Different combinations of parameters result in different controllers, such as PI controllers and PD controllers. The standard form of a PID controller is given in the *s*-domain as[26]

$$C(s) = P + I + D = K_p + \frac{K_i}{s} + K_d s \tag{3.41}$$

where K_p, K_i *and* K_d are called the proportional gain, the integral gain and the derivative gain respectively. The standard form can be expressed as

$$C(s) = K_p \left(1 + \frac{1}{T_i s} + T_d s\right) \tag{3.42}$$

Another PID structures can also be used such as the series form with transfer function

$$C(s) = \left(K_p + \frac{K_i}{s}\right)(K_d s + 1) \tag{3.43}$$

and it can also be expressed as

$$C(s) = K_p(1 + \frac{1}{T_i s})(T_d s + 1) \tag{3.44}$$

where T_i is called the integral time constant or reset time and T_d is called the derivative time constant or rate time.

In the time domain, the output of the PID controller *u* can be described as follows:

$$u(t) = K_p e(t) + K_i \int e(t)dt + K_d \frac{de(t)}{dt} \tag{3.45}$$

where $e(t)$ is the input to the controller.

Because of its simplicity, the PID controller can easily be implemented using different tools such as mechanical, or electronic devices, or software. It takes into account the three components, *I*, *P*, and *D* which represent the past, present and future, information of the control error, respectively. So, it is able to provide acceptable control performance [27].

The effects of each parameter on the step response of the system is illustrated in Table 3.1

Table 3.1 Effects of P, I, and D on the step response

Parameter	Rising Time	Overshoot	Settling Time	S.S. Error
K_p	Decrease	Increase	Small Change	Decrease
K_i	Decrease	Increase	Increase	Eliminate
K_d	Small Change	Decrease	Decrease	Small Change

As a matter of fact, more than 90% industrial processes are controlled by PID controllers mostly PI controllers [28]. PID controllers are also widely used with time-delay systems.

3.2.1 Tuning Methods for PID Controllers

The parameters of a PID controller can be tuned by many methods, such as trial-and-error tuning, empirical tuning like the well-known Ziegler‑Nichols method, analytical tuning, prediction approach tuning, optimized tuning and auto-tuning with identification of the plant model [29,30].

3.2.1a The Ziegler-Nichols step response method

The Ziegler-Nichols step response method is an experimental tuning method for open-loop plants. The first step in this method is to calculate two parameters *A* and *L* that characterize the plant. These two parameters (*A*, *L*) can be determined graphically from a measurement of the step response of the plant as illustrated in Figure 3.3. First, the point on the step response curve with the maximum slope is determined and the tangent is drawn. The intersection of the tangent with the vertical axis gives *A*, while the intersection of the tangent with the horizontal axis gives *L*.

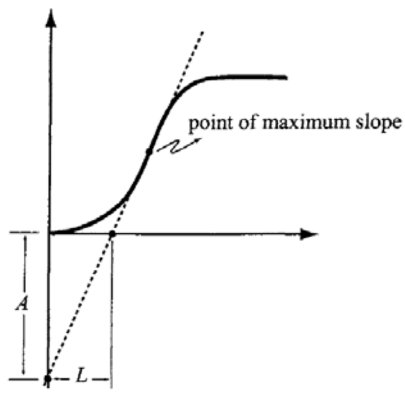


Figure 3.3 Graphical determination of parameters A and L.

Once A and L are determined, the PID controller parameters are then given in terms of A and L by the following formulas:

$$K_p = \frac{1.2}{A} \tag{3.46a}$$

$$K_i = \frac{0.6}{AL} \tag{3.46b}$$

$$K_d = \frac{0.6L}{A} \tag{3.46c}$$

When using the previous formulas for Kp, Ki, and Kd , the amplitude decay ratio is 0.25, which means that the first overshoot decays to $\frac{1}{4}\ th$ of its original value after one oscillation. It has been verified by several experimental results that this method gives a small settling time [1].

3.2.1b The Ziegler-Nichols frequency response method

The Ziegler-Nichols frequency-response method is a closed-loop tuning method. In this method, the two parameters to be calculated are the ultimate gain K_u and the ultimate period T_u which can be calculated experimentally in the following way:

Set the integral and differential gains to zero and hence the controller become in the proportional mode only. Close loop system is shown in Figure 3.4. The proportional gain K_p is then increased slowly until a periodic oscillation in the output is observed. This critical value of K_p is called the ultimate gain K_u.The resulting period of oscillation is referred to as the ultimate period T_u. Based on K_u and T_u, the Ziegler-Nichols frequency response method gives the following simple formulas for setting PID controller parameters according to table 3.2 :

Table 3.2 Ziegler-Nichols tuning formulas

Type of controller	K_p	T_i	T_d
P	$0.5K_u$	-	-
PI	$0.45K_u$	$0.833T_u$	-
PID	$0.6K_u$	$0.5\ T_u$	$0.125\ T_u$

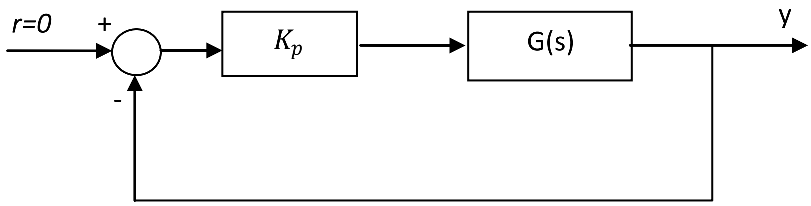


Figure 3.4 The closed-loop system with the proportional

3.2.2 PID for Delayed Processes

The degree of the numerator is higher than the degree of the denominator in representations of the PID controller given in Equations (3.42) and (3.44). Therefore, the ideal PID controller cannot be implemented in practice; because (C(s) is improper). A derivation action is needed to make C(s) proper which is done usually by means of low pass filter. For example, for the standard form, the transfer function of this part of the controller is D(s)

$$D(s) = \frac{K_p(T_d s)}{\alpha T_d s + 1} \tag{3.47}$$

where $\alpha \in (0, 1)$. For industrial controllers, the value of α normally varies between 0.05 and 0.5[25]. This parameter can be used to adjust the attenuation of noise and also the robustness of the closed loop. The filter can also be considered as a filter in cascade with the PID controller to give a proper transfer function

$$C(s) = \frac{K_p\left(1 + \frac{1}{T_i s} + T_d s\right)}{\alpha T_d s + 1} \tag{3.48}$$

For the series form the transfer function is

$$C(s) = K_p \frac{(1 + T_i s)}{T_i s} \frac{T_d s + 1}{\alpha T_d s + 1} \tag{3.49}$$

and for the parallel form

$$C(s) = K_p + \frac{K_i}{s} + \frac{K_d s}{\alpha K_d s + 1} \tag{3.50}$$

3.2.2a The Prediction approach

As explained in chapter 2, Smith predictor eliminates the effect of the delay on the closed-loop system by shifting the delay outside the feedback loop. Therefor after using the smith predictor, the PID tuning rule can be applied to the closed-loop system which has become delay-free system.

Consider a process model described by $G(s) = G_0(s)e^{-sh}$, where $G_0(s) = \frac{K_p}{1+Ts}$ and the primary controller as

$$C(s) = \frac{K_1(1 + T_1 s)}{T_1 s} \tag{3.51}$$

which is enough to fit the closed-loop behavior of the delay-free process. The PI can be tuned by cancelling the open-loop model pole ($T_1 = T$).

The characteristic equation of closed loop transfer function of Smith predictor (eq. 2.5) is given by $1 + C(s)G_0(s) = 1 + \frac{K_1 K_p}{Ts} = 1 + \frac{1}{T_0 s}$, wher $T_0 = \frac{T}{K_1 K_p}$

and the closed-loop transfer function is

$$\frac{y(s)}{r(s)} = \frac{C(s)G_0(s)}{1 + C(s)G_0(s)} e^{-sh} = \frac{1}{1 + T_0 s} e^{-sh} \tag{3.52}$$

T_0 can be used as a tuning parameter to define the closed-loop performance.

The equivalent controller is given by

$$C_e(s) = \frac{\frac{K_1(1 + Ts)}{Ts}}{1 + \frac{K_1 K_p}{Ts}(1 - e^{-sh})} = \frac{K_1(1 + Ts)}{Ts + K_1 K_p(1 - e^{-sh})} \tag{3.53}$$

This controller has integral action (note that s = 0 is a root of the denominator $Ts + K_1 K_p(1 - e^{-sh})$ and can be approximated by a PID if the dead time is substituted by a polynomial approximation.

3.2.2b PID approximation

If a $P_{11}(s)$ Pad’e approximation of dead time is used in Equation (3.53), $P_{11}(s) = \frac{1-sh/2}{1+sh/2}$, it follows that

$$C_e(s) = \frac{K_1(1 + Ts)}{Ts + K_1 K_p\left(1 - \frac{1 - \frac{sh}{2}}{1 + \frac{sh}{2}}\right)} = \frac{K_1(1 + Ts)}{Ts + K_1 K_p \frac{sh}{1 + \frac{sh}{2}}} \tag{3.54}$$

$$C_e(s) = \frac{K_1(1 + Ts)(1 + 0.5sh)}{Ts(1 + 0.5sh + \frac{K_1 K_p h}{T})} = \frac{K_1(1 + Ts)(1 + 0.5sh)}{Ts\left(1 + \frac{0.5sh}{1 + \frac{K_1 K_p h}{T}}\right)(1 + \frac{K_1 K_p h}{T})} \tag{3.55}$$

$C_e(s)$ can be made equal to a series PID controller with a filter in the derivative action

$$C_e(s) = k_p \frac{(1 + T_i s)}{T_i s} \frac{T_d s + 1}{\alpha T_d s + 1} \tag{3.56}$$

Where $T_i = T, T_d = 0.5h \ and$

$$\alpha = \frac{1}{1 + \frac{K_1 K_p h}{T}} = \frac{1}{1 + \frac{h}{T_0}}, \qquad k_p = \frac{T}{(h + T_0)K_p}$$

CHAPTER 4

Simulation Results

In this chapter, we demonstrate different methods to deal with delay in control systems. First, continuous systems have been considered with the two main approaches which are usually used to model the delay, Pad'e approximation method and Smith predictor method. In each case and after the delay has been accounted for, we design the control of the system by using PID and pole-placement control.
Second, we consider digital system case, where the delay is modeled directly to a rational function using modified *z*-transform. Modified *z*-transform is used to consider the cases when the delay is non integer multiple of the sampling time.

4.1 System design and implementation

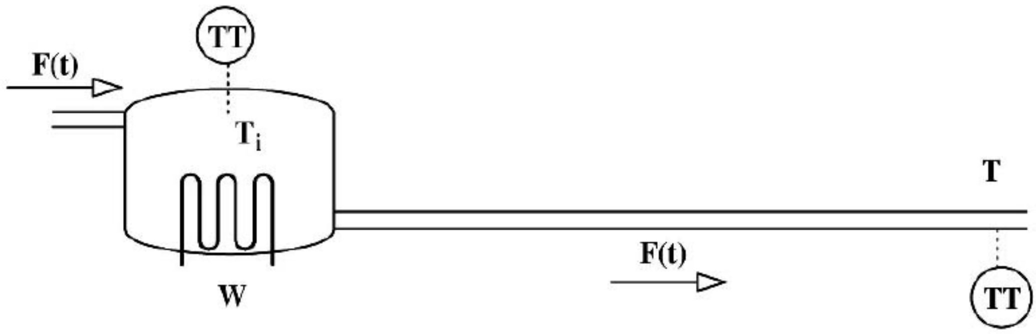


Fig. 4.1 A heated tank and along pipe

A water heater system shown in figure 4.1 will be studied as a case study; the water is heated in the tank using an electric resistor and driven by a pump along a thermally insulated pipe to the output of the system. The control input is the power *W* at the resistor and the plant output is the temperature *T* at the end of the pipe.

4.1.1 Continuous System

A linear model of the process $\mathrm{G(s)}$ can be obtained close to an operation point $\mathrm{W_0, T_0}$.

$$\mathrm{G(s)} = \frac{1}{(1 + 1.5\ \mathrm{s})(1 + 0.4\mathrm{s})} e^{-hs} \tag{4.1}$$

When a positive step is applied at W, the temperature inside the tank starts to increase. As the pipe is full of water at the initial temperature $\mathrm{T_0}$, this change is not immediately perceived at the output and it is necessary to wait until the hot water reaches the end of the pipe before it is noticed. Thus, after a delay *h*, defined by the flow and the length of the pipe, the output temperature *T* starts to rise with the same dynamics as the temperature inside the tank.

When a constant flow of water *F* is used, the delay *h* can be estimated using *F* and the volume of the pipe *V* as $\mathrm{h} = \frac{\mathrm{V}}{\mathrm{F}}$

4.1.2 Digital System

Consider that the dynamic behavior of the process is described by the continuous transfer function $G(s) = G_0(s)e^{-sh}$, where G(s) is the delay-time-free part of the process and h is the effective delay time and a sampling period T_s. A discrete description of the process is given by $\mathrm{G(z) = Z\{B_0(s)G(s)\}}$, where G(z) is the discrete transfer function relating the *Z* transform of the sampled output of the process and the *Z* transform of the discrete input that passes through a zero-order hold block $\mathrm{B_0(s)}$. From chapter 2,

1- If the delay is integer multiple of Ts, that is, an integer Δ exists such that $\mathrm{h = \Delta\, T_s}$, then $\mathrm{G(z) = G_0(z)\, z^{-\Delta}}$.

2- If the delay is non-integer multiple of Ts, such that $h = \Delta\, T_s + \delta h$, then $G(s) = G_0(s)e^{-sh} = G_0(s)A(s)e^{-\Delta\, T_s\, s}$, where A(s) is the rational function used to approximate $e^{-\delta h s}$. In this case the complete model is given by

$$G(z) = G_0(z)\, \mathrm{z}^{-\Delta}, \qquad \mathrm{G_0(z) = Z\{B_0(s)G(s)A(s)\}} \tag{4.2}$$

Example 1: For a continues model G(s) in equation 4.1, using $T_s = 0.1\, s$ and $h = 0.2\, s$, the digital model of *G(s)* is given by

$$G(z) = \frac{0.007509z + 0.006757}{z^4 - 1.714z^3 + 0.7286z^2}$$

Example 2: For a continues model G(s) in equation 4.1, using $T_s = 0.1\, s$ and $h = 0.25\, s$, thus writing $h = \Delta\, T_s + \delta h = 0.1\Delta + \delta h$, gives Δ= 2 and $\delta h = 0.05$.

Approximating $e^{-\delta hs} = e^{-0.05s}$ by $\frac{1}{1+0.05s}$, the model can be written as

$$G(s) \cong \frac{1}{(1+1.5\,s)(1+0.4s)(1+0.05s)} e^{-0.2s},$$

the digital model of *G(s)* is given by,

$$G(z) = \frac{0.00331z^2 + 0.007971\,z + 0.001054}{z^5 - 1.85z^4 + 0.9606z^3 - 0.0986z^2}$$

4.2 Delay modeling in continuous systems

4.2.1 Pad'e approximation

In Pad'e approximation, the delay is modeled as a rational function (refer to chapter 2). In our system, we have modeled the delay as

$$e^{-sh} \cong \frac{1-\frac{h}{2}s}{1+\frac{h}{2}s} \qquad (4.3)$$

Where h is the delay.

4.2.1a PID Control method

First, we design the system using PID Control. Optimized-tuning has been considered to realize the proportional, the integral and the derivative gains for a given system. The constraints which has been used are

- The overshoot is to be less than 5%,
- The undershoot is to be less than 20%
- The settling time is to be less than 10 s,
- The rising time is to be less than 4 s,
- The steady –state error is to be less than 2 %

To achieve the gains that meet the constraint, Gradient descent as the optimization algorithm has been utilized to optimize the response signal subject to the constraints. This algorithm calculates gradients based on the refined method

The achieved responses and corresponding gains for different delays are given in Table 4.1. The resulted system response is also depicted in Figure 4.2.

4.2.1b Pole-placement method

For the same system specification given in the previous section, algebraic Riccati equation (ARE) approach is used to achieve optimal state feedback gain. For the given system specifications, the poles should be within a specified region. The region is given as a circle with radius r in the left-half plane. The centre of the circle has coordinate $(-(r+\gamma),0)$. The system matrix A is modified to achieve this constraint before it is passed to ARE. This ARE is given in chapter 3 as,

$$\frac{(\mathrm{A}-\alpha\mathrm{I})^{\mathrm{T}}}{\mathrm{r}}\mathrm{P}\frac{(\mathrm{A}-\alpha\mathrm{I})}{\mathrm{r}}-\mathrm{P}-\frac{(\mathrm{A}-\alpha\mathrm{I})^{\mathrm{T}}}{\mathrm{r}}\mathrm{PB}(\mathrm{r}^2\mathrm{R}+\mathrm{B}^{\mathrm{T}}\mathrm{PB})^{-1}\mathrm{B}^{\mathrm{T}}\mathrm{P}\frac{(\mathrm{A}-\alpha\mathrm{I})}{\mathrm{r}}=\mathrm{Q}$$

where P is a positive definite symmetric solution of the Riccati equation, and the state feedback law to assign all the closed-loop poles of system in the disk D is,

$\mathrm{u}=(\mathrm{r}^2\mathrm{R}+\mathrm{B}^{\mathrm{T}}\mathrm{PB})^{-1}\mathrm{B}^{\mathrm{T}}\mathrm{P}(\mathrm{A}-\alpha\mathrm{I})\mathrm{x}=\mathrm{Kx}$, Wher K is the optimal gain of system.

If the system has no delay, then the solution of algebraic Riccati equation is

$$P=\begin{bmatrix}0.2110 & 0.6904\\ 0.6904 & 2.6834\end{bmatrix};$$

and the corresponding gain is K=[0.2110   0.6904].

If the delay h= 3.5s, , then the solution of algebraic Riccati equation is

$$P=\begin{bmatrix}2.7008 & 11.0421 & 10.7549\\ 11.0421 & 47.6483 & 50.3118\\ 10.7549 & 50.3118 & 67.6228\end{bmatrix};$$

and the corresponding gain is $K=[2.7008\ \ 11.0421\ \ 10.7549]$.

The achieved responses and corresponding gains for different delays are given in Table 4.2. The resulted system responses are also depicted in Figure 4.3.

Table 4.1: Achieved responses and corresponding gains for Pad'e modeling using PID controller

Delay h "sec"	Rising time	Sett. Time	Ess	O.S.	U.S.	Gains
0	0.164	0.44	0.0050	0.015	0	Kp=15.76; Ki=10.35; Kd=5.063;
1	2.4	8	0.01	0	0.05	Kp=.8066; Ki=.3979; Kd=.0806;
2	3.98	12	0.001	0.018	0.076	Kp=0.4726; Ki=0.2383; Kd=0.0268;
3.5	5.67	17	0.0001	0.01	0.11	Kp=0.3907; Ki=0.1589; Kd=0.0248;
5	6.42	20	0.0001	0.01	0.15	Kp=0.3944; Ki=0.1271; Kd=0.0088;

It is clear from Table 4.1 and as accepted, the system without delay ($h = 0$) gives the best system response. When the delay is greater than $2s$, the required rising and settling time cannot be realized.

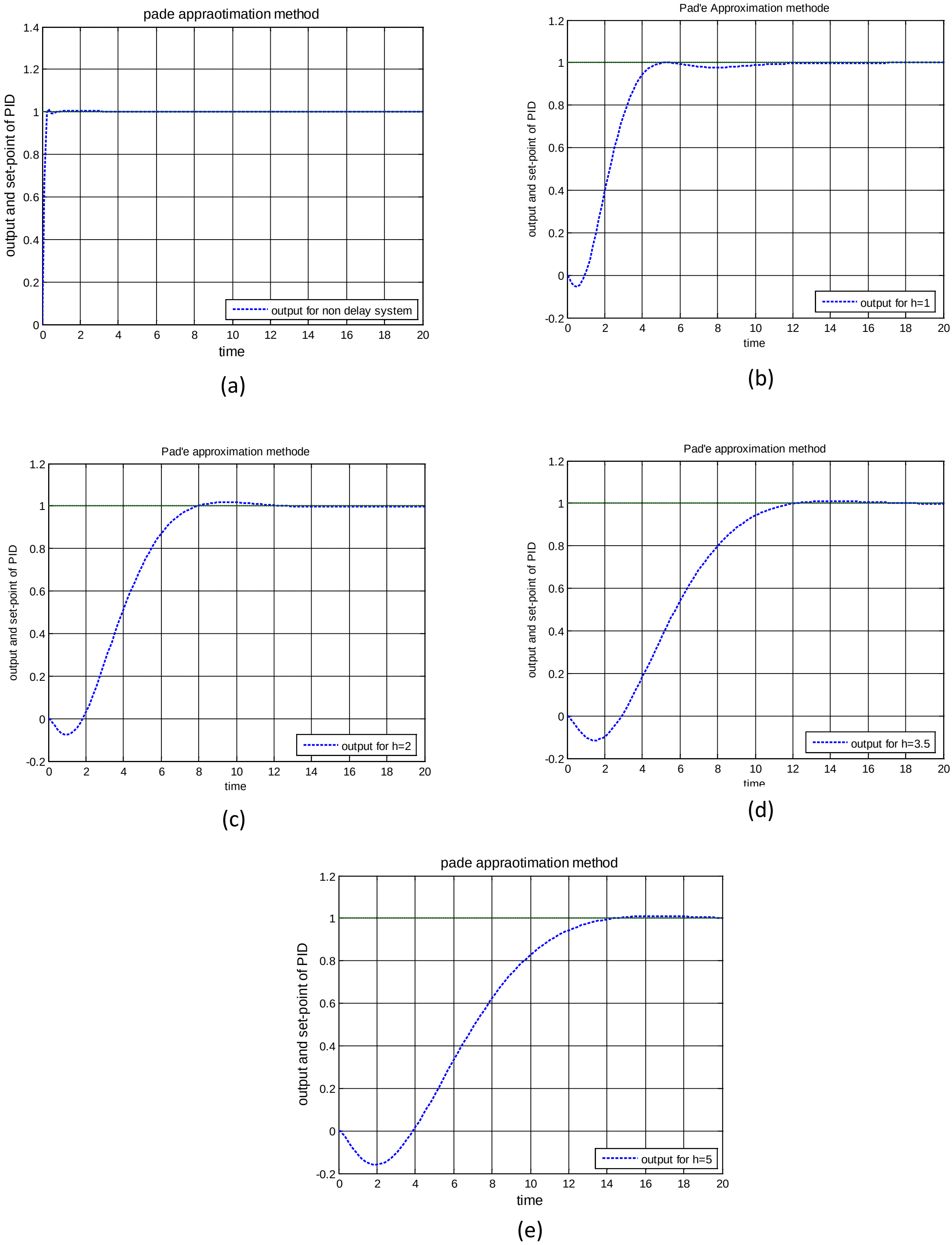
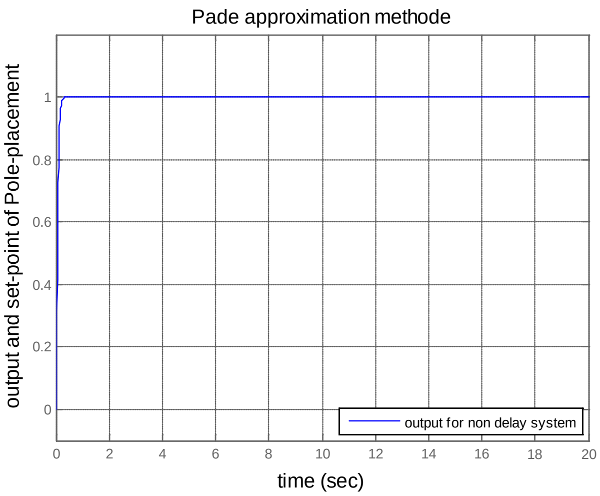


Figure 4.2: System responses for different delays for Pad'e modeling using PID controller (a) Non delayed system (b) Delay *h* = 1s, (c) Delay *h* = 2s, (d) Delay *h* = 3.5s, (e) Delay *h* =5s

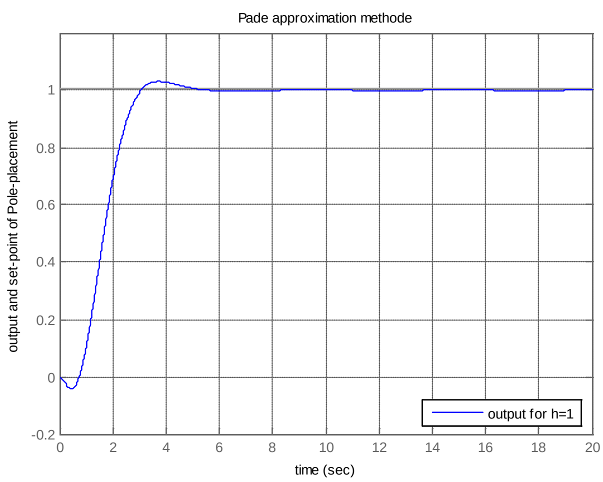
Table 4.2: Achieved responses and corresponding gains for pad'e modeling using Pole-placement method

Delay h "sec"	Rising time	Sett. Time	Ess	O.S.	U.S.	Closed loop poles	Gains
0	0.118	0.36	0	0	0	l = -2.3925 -0.9852	k = 0.2110 0.6904 Alpha=30
1	1.55	5.16	0	0.031	0.041	l = -1.9315 -1.0993 + 1.1803i -1.0993 - 1.1803i	k = 1.9634 10.1081 13.6696 Alpha=1
2	1.7	6	0	0.03	0.071	l = -1.6546 -0.9556 + 1.0495i -0.9556 - 1.0495i	k = 2.3991 10.4751 11.4093 Alpha=1
3.5	1.95	7.8	0	0.019	.09	l = -1.5267 -0.9561 + 0.8984i -0.9561 - 0.8984i	k = 2.7008 11.0421 10.7549 Alpha=1
5	2.81	9.8	0	0.008	0.16	l = -1.8347 -0.7594 + 0.4909i -0.7594 - 0.4909i	k = 1.2870 4.7748 3.5994 Alpha=.5

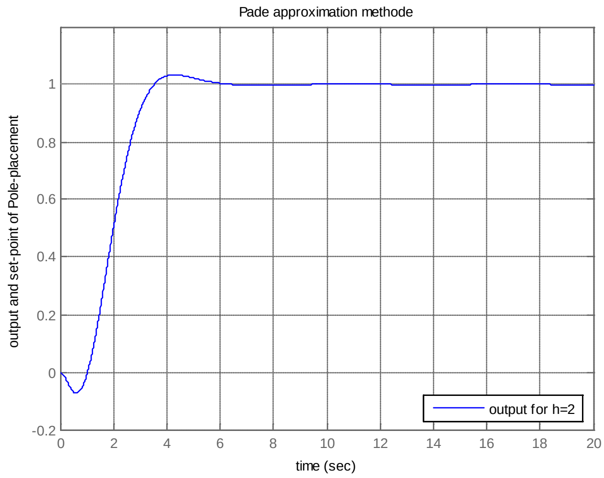
It is clear from the Table 4.2 and as accepted, the system without delay ($h = 0$) gives the best system response. In case of using pole-placement with Pad'e approximation method, we can see that until the delay h =5, we still get the requested system parameters. At $h > 5$ the settling time starts to diverge from the required value.



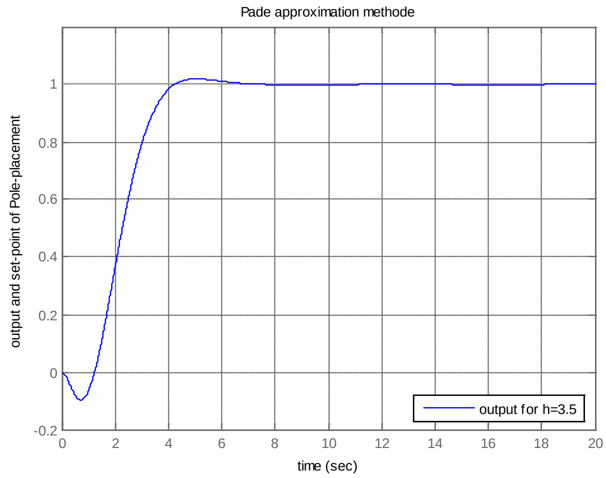
(a)



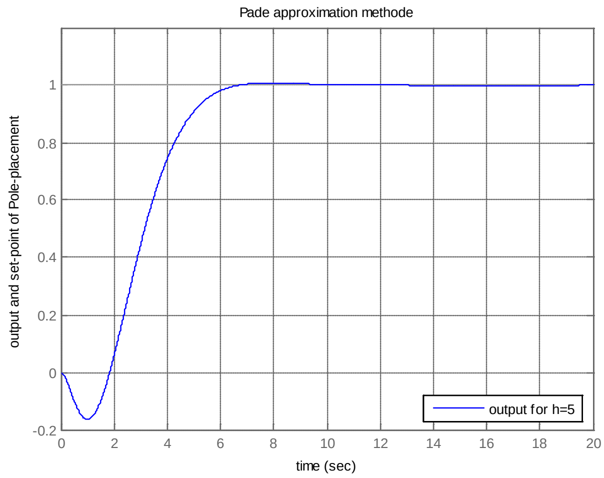
(b)



(c)



(d)



(e)

Figure 4.3: System responses for different delays for Pad'e modeling using pole-placement method (a) Non delayed system, (b) Delay *h* = 1s, (c) Delay *h* = 2s, (d) Delay *h* = 3.5s, (e) Delay *h* =5s

4.2.1c Comparison between PID and pole-placement methods

By observing the results in Tables 4.1 and 4.2, it can be noticed that the pole-placement method is much better than PID method. The two factors which are mainly improved by using ARE over PID are the settling time and the rising time. For example, when the delay *h* equals 3.5 s, the settling time and rising time are 7.8 s, 1.95 s, respectively in pole placement method. While the settling time is 17 s and the rising time equal 5.67 s in case of PID method. The ARE and PID methods give close results for the other system parameters (undershoot, overshoot and the steady state error).

4.2.2 Smith Predictor

In Smith predictor method is built around moving the delay outside the feedback loop and the design process become the same as a delay-free system. However, very high controller gain cannot be used as the delay still puts fundamental limitations on the achievable bandwidth**.** The delay still has an effect because there is a difference between the real system and the nominal model. Smith predictor assumes a perfect nominal model of the system which -unfortunately -cannot always be realized.

4.2.2a PID method

The same specifications in section 4.1.1a have been used in this section. The achieved responses and corresponding gains for different delays are given in Table 4.3. The resulted system responses are also depicted in Figure 4.4

Table 4.3: Achieved responses and corresponding gains for Smith predictor modeling using PID controller

Delay h "sec"	Rising time	Sett. Time	Ess	O.S.	U.S.	Gains
0	0.164	0.44	0.0050	0.015	0	Kp=15.76; Ki=10.35; Kd=5.063;
1	0.82	5	0.001	0.049	0	Kp=3.423; Ki=1.651; Kd=.1;
2	0.728	5.5	0.0001	0.031	0	Kp=4.526; Ki=2.171; Kd=0.525;
3.5	1.415	8	0	0.028	0	Kp=1.65; Ki=1.111; Kd=0.01902;
5	0.678	10	0.0001	0.02	0	Kp=4.885; Ki=2.278; Kd=0.6856;

From table 4.3, we can see again that the system without delay (h =0) gives the best system response. In case of using PID controller with Smith predictor to model the delay, we can see that until the delay h =5, we still get the specified system parameters. At h > 5 the settling time starts to diverge from the required value.

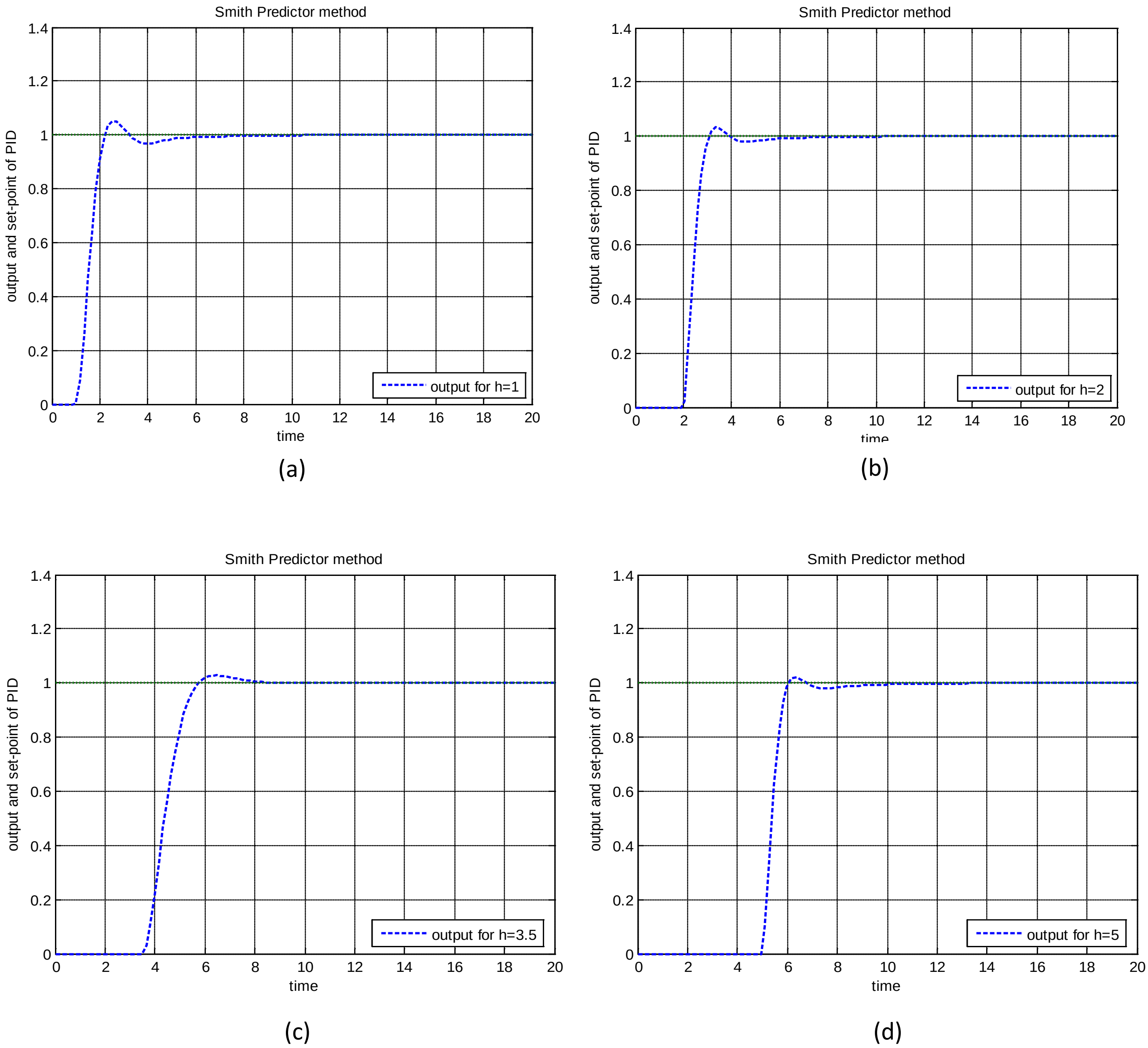


Figure 4.4: System responses for different delays for Smith Predictor modeling usingPID method (a) Delay *h* = 1s, (b) Delay *h* = 2s, (c) Delay *h* = 3.5s, (d) Delay *h* =5s

4.2.2b Pole-placement method

The same specifications in section 4.1.1a have been used in this section.

If the system has no delay, then the solution of algebraic Riccati equation is

$$P = \begin{bmatrix} 0.2110 & 0.6904 \\ 0.6904 & 2.6834 \end{bmatrix};$$

and the corresponding gain is K=[0.2110   0.6904].

If the delay h= 3.5s, then the solution of algebraic Riccati equation is

$$P = \begin{bmatrix} 0.2110 & 0.6904 \\ 0.6904 & 2.6834 \end{bmatrix};$$

and the corresponding gain is $K = [0.2110 \quad 0.6904]$.

The achieved responses and corresponding gains for different delays are given in Table 4.4. The resulted system responses are also depicted in Figure 4.5.

Table 4.4: Achieved responses and corresponding gains for Smith predictor modeling using Pole-placement method

Delay h "sec"	Rising time	Sett. Time	Ess	O.S.	U.S.	Closed loop poles	Gains
0	0.118	0.36	0	0	0	l = -2.3925 -0.9852	k = 0.2110 0.6904
1	0.12	1.3	0	0	0	l = -2.3925 -0.9852	k = 0.2110 0.6904
2	0.12	2.3	0	0	0	l = -2.3925 -0.9852	k = 0.2110 0.6904
3.5	0.12	3.8	0	0	0	l = -2.3925 -0.9852	k = 0.2110 0.6904
5	0.12	5.3	0	0	0	l = -2.3925 -0.9852	k = 0.2110 0.6904
10	0.12	10.2	0	0	0	l = -2.3925 -0.9852	k = 0.2110 0.6904

From table 4.4, we can see again that the system without delay (*h* =0) gives the best system response. In case of using pole-placement method with Smith predictor to model the delay, we can see that until the delay *h =5*, we still get the specified system parameters. At *h > 10* the settling time starts to diverge from the required value. We notice also that by using this method the rising time stay constant regardless of the delay value.

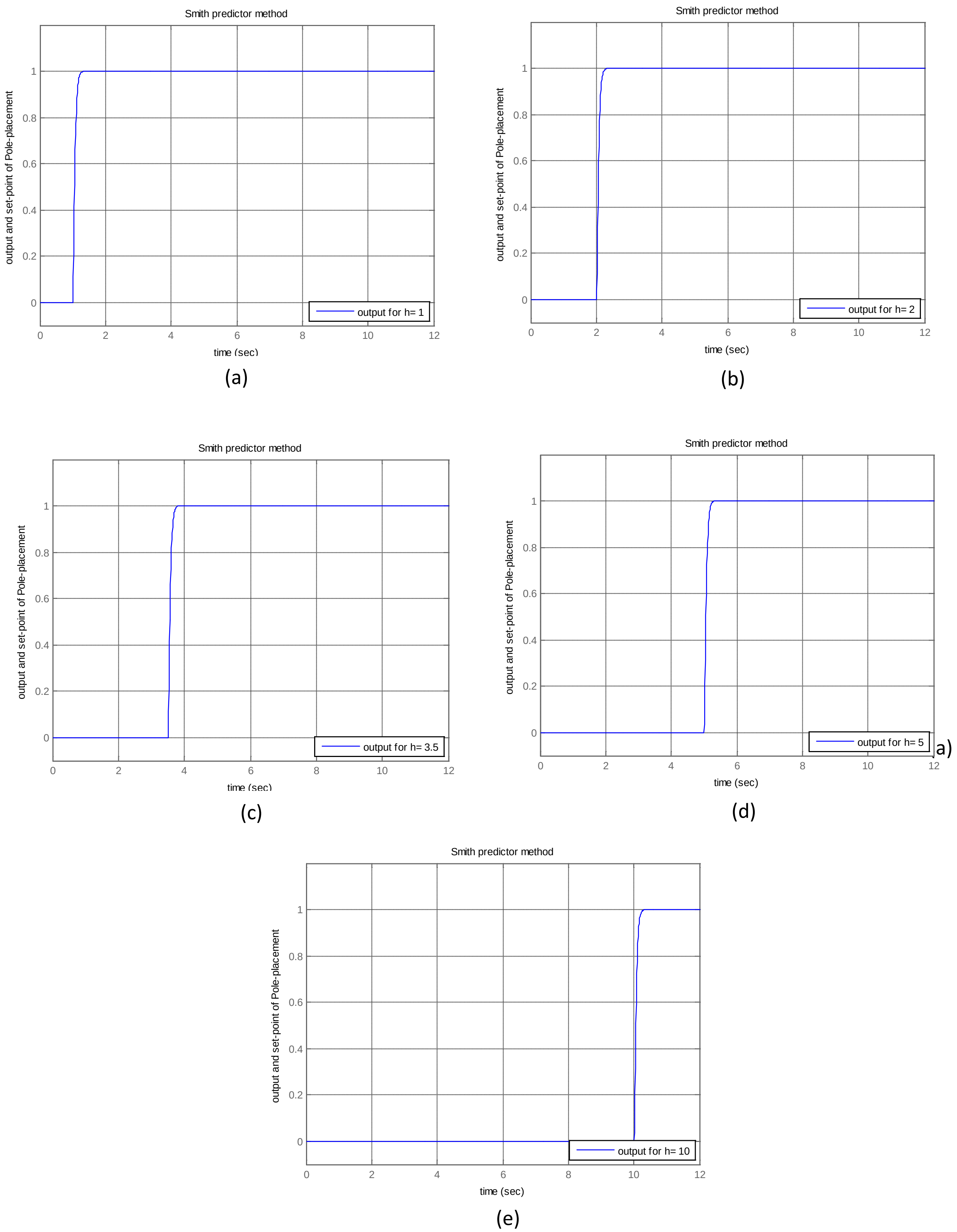


Figure 4.5: System responses for different delays for Smith Predictor modeling using pole-placement method (a) Delay *h* = 1s, (b) Delay *h* = 2s, (c) Delay *h* = 3.5s, (d) Delay *h* =5s, (e) Delay *h* =10s

4.2.2c Comparison between PID and pole-placement methods (Smith case)

By observing the results in Table 4.3 and Table 4.4, it is very clear that pole-placement method (ARE approach) gives much better system parameters than the PID method. For example, when the delay h equals to 3.5 s, the system parameters are given by o.s. = 0, u.s. = 0, ts = 3.8 s, tr = 0.12 s, ess = 0.0 in case of ARE approach; whereas in case of PID, the system parameters are given by o.s. = 0.028, u.s. = 0, ts = 8 s, tr = 1.415 s, ess = 0.0.

4.2.3 Comparison between Pad'e and Smith methods

In PID case, the Smith method generally gives better results than the Pade' method. For example, when the delay *h* equals 2 s, the system parameters are u.s. = 0, ts = 5.5 s, tr = 0.728 s, ess = 0.0001 in case of smith method. whereas in case of Pad'e, the system parameters are given by u.s. = 0.076, ts = 12 s, tr = 3.98 s and ess = 0.001. However, Pade' method gives slightly better overshoot o.s. = 0.018 compare to o.s. = 0.031 in case of Smith.

In ARE case, the Smith method gives also better system parameters than the Pade' method. For example, when the delay *h* equals 2 s, the system parameters are o.s = 0, u.s. = 0, ts = 2.3 s, tr = 0.12 s, ess = 0.0 in case of smith method. Whereas in case of Pade, the system parameters are given by o.s. = 0.03, u.s. = 0.071, ts = 6 s, tr = 1.7 s and ess = 0.0.

4.3 Delay modeling in Digital Simulation using Modified z-transform

The delay in digital system is modeled as a power of *z*. Unlike the continuous system, the delay can be considered as part of the closed loop transfer function. Nevertheless, it increases the order of the system significantly.

As it has been discussed in chapter 2, non-integer index of *z* is used in the *z*-transform to account for non integer multiple of the sampling time when the delay is considered.

4.3a PID method

The same specifications given in section 4.1.1a are also used in this section. The achieved responses and corresponding gains for different delays are given in Table 4.5. The resulted system responses are also depicted in Figure 4.6.

Table 4.5: Achieved responses and corresponding gains for Modified *z* modeling using PID controller

Delay h "sec"	Rising time	Sett. Time	Ess	O.S.	U.S.	Gains
0	0.245	0.8	0.003	0.05	0	Kp=9.716; Ki=4.884; Kd=2.446;
0.1	0.649	1.375	0.012	0.002	0	Kp=3.14; Ki=1.54; Kd=.666;
0.2	0.708	2.86	0.005	0.022	0	Kp=2.871; Ki=1.411; Kd=.686;
0.25	0.811	3.3	0.011	0.012	0	Kp=2.745; Ki=1.511; Kd=.8818;
0.5	1	3.317	0.006	0.005	0	Kp=1.723; Ki=.8841; Kd=.6059;
1.7	1.9	9.5	0.01	0.034	0	Kp=.8998; Ki=.3487; Kd=.4801;

From Table 4.5, we can see again that the system without delay (h =0) gives the best system response. In case of using PID controller with modified z- transform to model the delay, we can see that until the delay equal five times the sampling times h =5Ts, we still get the specified system parameters. At h > 17Ts, the settling time starts to diverge from the required value.

4.3b Pole-placement method

The same specifications given in section 4.1.1a are also used in this section.

If the system has no delay, then the solution of algebraic Riccati equation is

$$P = 1.0 * 10^4 \begin{bmatrix} 0.0173 & 0.1426 \\ 0.1426 & 1.3585 \end{bmatrix};$$

and the corresponding gain is K=[46.7275 238.0405].

If the delay h= 0.5s, , then the solution of algebraic Riccati equation is

$$P = \begin{matrix} 110.8 & -291.7 & 316.3 & -181.6 & 58.60 & -10.09 & 0.7255 \\ -291.7 & 779.8 & -858.4 & 500.3 & -163.6 & 28.56 & -2.079 \\ 316.3 & -858.4 & 960.2 & -568.8 & 189.06 & -33.52 & 2.4801 \\ -181.6 & 500.3 & -568.8 & 342.8 & -116.1 & 20.97 & -1.582 \\ 58.60 & -163.6 & 189.06 & -116.1 & 40.09 & -7.414 & 0.574 \end{matrix}$$

-10.09	28.56	-33.52	20.97	-7.414	1.410	-0.113
0.7255	-2.079	2.4801	-1.582	0.574	-0.113	0.0096

and the corresponding gain is,

$K = [-4.1782 \quad 10.1173 \quad -8.9555 \quad 3.8941 \quad -1.0154 \quad 0.1472 \quad -0.0092]$.

The achieved responses and corresponding gains for different delays are given in Table 4.6. The resulted system responses are also depicted in Figure 4.7.

Table 4.6: Achieved responses and corresponding gains for Modified z modeling using Pole-placement method

Delay h "sec"	Rising time	Sett. Time	Ess	O.S.	Closed loop poles	Gains
0	0.03	0.08	0.002	0.002	l = 0.3984 0.3066	K = 46.7275 238.0405
0.1	0.326	1.8	0.0024	0.001	l = -0.7129 0.7221 + 0.1238i 0.7221 - 0.1238i	K = 0.2636 0.2456 -0.3381
0.2	0.5	2	0.004	0.013	l = -0.6888 + 0.0861i -0.6888 - 0.0861i 0.6972 + 0.1468i 0.6972 - 0.1468i	K = -0.4218 1.3887 -0.9190 0.0815
0.25	0.6	2.5	0.0013	0.028	l = 0.6825 + 0.1582i 0.6825 - 0.1582i -0.7051 + 0.2073i -0.7051 - 0.2073i -0.6281	K = -0.7218 2.0273 -1.4974 0.3114 -0.0210
0.5	0.7	2.6	0.001	0.014	l = 0.6012 + 0.1910i 0.6012 - 0.1910i -0.2816 + 0.4970i -0.2816 - 0.4970i -0.3497 + 0.1744i -0.3497 - 0.1744i -0.3667	K = -4.1782 10.1173 -8.9555 3.8941 -1.0154 0.1472 -0.0092
0.7	0.9	2.8	0.002	0.05	l = 0.5318 + 0.1918i 0.5318 - 0.1918i -0.0419 + 0.4993i -0.0419 - 0.4993i -0.2220 + 0.2600i -0.2220 - 0.2600i -0.2789 + 0.1097i -0.2789 - 0.1097i -0.3003	K = -7.0827 19.8725 -23.1447 15.5883 -6.9975 2.0938 -0.4028 0.0452 -0.0023

From Table 4.6, It can be seen again that the system without delay (*h* =0) gives the best system response. In case of using pole-placement method with modified z-transform to model the delay, we can see that until the delay equal five times the sampling times h =5Ts, we still get the specified system parameters. At h > 7TS the settling time starts to diverge from the required value.

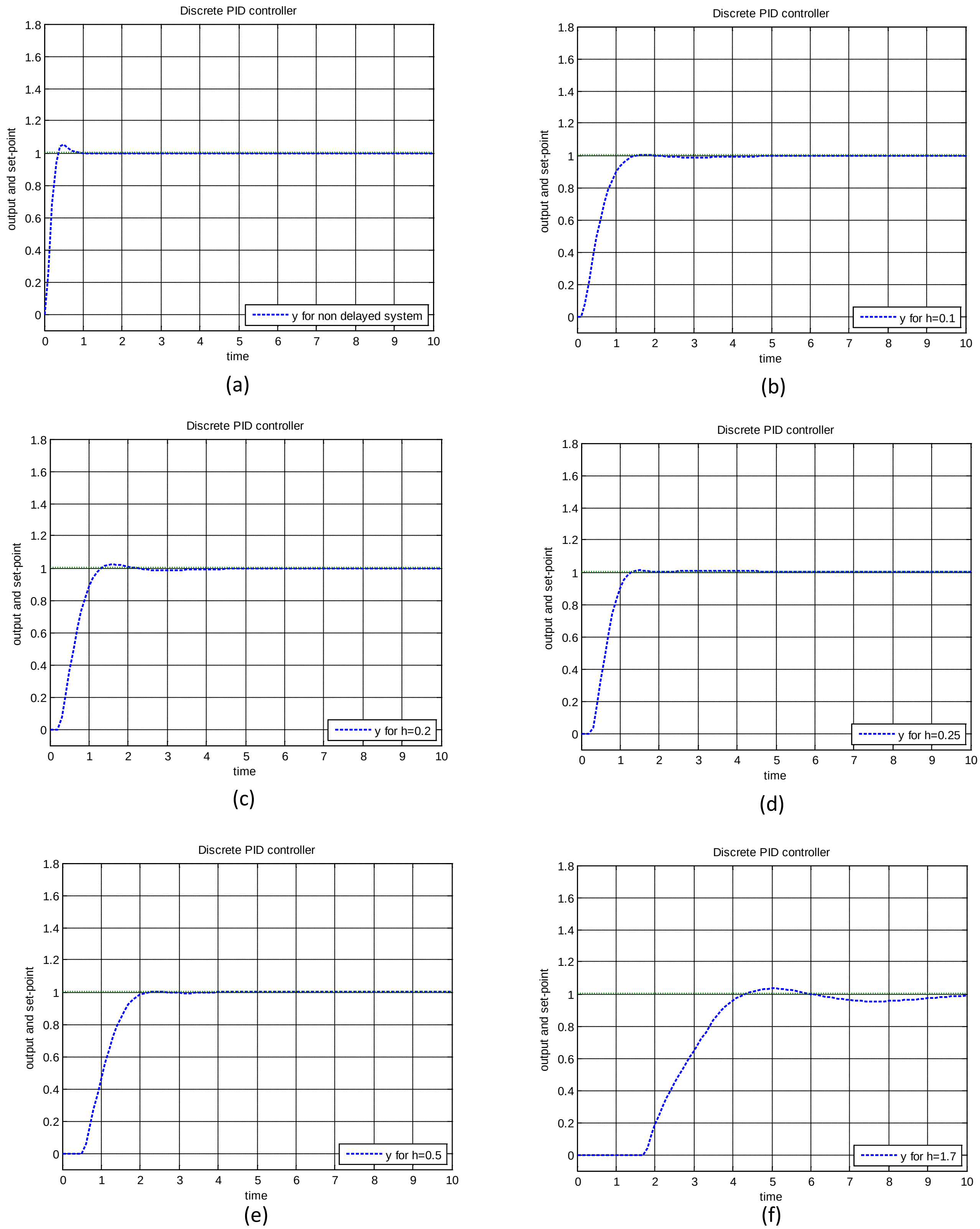


Figure 4.6: System responses for different delays for modified z modeling using PID controller (a) Non delayed system (b) Delay h = 0.1s, (c) Delay h = 0.2s, (d) Delay h = 0.25s, (e) Delay h = 0.5s, (f) Delay h = 1.7s

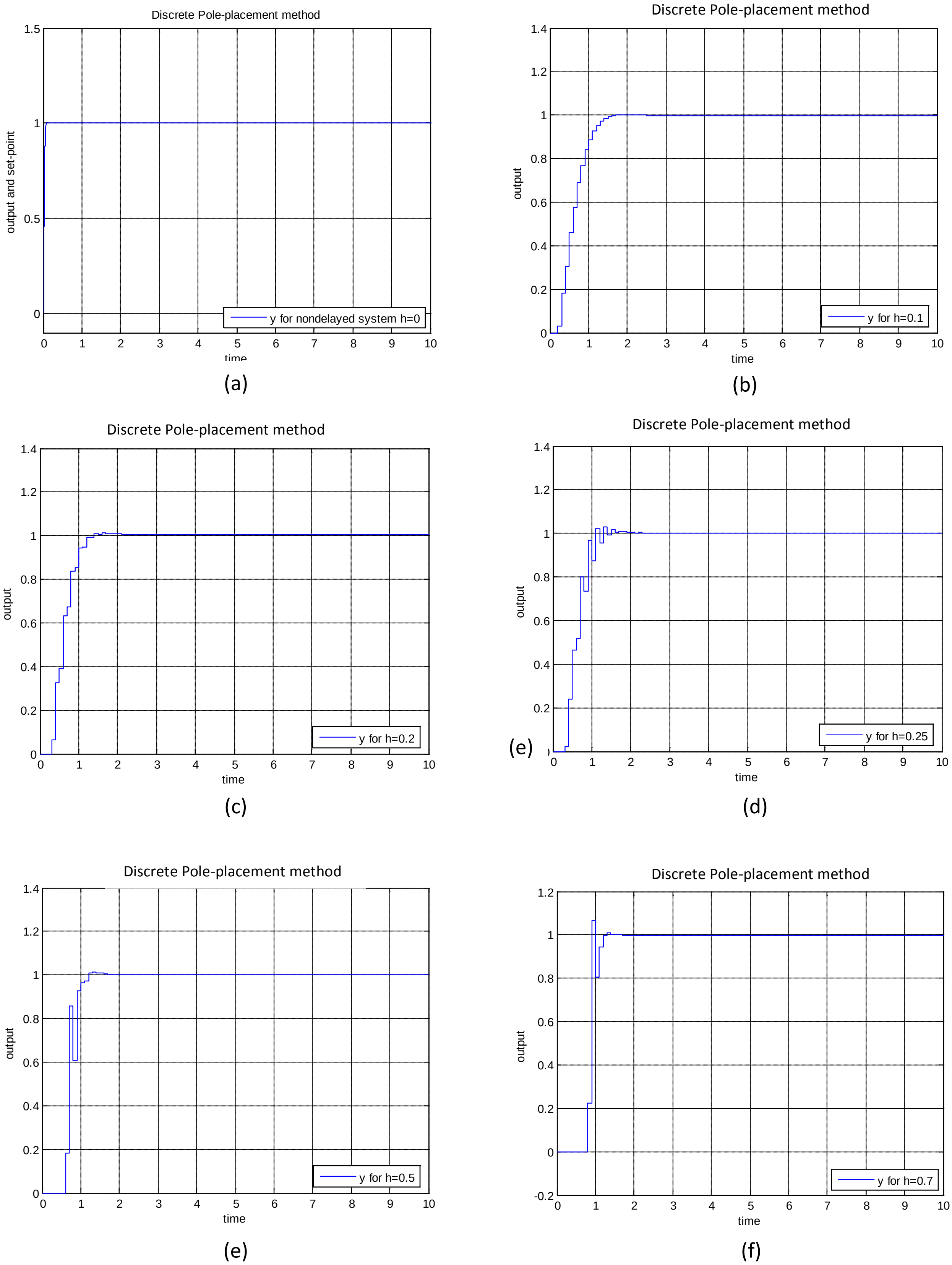


Figure 4.7: System responses for different delays for modified z modeling using pole-placement method (a) Non delayed system (b) Delay *h* = 0.1s, (c) Delay *h* = 0.2s, (d) Delay *h* = 0.25s, (e) Delay *h* = 0.5s, (f) Delay *h* = 0.7s

4.3c Comparison between PID and pole-placement methods

By observing the results in Tables 4.5 and 4.6, it can be noticed that the pole-placement method is generally better than PID method (if we exclude the overshoot which is slightly better in case of PID). For example, when the delay *h* equals 0.25 s, ts = 2.5 s, tr = 0.6 s, ess = 0.0013, o.s. = 0.028, in case of ARE method. While the ts = 3.3 s, tr = 0.81 s, ess = 0.011, o.s. = 0.012 in case of PID method. It is also observable that the digital system gives better results than the analogue one in all system parameters.

4.4 Summary of the outcomes

Different methods to compensate dead-time systems have been analysed and compared. We have found that the smith predictor method in continuous system gives better results than the Pad'e. It also has been found that the ARE approach gives better results than PID methods when either Smith predictor or pad'e approximation is used. Hence, it can be concluded that Smith predictor to model the delay combined with ARE to design the controller gives the best result.

In digital system, the ARE approach also gives better results than PID method. Digital system in general gives better system response for different value of the delay than the continuous one.

CHAPTER5

Conclusion and future work

Delay is inevitable in many electrical and mechanical systems. When such delayed systems are needed to be controlled, the design becomes a real challenge. In continuous-time systems, the delay is expressed as infinite dimension term which prevents applying different traditional design methods directly. In discreet system, though the delay can be expressed as a finite-dimension term, it increases the dimension and hence the complexity of the system significantly. From the previous argument, it is clear that there is a need to model the delay before starting design the controller of the system.

In this thesis, we analyzed and compared different methods for compensation of delayed systems. For continuous-time systems, the delay has been modeled by using Smith predictor and Pad'e approximation. In each case, the controller was designed using both ARE and PID approaches. We concluded that the smith predictor method in continuous system gave better results than the Pad'e. We also concluded that the ARE approach gave better results than PID methods when either Smith predictor or pad'e approximation was used. Hence, it can be concluded that Smith predictor for modeling the delay combined with ARE for designing the controller gave the best result.

In digital system, the delay has been compensated by using the modified z-transform. We found that the ARE approach also gave better results than PID method. Digital system in general gave better system response for different value of the delay than the continuous one.

In this study, constant delay was assumed throughout the thesis, where as in some practical system the delay can be variable or unknown. System with variable delay can be considered for future work. We considered internal delay only which is not expressed inside the state. System with state delay can be simulated and studied.

Smith predictor is used for stable plants only. If we want to consider unstable plants, modified Smith predictor can be used in future work.

Bibliography

[1] J. Silva Guillermo, Aniruddha Datta and S.R Bhattacharyya , *PID Controllers for Time-Delay Systems*, Birkhauser Boston, 2005.

[2] F.G. Shinskey, *Process Control Systems*, McGraw-Hill Book Company,1967.

[3] Qing-Chang Zhong, *Robust Control of Time-delay Systems*, Springer-Verlag London Limited 2006

[4] Svante Bjorklund, "A Survey and Comparison of Time-Delay Estimation Methods in Linear Systems," Linkoping Studies in Science and Technology Thesis No. 1061, Linkoping 2003

[5] L. Mirkin and Z. J. Palmor, "Control Issues in Systems with Loop Delays," the Isreal Science Foundation (grant No. 298/04) and the Israel Electric Company (grant No. 793-2000-3).

[6] Hiroshi Shinozaki," Lambert W Function Approach to Stability and Stabilization Problems for Linear Time-Delay Systems," Kyoto Institute of Technology, Thesis, December 2007

[7] J.P. Richard," Time-delay Systems: An Overview of Some Recent Advances and Open Problems," Automatica, 39:1667–1694, 2003.

[8] T. Furukawa and E. Shimemura, "Predictive Control for Systems with Time Delay," International Journal of Control, Vol. 37, No. 2, pp. 399–412, 1983.

[9] N. Abe and E. Nobuyama, "Control of Delay Systems from Fundamentals to Frontiers #1: Introduction of Time Delay Systems-Transfer Function Approach," Journal of the Society of Instrument and Control engineers, Vol. 44, No. 11, pp. 799–804, 2005, (in Japanese).

[10] E. Nobuyama and N. Abe, "Control of Delay Systems from Fundamentals to Frontiers #2: Introduction of Time Delay Systems-State Space Approach," Journal of the Society of Instrument and Control engineers, Vol. 44, No. 12, pp. 883–887, 2005,(in Japanese).

[11] Costas Kravaris, Raymond A. Wright,” Deadtime Compensation for Nonlinear Processes,” AIChE Journal, 1989 Vol. 35, No.9

[12] Huang, H.-P., Chen, C.-L., Chao, Y.-C. and Chen P.-L,” A Modified Smith Predictor with an Approximate Inverse of Dead Time,” AIChE Journal, 36, 1025-1031, 1990.

[13] J. J. Hench, C. He, V. Kuˇcera, and V. Mehrmann ,” Dampening Controllers via a Riccati Equation Approach,” IEEE TRANSACTIONS ON AUTOMATIC CONTROL, VOL. 43, NO. 9, 1998.

[14] S.I. Niculescu and Erik I. Verriest, ”Delay-independent Stability of Linear Neutral Systems: A Riccati Equation Approach,” Stability and Control of Time-delay System, vol. 228, pp. 92-100, 1998.

[15] Syder J, T Heeg, and A O’Dwyer, “ Dead-time Compensators: Performance and Robustness Issues,” European Journal of Control, 2000.

[16] N. Abe and K.Yamanaka ,” Smith Predictor Control and Internal Model Control - A Tutorial,” SICE Annual Conference in Fukui, 2003

[17] K.J. Astrom, ”Limitations on control system performance,” European Journal of Control, 6(1):1–19, 2000.

[18] C. Frantisˇek, “Modelling and Control of Discrete Event Dynamic Systems,” BRICS Report Series, ISSN 0909-0878, October 2000

[19] Brian Roffel and Ben H. Betlem, *Advanced Practical Process Control*, springer, November 2003

[20] Bittanti, Laub, and Willems, *The Riccati Equation*, Spring Verlag, 1991.

[21] K. Zhou and J.C. Doyle, *Essentials of Robust Control*, Prentice-Hall, Upper Saddle River, NJ, 1998.

[22] M. Gopal, *Modern Control System Theory*, New Age International (p) Ltd., Puplisher 2006.

[23] K. Furuta and S.B. Kim, "Pole Assignment in a Specified Disk," IEEE Transactions on Automatic control VOL. AC-32. NO. 5, MAY 1987

[24] Pradeep Misra, "LQR Design with Prescribed damping and Degree of Stability," Proceedings of the 1996 IEEE International Symposium

[25] J.E. Normey-Rico and E.F. Camacho, *Control of Dead-time Processes*, Springer -Verlag London Limited 2007

[26] K.H. Ang, G. Chong, and Y. Li, " PID Control System Analysis, Design, and Technology," IEEE Trans. Control Syst. Technology, 13(4):559–576, 2005.

[27] K.J. Astrom and T. Hagglund, *PID Controllers: Theory, Design, and Tuning*, Instrument Society of America, Research Triangle Park, NC, 2nd edition, 1995

[28] K.J. Astrom and T. Hagglund, "The Future of PID control," Chem. Eng. Progress, 9:1163–1175, 2001.

[29] K.J. Astrom and T. Hagglund, *Automatic Tuning of PID Controllers*, Instrument Society of America, Research Triangle Park, NC, 1988.

[30] P. Cominos and N. Munro, "PID Controllers: Recent Tuning Methods and Design to Specification," IEE Proc. Control Theory Appl., 149(1):46–53, 2002.

Appendices

Appendix A

MATLAB Methode

To solve ARE by MATLAB commaned, we use care and dare for solving continuous-time algebraic Riccati equations and discrete-time algebraic Riccati equations respectively.

CARE Command:

$[P, L, G] = \mathrm{care(A, B, Q)}$

$[P, L, G] = \mathrm{care(A, B, Q, R, S, E)}$

$[P, L, G, report] = \mathrm{care(A, B, Q, \ldots\ldots)}$

$[P_1, P_2, D, L] = \mathrm{care(A, B, Q, \ldots, 'factor')}$

Description

- $[P, L, G] = \mathrm{care(A, B, Q)}$ computes the unique solution P of the continuous-time algebraic Riccati equation

$$A^T P + PA - PBR^{-1}B^T P + Q = 0$$

The care function also returns the gain matrix $G = R^{-1}B^T PE$

$[P, L, G] = \mathrm{care(A, B, Q, R, S, E)}$ solves the more general Riccati equation

$$A^T PE + E^T PA - (E^T PB + S)R^{-1}(B^T PE + S^T) + Q = 0$$

When omitted, R, S, and E are set to the default values R=I, S=0, and E=I. Along with the solution P, care returns the gain matrix $G = R^{-1}(B^T PE + S^T)$

and a vector L of closed-loop eigenvalues, where $\mathrm{L = eig(A - B * G, E)}$

- $[P, L, G, report] = \mathrm{care(A, B, Q, \ldots\ldots)}$ returns a diagnosis report with:
    1) -1 when the associated Hamiltonian pencil has eigenvalues on or very near the imaginary axis (failure)
    2) -2 when there is no finite stabilizing solution P
    3) The Frobenius norm of the relative residual if P exists and is finite.

This syntax does not issue any error message when P fails to exist.

- $[P_1, P_2, D, L] = \mathrm{care(A, B, Q, \ldots, 'factor')}$ returns two matrices P_1 , P_2 and a diagonal scaling matrix D such that $P = D * \frac{P_2}{P_1} * D$

The vector L contains the closed-loop eigenvalues. All outputs are empty when the associated Hamiltonian matrix has eigenvalues on the imaginary axis.

DARE Command:

$[P, L, G] = \mathrm{dare(A, B, Q)}$

$[P, L, G] = \mathrm{dare(A, B, Q, R, S, E)}$

$[P, L, G, report] = \mathrm{dare(A, B, Q, \ldots\ldots)}$

$[P_1, P_2, D, L] = \mathrm{dare(A, B, Q, \ldots, 'factor')}$

Description

- $[P, L, G] = \mathrm{dare(A, B, Q)}$ computes the unique stabilizing solution P of the discrete-time algebraic Riccati equation

$$A^T PA - P - A^T PB(B^T PB + R)^{-1} B^T PA + Q = 0$$

The dare function also returns the gain matrix $G = (B^T PB + R)^{-1} B^T PA$ and the vector L of closed loop eigenvalues, where $\mathrm{L = eig(A - B * G, E)}$

- $[P, L, G] = \mathrm{dare(A, B, Q, R, S, E)}$ solves the more general discrete-time algebraic Riccati equation

$$A^T PE + E^T PE - (A^T PB + S)(B^T PB + R)^{-1}(B^T PA + S^T) + Q = 0$$

or, equivalently, if R is nonsingular,

$$E^T PE = F^T PF - F^T PB(B^T PB + R)^{-1} B^T PF + Q - SR^{-1} S^T$$

Where $= A - BR^{-1}S$. When omitted, R, S, and E are set to the default values R=I, S=0, and E=I. The dare function returns the gain matrix

$G = (B^T PB + R)^{-1}(B^T PA + S^T)$

and a vector L of closed-loop eigenvalues, where $\mathrm{L = eig(A - B * G, E)}$

- $[P, L, G, report] = \mathrm{dare(A, B, Q, \ldots\ldots)}$ returns a diagnosis report with:
  3) -1 when the associated symplectic pencil has eigenvalues on or very near the unit circle
  4) -2 when there is no finite stabilizing solution P
  3) The Frobenius norm if P exists and is finite

This syntax does not issue any error message when P fails to exist.

- $[P_1, P_2, D, L] = \mathrm{dare(A, B, Q, \ldots, 'factor')}$ returns two matrices P_1 , P_2 and a diagonal scaling matrix D such that $P = D * \frac{P_2}{P_1} * D$.

  The vector L contains the closed-loop eigenvalues. All outputs are empty when the associated Symplectic matrix has eigenvalues on the unit circle.

Appendix B
MATLAB Codes for continuous system

1- PID Design without delay Code

```
close all
clear all
clc
%******Design of PID controller (discrete model)
numc=[1];
denc=conv([1.5 1],[0.4 1])

%starting value of PID Gaines********************
Kp=1;
Ki=.1;
Kd=.1;
tsim=5;

%PIDnondelay_test

sim('PIDnondelay_test');

% parameters
%tamletra=12;
%tamnum=12;
plot(t,y,'--',t,ref,':','LineWidth',2)
%plot(t,y,'--',t,y1,':','LineWidth',2)
%legend('y for L=0.5','y for L=0','set-point','Location','SouthEast')
legend('output for non delay system','Location','SouthEast')
axis([0 tsim -0.1 1.8]);
%ylabel('output and set-point', 'FontSize', tamletra);
ylabel('output and set-point of PID');
%xlabel('time', 'FontSize', tamletra);
xlabel('time');
grid
```

2- PID Design with delay Code (Smith Predictor approach)

```
close all
clear all
clc
%******Design of PID controller
numc=[1];
denc=conv([1.5 1],[0.4 1])
 %******* Delay Model "Smith Predictor "**************
%       h=1;
%       h=2;
%       h=3.5;
      h=5;


%****** Gaines************************
%  Kp=14.85;
%  Ki=9.088;
```

```
%  Kd=4.613;
%******************Optimal value******************
Kp=1;
Ki=1.1;
Kd=.21;
%*********************************$$$$$$$$$$$$$$$$$$$$$$
tsim=20;

PIDsmith1
```

3- PID Design with delay Code (Pad'e approximation approach)

```
close all
clear all
clc
%******Design of PID controller (discrete model)
numc=[1];
denc=conv([1.5 1],[0.4 1])

%******* Delay Model "Pade Approximation"**************
 %       h=1;
%       h=2;
 %       h=3.5;
      h=5;
 nump=[-h/2 1];
 denp=[h/2 1];

%****** Gaines************************
%  Kp=14.85;
%  Ki=9.088;
%  Kd=4.613;
%******************Optimal value******************
Kp=.1;
Ki=.1;
Kd=.1;
%*********************************$$$$$$$$$$$$$$$$$$$$$$
tsim=30;

PIDpade
```

4- Pole placement Design without delay Code

```
%***** Heated Tank ********************
clc
close all
%*******Heated Tank SYSTEM Without Delay**************
num=[1];
den=[0.6 1.9 1];
%*************
H= tf (num,den)
[a,b,c,d] = tf2ss(num,den);
open_loop= eig(a)
%*********%********** General Solution ******************
```

```
r = 1;
co=ctrb(a,b)
ran=rank(co)
[x,l,k] = care(a,b,c'*c,r)
t = 0:0.01:5;
u = 1*ones(size(t));
sys = ss(a,b,c,0);
Nbar=rscale(sys,k);
sys_cl=ss(a-b*k,b,c,0);
lsim(sys_cl,Nbar*u,t);
axis([0 2 0 1.2])
grid
%*********** poles shifted to the left alpha  ********************
r = 1;
alpha =0.5
alpha_I = alpha*eye(2);
new_a = a-alpha_I;
co=ctrb(new_a,b);
ran=rank(co);
[x,l,k] = care(new_a,b,c'*c,r)
t = 0:0.01:5;
u = 1*ones(size(t));
sys = ss(new_a,b,c,0);
Nbar=rscale(sys,k);
sys_cl=ss(new_a-b*k,b,c,0);
figure
lsim(sys_cl,Nbar*u,t);
axis([0 2 -.1 1.2])
legend('output for non delay system','Location','SouthEast')
ylabel('output and set-point of Pole-placement');
%title('Pade approximation methode');
xlabel('time');
grid
```

5- Pole placement Design with delay Code (Pad’e approximation approach)

```
% Heated Tank ********************
clc
close all
%*******Heated Tank SYSTEM Without Delay***************
numc=[1];
denc=[0.6 1.9 1];
%******* Delay Model "Pad'e Approximation"**************
       h=1;
       %h=2;
      % h=3.5;
      % h=5;
 nump=[-h/2 1];
 denp=[h/2 1];
%nump=[h^2 -6*h 12];
%denp=[h^2 6*h 12];
%*******Heated Tank SYSTEM With Delay h ***************
num=conv(numc,nump)
den=conv(denc,denp)
H= tf (num,den)
[a,b,c,d] = tf2ss(num,den);
```

```
open_loop= eig(a)
%*********%********** General Solution ******************
r = 1;
co=ctrb(a,b)
ran=rank(co)
[x,l,k] = care(a,b,c'*c,r)
t = 0:0.01:20;
u = 1*ones(size(t));
sys = ss(a,b,c,0);
Nbar=rscale(sys,k);
sys_cl=ss(a-b*k,b,c,0);
lsim(sys_cl,Nbar*u,t);
axis([0 20 -.20 1.2])
grid
%*********** poles shifted to the left alpha *******************
r = 1;
alpha =.5
alpha_I = alpha*eye(size(a));
new_a = a-alpha_I;
co=ctrb(new_a,b);
ran=rank(co);
[x,l,k] = care(new_a,b,c'*c,r)
t = 0:0.01:20;
u = 1*ones(size(t));
sys = ss(new_a,b,c,0);
Nbar=rscale(sys,k);
sys_cl=ss(new_a-b*k,b,c,0);
figure
lsim(sys_cl,Nbar*u,t);
axis([0 20 -.2 1.2])
legend('output for h=5','Location','SouthEast')
ylabel('output and set-point of Pole-placement');
title('Pade approximation methode');
xlabel('time');
grid
```

6- Pole placement Design with delay Code (Smith Predictor approach)

```
%***** Heated Tank ********************
clc
close all
clear
%******* Heated Tank SYSTEM With Delay **************
s = tf('s');

h=5;
sys = 1/(0.6*s^2+1.9*s+1)
[a,b,c,d]= ssdata(sys)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
DelayT = struct('delay',h,'a',a,'b',b,'c',c,'d',d)
sys = delayss(a,b,c,d,DelayT)

%*********%********** General Solution ******************
r = 1;
co=ctrb(a,b)
ran=rank(co)
```

```
[x,l,k] = care(a,b,c'*c,r)
t = 0:0.01:h+5;
u = 1*ones(size(t));
sys = ss(a,b,c,0,'OutputDelay',h);
Nbar=rscale(sys,k);
sys_cl=ss(a-b*k,b,c,0,'OutputDelay',h);
lsim(sys_cl,Nbar*u,t);
axis([0 h+5 0 1.2])
grid

%*********** poles shifted to the left alpha  ********************
r = 1;
alpha =0.5
alpha_I = alpha*eye(2);
new_a = a-alpha_I;
co=ctrb(new_a,b);
ran=rank(co);
[x,l,k] = care(new_a,b,c'*c,r)
t = 0:0.01:9;
u = 1*ones(size(t));
sys = ss(new_a,b,c,0,'OutputDelay',h);
Nbar=rscale(sys,k);
sys_cl=ss(new_a-b*k,b,c,0,'OutputDelay',h);
figure;lsim(sys_cl,Nbar*u,t);
axis([0 h+1 -.1 1.2])
legend('output for h= 5','Location','SouthEast')
ylabel('output and set-point of Pole-placement');
title('Smith pre dictor method');
xlabel('time');
grid
```

MATLAB Codes for Digital system

1- PID design without Delay

```
%*******PID Controller
close all
clear all
clc
%******Design of PID controller (discrete model)
numc=[1];
denc=conv([1.5 1],[0.4 1])
%******Sampling Time
Ts=.1
%******Modefied z transforme of the plant
Kp=1;
Ki=.1;
Kd=.1;
[numd0,dend0]=c2d(numc,denc,Ts)
%tsim=20;
%PIDDiscrete
```

2- PID design with Delay (Modified z-transformed approach)

```
%*******PID Controller
close all
clear all
%******Design of PID controller (discrete model)
numc=[1];
denc=conv([1.5 1],[0.4 1]);
%******Sampling Time
Ts=.1;
%****** Delay equal Ts*****************
nonint_delay=.25;
delay=fix(nonint_delay/Ts)*Ts
deltal=nonint_delay-delay

%************** modeled error combined with the plant************
numc=[1];
denc=conv(conv([deltal 1],[1.5 1]),[0.4 1]);
%******Modefied z transforme of the plant
[numd3,dend3]=cp2dp(numc,denc,Ts,delay);
Kp=1;
Ki=.1;
Kd=.1;
tsim=20;
PIDDiscretebignonint
```

3- Pole placement design without Delay (Modified z-transformed approach)

```
% Heated tank********
clc
close all
%******** Heated Tank System without Delay (Discret
Model)********************
numc=[1];
denc=[0.6 1.9 1];
Ts=1/10;
[A,B,C,D] = tf2ss(numc,denc);
Htank = ss(A,B,C,D);
%*****************************************************************
    % AUV_tf=tf(numc,denc);
    % AUV_tf_d=c2d(AUV_tf,Ts,'zoh')%Discrete transfer functon
    % [numd,dend]=tfdata(AUV_tf_d,'v')
    % [F,G,H,J] = tf2ss(numd,dend)
    %***************** PZ MAP **************
    % sys_d=tf(numd,dend,Ts)
    % pzmap(sys_d)
    % axis([-1 1 -1 1])
    % zgrid
%******************************************************************
Htank_d = c2d(Htank,Ts,'zoh');%Discrete state space
[F,G,H,J]=ssdata(Htank_d)
    %AUV_d = ss(F,G,H,J,Ts)
open_poles=eig(F)
co = ctrb(Htank_d);
ob = obsv(Htank_d);
```

```
Controllability = rank(co)
Observability = rank(ob)
%***********************************
t = 0:0.01:10;
u = 1*ones(size(t));
%************Riccati Equation*******
R=1;
%zeta= .7;r=.2;alpha=.08(r/alpha)=2.5;
r = .2;
alpha=r/2.5;
alpha_I=alpha*eye(size(F));
gama= r+alpha;
    %maxlimit=gama+r
gama_I = gama*eye(size(F));
    %new_F  = ((F-gama_I)/r)-alpha_I;
new_F  = ((F-gama_I)/r);
co=ctrb(new_F,G);
ran=rank(co);
[x,l,K] = dare(new_F,G,H'*H,R*r^2)

%************without delay**********************
Nbar = 8.07;
%**************************************************
sys_cl = ss(new_F-G*K,G*Nbar,H,J,Ts);
    %***************** PZ MAP **************
    % %sys_d=tf(numd,dend,Ts)
    % sys_d_cl= ss2tf(new_F-G*K,G*Nbar,H,J,Ts);
    % pzmap(sys_d_cl)
    % axis([-1 1 -1 1])
    % zgrid
%******************************************
figure
[y] = lsim(sys_cl,u);
stairs(t,y)
axis([0 1 -.1 1.5])
legend('y for nondelayed system h=0','Location','SouthEast')
 ylabel('output ');
 xlabel('time');
 grid
```

4- Pole placement design with Delay (Modified z-transformed approach)

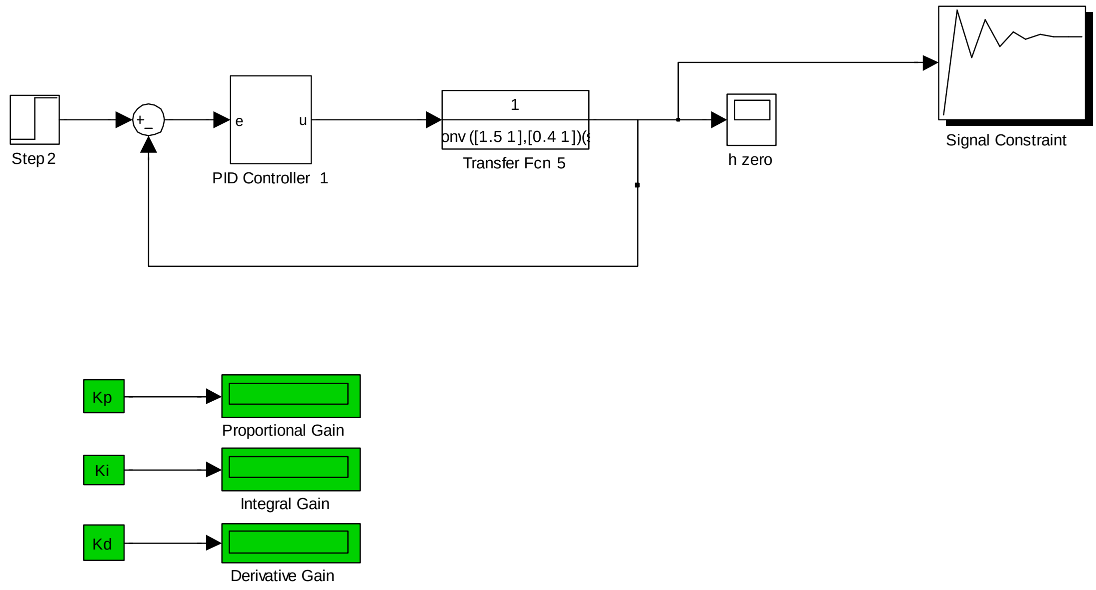
```
%*** Heated Tank System ***
clc
close all
%******** Heated Tank System with Delay (Discret
Model)*******************
numc=[1];
denc=[0.6 1.9 1];
%********** def. DELAY
% delay=.1;
% r = .33
% alpha=r/2.5
% Nbar = 1.356;
```

```
% %*****************
%   delay=.2;
%   r = .33
%   alpha=r/2.5
%   Nbar = .92;
% %*******************
   delay=.5;
   r = .33
   alpha=r/2.5
   Nbar = .095;

% ****** Delay (noninteger multiple of Ts) **************
%         nonint_delay=.25;
%         ************** modeled error combined with the
plant************
%         numc=[1];
%         denc=conv(conv([.05 1],[1.5 1]),[0.4 1]);
%         delay=.2;
%         r = .33
%         alpha=r/2.5
%         Nbar = .76;
% %********************
    % sys=tf(numc,denc)
    % fb=bandwidth(sys)
Ts=1/10;
%****modified z-transform
[numd,dend]=cp2dp(numc,denc,Ts,delay);
%***************** PZ MAP **************
sys_d=tf(numd,dend,Ts)
figure
step(sys_d)
grid
figure
pzmap(sys_d);
axis([-1 1 -1 1]);
zgrid;
%*****************************************
[F,G,H,J] = tf2ss(numd,dend)
AUV_d = ss(F,G,H,J,Ts);
open_poles=eig(F)
co = ctrb(AUV_d);
ob = obsv(AUV_d);
Controllability = rank(co)
Observability = rank(ob)
%***********************************
 t = 0:Ts:10;
 u = 1*ones(size(t));
%***********Riccati Equation*******
R=1;
%zeta= .7;r=.2;alpha=.08(r/alpha)=2.5;
%r = .33
%alpha=r/2.5
alpha_I=alpha*eye(size(F));
gama= r+alpha;
maxlimit=alpha+2*r
gama_I = gama*eye(size(F));
%new_F  = ((F+gama_I)/r)+alpha_I;
new_F  = ((F-gama_I)/r);
co=ctrb(new_F,G);
ran=rank(co);
[x,l,K] = dare(new_F,G,H'*H,R*r^2)
```

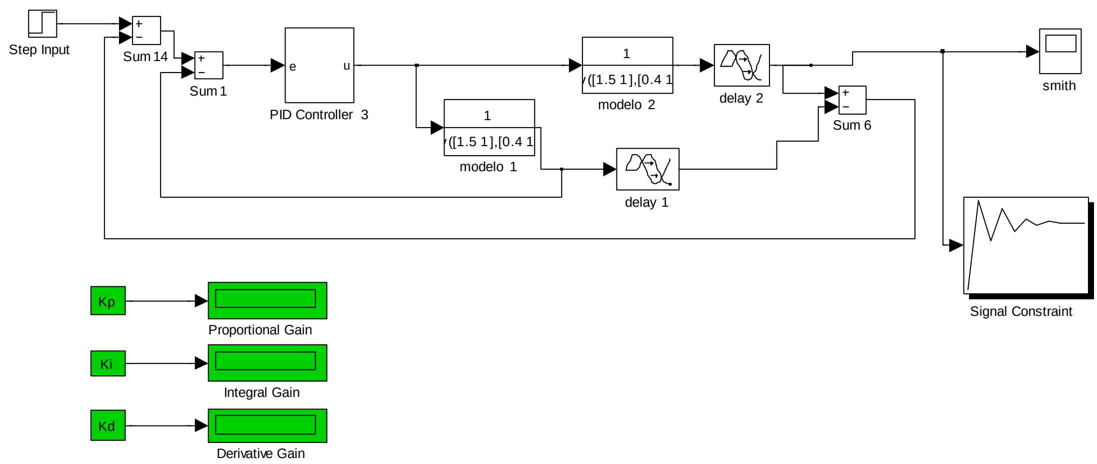
```
    %[x,l,K] = dare(F,G,H'*H,R*r^2)
%***************with delay************************
    %Nbar = 7100;
%**************************************************
sys_cl = ss(new_F-G*K,G*Nbar,H,J,Ts);
%****************** PZ MAP **************
    % figure
    % sys_d=tf(numd,dend,Ts);
    % sys_d_cl= ss2tf(new_F-G*K,G*Nbar,H,J);
    % sys_d_cl1=tf(sys_d_cl)
    % pzmap(sys_d_cl1)
    % axis([-1 1 -1 1])
    % zgrid
%******************************************
figure
[y] = lsim(sys_cl,u,t);
stairs(t,y)
%axis([0 2 -.1 1.5])

legend('y for h=0.5','Location','SouthEast')
 ylabel('output ');
 xlabel('time');
 grid
```

Appendix C
Design using Simulink toolbox

1- PID controller without Delay



2- PID controller Design with Delay(Smith Predictor approach)



3- PID controller Design with Delay(Modified z-transformed approach)