UNIVERSITY OF ÇUKUROVA
INSTITUTE OF NATURAL AND APPLIED SCIENCES

MSc. THESIS

Salih Serhan YURDAKUL

SELF-TUNING PID CONTROL WITH EXPERIMENTAL APPLICATIONS

DEPARTMENT OF ELECTRICAL AND ELECTRONICS ENGINEERING

ADANA, 2009

ÇUKUROVA ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

SELF-TUNING PID CONTROL WITH EXPERIMENTAL APPLICATIONS

Salih Serhan YURDAKUL

YÜKSEK LİSANS TEZİ

ELEKTRİK-ELEKTRONİK MÜHENDİSLİĞİ

Bu tez 25 / 03 / 2009 Tarihinde Aşağıdaki Jüri Üyeleri Tarafından Oybirliği ile Kabul Edilmiştir.

İmza.....	İmza.....	İmza.....
Doç.Dr. İlyas EKER	Yrd.Doç.Dr. Ulus ÇEVİK	Yrd.Doç.Dr. Ramazan ÇOBAN
Danışman	Üye	Üye

Bu tez Enstitümüz Elektrik ve Elektronik Mühendisliği Anabilim Dalında hazırlanmıştır.

Kod No : 3412

T.C.

Prof. Dr. Aziz ERTUNÇ
Enstitü Müdürü
İmza ve Mühür

Not: Bu tezde kullanılan özgün ve başka kaynaktan yapılan bildirişlerin, çizelge, şekil ve fotoğrafların kaynak gösterilmeden kullanımı, 5846 sayılı Fikir ve Sanat Eserleri Kanunundaki hükümlere tabidir.

ABSTRACT

MSc. THESIS

SELF-TUNING PID CONTROL WITH EXPERIMENTAL APPLICATIONS

Salih Serhan YURDAKUL

DEPARTMENT OF ELECTRICAL AND ELECTRONICS ENGINEERING
INSTITUTE OF NATURAL AND APPLIED SCIENCES
UNIVERSITY OF ÇUKUROVA

Supervisor : Assoc.Prof. Dr. İlyas EKER

Year : 2009 Pages: 79

Jury : Assoc.Prof. Dr. İlyas EKER
Assist.Prof. Dr. Ulus ÇEVİK
Assist.Prof. Dr. Ramazan ÇOBAN

Proportional-Integral-Derivative (PID) controllers have been widely used in industry and these constitute an important part of industrial control systems. So any development in PID design and implementation methodology is important for industrial control systems.

Because of rapidly increasing development in computer technology, computational complexity of self-tuning control algorithm is not a limitation any more. This situation makes the self-tuning PID control an important alternative to conventional PID control.

In this thesis, self-tuning pole assignment PID control is presented. Experimental implementation is performed in laboratory. The results of the present research are compared with that of obtained from conventional PID and some other PID control methods, thus the advantages and disadvantages of self-tuning PID control are emphasized.

Keywords: Self-tuning control, PID control, pole assignment control.

ÖZ

YÜKSEK LİSANS TEZİ

SELF-TUNING PID CONTROL WITH EXPERIMENTAL APPLICATIONS

Salih Serhan YURDAKUL

ELEKTRİK - ELEKTRONİK MÜHENDİSLİĞİ ANABİLİM DALI
FEN BİLİMLERİ ENSTİTÜSÜ
ÇUKUROVA ÜNİVERSİTESİ

Danışman : Doç. Dr. İlyas EKER

Yıl : 2009, Sayfa: 79

Juri : Doç. Dr. İlyas EKER
Yrd.Doç. Dr. Ulus ÇEVİK
Yrd.Doç. Dr. Ramazan ÇOBAN

Oransal-İntegral-Türevsel (PID) kontrolcüler endüstride yaygın olarak kullanılmakta ve endüstriyel kontrol sistemlerinin önemli bir parçasını oluşturmaktadır. Bu yüzden PID tasarım ve uygulama metotlarında yapılacak herhangi bir gelişme endüstriyel kontrol sistemleri için önemli yer teşkil edecektir.

Bilgisayar teknolojilerindeki hızla artan gelişmeler sayesinde kendinden-ayarlamalı kontrol algoritmalarındaki hesap karmaşası bir sınırlayıcı etmen olmaktan çıkmıştır. Bu durum sayesinde kendinden-ayarlamalı PID kontrol diğer klasik PID kontrole göre önemli bir alternatif olmuştur.

Bu çalışmada kendinden-ayarlamalı kutup atamalı PID kontrol sunulmuştur. Deneysel uygulamalar laboratuar ortamında gerçekleştirilmiştir. Elde edilen araştırma sonuçları klasik PID ve diğer bazı PID kontrol metotlarından elde edilen sonuçlar ile karşılaştırılmış, böylece kendinden ayarlamalı PID kontrolün avantajları ve dezavantajları vurgulanmıştır.

Anahtar Kelimeler: Kendinden-ayarlamalı kontrol, PID kontrol, kutup atamalı kontrol.

ACKNOWLEDGEMENTS

I would like to thank my supervisor, Assoc. Prof. Dr. İlyas EKER for his valuable support and guidance during this research and preparation of this thesis. I would like to thank also my family for their continuous encouragement and morale support. Many thanks are also to my friends especially Mehmet DURSUN and Alkan ALKAYA for their supports. In addition, special thanks to Assist. Prof. Dr. Tolgay KARA from Gaziantep University for his supports to experimental works.

TABLE OF CONTENTS **PAGE**

ABSTRACT……I
ÖZ……II
ACKNOWLEDGEMENTS ……III
TABLE OF CONTENTS……IV
LIST OF TABLES……VI
LIST OF FIGURES……VII
1 INTRODUCTION 1
1.1 History of Adaptive Control 1
1.2 Self-Tuning Systems 2
1.3 Development of Self-tuning Controllers 3
1.4 Literature Survey 4
2 PID CONTROL 6
2.1 The Review of PID History 6
2.2 The Feedback Principles 8
2.3 On-Off Control 9
2.4 Proportional Control 10
2.5 Integral Control 11
2.6 Derivative Control 12
2.7 Proportional + Integral (PI) Control 13
2.8 Proportional + Integral + Derivative (PID) Control 14
3 SELF-TUNING CONTROL 16
3.1 System Identification for Self-Tuning 17
3.1.1 Least Squares Algorithm 18
3.1.2 Recursive Least Squares Algorithm 22
3.2 Controller Tuning Methods 25
3.2.1 Ziegler-Nichols Tuning Methods 25
3.2.2 Cohen-Coon Reaction Curve Method 28
3.2.3 Chien-Hrones-Reswick Method 29
3.2.4 Pole Assignment Self-Tuning Control 30

4 POLE ASSIGNMENT SELF-TUNING PID CONTROL 35
4.1 Plant Model 35
4.2 Controller Design 37
4.3 Design of $T(z^{-1})$ 41
4.4 Digital PID Implementation 42
5 EXPERIMENTAL APPLICATIONS 45
5.1 Hardware of the Experimental System 45
5.2 Software of the Experimental System 46
5.3 Plant Model 47
5.4 Ziegler - Nichols Process Reaction Method Results 53
5.5 Cohen – Coon Reaction Curve Method Results 57
5.6 Self-Tuning Pole Assignment Method Results 61
6 CONCLUSIONS 73
REFERENCES……...……75
BIOGRAPHY……...…...……79

LIST OF TABLES **PAGE**

Table 3.1 PID controller parameters obtained from the Z-N PRC method. 27
Table 3.2 PID controller parameters obtained from the Z-N oscillation method. 28
Table 3.3 PID controller parameters obtained from the Cohen-Coon method. 29
Table 3.4 Chien-Hrones-Reswick method with 0% overshoot. 29
Table 3.5 Chien-Hrones-Reswick method with 20% overshoot. 30
Table 5.1 Time domain specifications of Z-N process reaction method. 57
Table 5.2 Time domain specifications of Cohen-Coon process reaction method..... 60
Table 5.3 Time domain specifications of self-tuning PI and PID control system. 66
Table 5.4 Time domain specifications. 68
Table 5.5 Time domain specifications for load test. 72

LIST OF FIGURES **PAGE**

Figure 2.1 Block diagram of a simple feedback system. 9
Figure 2.2 Controller characteristics for ideal on-off control (a) relay control, (b) dead-zone control, (c) hysteresis control. 10
Figure 2.3 Characteristic of proportional control..... 10
Figure 3.1 The three stages of control system. 16
Figure 3.2 Self-tuning controller structure..... 17
Figure 3.3 Transfer function of a system 18
Figure 3.4 Scheme of recursive least squares method 23
Figure 3.5 Z-N process reaction method. 27
Figure 3.6 Design of a feedback controller. 30
Figure 3.7 Self-tuning pole assignment system. 33
Figure 3.8 Timing and sequence diagram for self-tuning. 34
Figure 4.1 Block diagram of ARX model..... 36
Figure 4.2 Block diagram of control loop. 38
Figure 5.1 Experimental system. 45
Figure 5.2 Hardware of the experimental system. 46
Figure 5.3 The block diagram of the experimental system. 47
Figure 5.4 Schematic diagram of the electrical drive system. 49
Figure 5.5 Step response of DC motor. 50
Figure 5.6 Step response of model. 51
Figure 5.7 Step responses of measured output and model output. 52
Figure 5.8 Modeling error. 52
Figure 5.9 Responses to step setpoint change for P controller. 54
Figure 5.10 Responses to step setpoint change for PI controller. 54
Figure 5.11 Responses to step setpoint change for PID controller. 55
Figure 5.12 Response to step setpoint change for Z-N PRC method..... 56
Figure 5.13 Control signals of Z-N PRC method. 56
Figure 5.14 Responses to step setpoint change for P controller. 58
Figure 5.15 Responses to step setpoint change for PI controller. 58

Figure 5.16 Responses to step setpoint change for PID controller. 59
Figure 5.17 Responses to step setpoint change for Cohen-Coon RCM. 59
Figure 5.18 Control signals of Cohen-Coon RCM designed control system. 60
Figure 5.19 Response of the self-tuning PI control system. 62
Figure 5.20 Variations in identification parameters for self-tuning PI control. 62
Figure 5.21 Variations of self-tuning PI controller gains. 63
Figure 5.22 Responses of the self-tuning PID control system. 64
Figure 5.23 Variations of identification parameters for self-tuning PID control. 64
Figure 5.24 Variations of self-tuning PID controller gains. 65
Figure 5.25 Responses to step setpoint change with self-tuning PI and self-tuning PID controllers. 65
Figure 5.26 Control signals of self-tuning PI and self-tuning PID control system. 66
Figure 5.27 PI controller responses to step setpoint change for self-tuning, Z-N and Cohen-Coon system. 67
Figure 5.28 PID controller responses to step setpoint change for of self-tuning, Z-N and Cohen-Coon system. 68
Figure 5.29 Responses of different control systems. 69
Figure 5.30 Control signals of the systems. 69
Figure 5.31 Load Disturbance 70
Figure 5.32 Output speeds to ±91.87 rpm square wave external load disturbance. 71
Figure 5.33 Control signals to ±91.87 rpm square wave external load disturbance. 71

1 INTRODUCTION

1.1 History of Adaptive Control

According to the Webster's dictionary, "to adapt" means: "To adjust oneself to particular conditions; To bring oneself in harmony with a particular environment; To bring one's acts, behavior in harmony with a particular environment", while adaptation means: "adjustment to environmental conditions; alteration or change in form or structure to better fit the environment". For a control system, the plant constitutes the environment. Plant nonlinearities can be found in most of the processes from flight to process control. For instance, in steel rolling mills, paper machines or rotary kilns, the dynamics can change due to nonlinear actuators or sensors (e.g. nonlinear valves, pH probes), flow and speed variations, raw material variability or wear and tear. For an aircraft, nonlinearities are mainly correlated with the compressibility of the air and the turbulent flow around control and lift surfaces. In ship steering, changing wave characteristics represent a major challenge. It is well recognized that linear feedback can cope fairly well with parameter changes within certain limits [Dumont, Huzmezan, 2002].

Most current techniques for designing control systems are based on a good understanding of the plant under study and its environment. However, in a number of instances, the plant to be controlled is too complex and the basic physical processes in it are not fully understood. Control design techniques then need to be augmented with an identification technique aimed at obtaining a progressively better understanding of the plant to be controlled. It is thus intuitive to aggregate system identification and control. Often, the two steps will be taken separately. If the system identification is recursive, that is, the plant model is periodically updated on the basis of previous estimates and new data-identification and control may be performed concurrently [Sastry, Bodson, 1994].

Abstractly, system identification could be aimed at determining if the plant to be controlled is linear or nonlinear, finite or infinite dimensional, and has continuous or discrete event dynamics. Applications of such systems arise in several contexts: advanced flight control systems for aircraft or spacecraft, robot manipulators, process control, power systems, and others [Sastry, Bodson, 1994].

Adaptive control, then, is a method of applying some system identification technique to obtain a model of the process and its environment from input-output experiments and using this model to design a controller. The parameters of the controller are adjusted during the operation of the plant as the amount of data available for plant identification increases. For a number of simple PID (**P**roportional + **I**ntegral + **D**erivative) controllers in process control, this is often done manually. However, when the number of parameters is larger than three or four and they vary with time, automatic adjustment is needed. The design techniques for adaptive systems are studied and analyzed in theory for unknown but fixed (that is, time invariant) plants. In practice, they are applied to slowly time-varying and unknown plants [Sastry, Bodson, 1994].

1.2 Self-Tuning Systems

Conventional control system design procedure needs to know about system behavior and system characteristics. Additionally, constant coefficients must be suitable for all determined operating conditions. However, most systems are not suitable to determine system characteristics for all conditions because, these systems include undetermined parts and these parts have some difficulties to find their characteristics. For that reason, tuning and calibration is necessary to most conventional control systems.

The basic idea of a self-tuning system is to construct an algorithm that will automatically change its parameters to meet a particular requirement or situation. Tuning concept includes two different mechanism approaches: The first approach is based on initial auto-tuning of system parameters which are time-invariant but not

desired constantly by the designer. Thus, after initial tuning, the adjustment mechanism is not required and can be disabled until another tuning process is requested by a control mechanism or system operator. The second is based on continuous adjustment so the adjustment mechanism works continuously. However, initial-adjusted systems are called auto-tuning and continuous adjusting systems are called self-tuning [Sastry, Bodson, 1994]. Their properties are:

- **Auto-tuning**
  - Once controller parameters near convergence, adaptation is stopped.
  - Used for time invariant or very slowly varying processes.
  - Used for periodic, usually on-demand tuning.
- **Self-tuning**
  - Continuous updating of controller parameters.
  - Used for truly time-varying plants.

1.3 Development of Self-tuning Controllers

The approach used in **S**elf-**T**uning **C**ontrollers (STC) was first mentioned in the work of Kalman in 1958 [Kalman, 1958]. Single-purpose computer to identify the parameters of a linear model process and subsequently was designed the control is calculated using minimum quadratic criteria. This problem was revived in the early 1970s by the work of Peterka [Peterka, 1970], Astrom and Wittenmark [Aström, Wittenmark, 1973] and others. The approach has been developed significantly since then. The first STCs were designed so as to minimize system output variance where some disadvantages were removed by the general minimization of output dispersion method developed by Clarke and Gawthrop [Clarke, Gawthrop, 1975]. These are known as single-step methods because only one sample of the process output is considered in the quadratic criterion. One great disadvantage is that they are unable to control the so called nonminimum phase systems.

This problem can be solved by using multi-step criteria (to limit an infinite number of steps) which is a solution to the general quadratic problem. Peterka

analyzed the probability rate of the Bayessian approach to adaptive control based on linear quadratic synthesis [Peterka, 1986]. In general this control synthesis involves rather complex iteration calculations [Karny, Halouskova, Böhm, Kulhavi, Nedoma, 1985]. It has been shown that analytic methods may be used to find relatively simple explicit relationships to determine the optimal controller for one-dimensional models that are no higher than second order [Böhm, Halouskova, Karny, Peterka, 1984].

In the late 1970s, early 1980s the first work was done on STCs based on pole assignment [Wellstead, Edmunds, Prager, Zanker,1979]. During the 1980s much attention was also paid to single- and multi-step prediction adaptive methods [Clarke, Mohtadi, Tuffs, 1987]. Hybrid STCs using a δ operator have similarly been analyzed [Gawthrop, 1980]. Alongside these developments there has been exploration into synthesis methods for digital PID controllers which might be able to use parameter estimates gained through recursive identification to calculate controller intervention. These parameters are then used to calculate the PID controller elements, that is, gain K_p, and integral and derivative time constants T_i and T_d [Banyasz, Keviczky, 1982].

1.4 Literature Survey

Ren and his friends presented the motion control and stability analysis of a Two-Wheeled Vehicle (TWV) [Ren, Chen, Tsai, 2008]. The TWV is driven using two independent wheel motors, upon which a vehicle body is mounted. A mathematical model of the TWV is obtained using dynamic analysis. The TWV is inherently unstable and its motion is controlled through the actions of the wheel motors. Vehicle action depends on both the desired wheel response and the tilt angle. A self-tuning **P**roportional-**I**ntegral-**D**erivative (PID) control strategy, based on a deduced model, is proposed for implementing a motion control system that stabilizes the TWV and follows the desired motion commands. The controller parameters are tuned automatically, on-line, to overcome the disturbances and parameter variations.

A new PID parameter self-tuning algorithm was proposed which adapts to single plant [Zhong, Chunpeng, Dawei, 2007]. Guaranteeing the stabilization of closed loop systems and satisfying the conditions of self-tuning, the recursive least

squares method and estimate the change of model parameter of the plant online can be used.

A tuning method was presented for determining the parameters of PID control for an automatic voltage regulator (AVR) system using a chaotic optimization approach based on Lozi map [Coelho, 2007]. Since chaotic mapping enjoys certainty, ergodicity and the stochastic property, the proposed chaotic optimization introduces chaos mapping using Lozi map chaotic sequences which increases its convergence rate and resulting precision. Simulation results are promising and show the effectiveness of the proposed approach. Numerical simulations based on proposed PID control of an AVR system for nominal system parameters and step reference voltage input demonstrate the good performance of chaotic optimization.

Yamamoto et al. [Yamamato, Kaneda, Tanaka, 1995] presented a simple scheme for designing self-tuning pole-assignment controllers which can take account of the stability margin and control performance without solving the Diophantine equation. Furthermore, a self-tuning PID control algorithm is derived based on this method.

Gawthrop [Gawthrop, 1986] presented class of controllers with integral action is shown to arise directly from appropriate system models. Via the zero-gain predictor approach, a corresponding class of hybrid self-tuning controllers is shown to have both integral actions in the controller and offset removal in the tuning algorithm.

Yamamoto et al. [Yamamoto, Fujii, Kaneda, 1996] proposed a self-tuning algorithm of PID gains for unknown time-delay systems. The relationship between PID and pole assignment control system was shown first and then a tuning algorithm for PID gains based on this relationship was derived. Furthermore this method was extended to an explicit self-tuning structure which enables us to construct self-tuning PID control systems for nonminimum-phase with unknown time-delays.

Fong-Chwee and Sirisena [Fong-Chwee, Sirisena, 1988] proposed self-tuning PID controllers which can overcome fractional dead time, known and constant dead time plus time varying dead time.

2 PID CONTROL

2.1 The Review of PID History

PID controllers have survived many changes in technology. In 1920's it begins with pneumatic controllers, trough directional control to the Distributed Control System (DCS) [YU, 1999]. Pneumatic controllers were based on the flapper-nozzle amplifier. Movement of the flapper arm towards or away from the nozzle causes a change of back pressure in the pneumatic circuits and this change in pressure results in a movement of a diaphragms bellows. This movement can be applied to a pilot valve which in turn controls the opening and closing of the main control valve [Bennett, 2001].

Despite rapid evolution in control hardware over past 50 years, the PID controllers remain the workhouse in process industries. The proportional action (P mode) adjusts controller output according the size of the error. The integral action (I mode) can eliminate the steady state offset and future trend is anticipated via derivative action (D mode) [YU, 1999].

During the 1920s there was much discussion of the need a controller to anticipate an increase in the error and there were a variety of proposals to make controllers respond to a rate of change in the measured variable. Most schemes, however, did not provide derivative control action since the actuating mechanism introduced an integral term. The so called anticipating control resulted in the controlled variable being made proportional to error this did give a faster response since it replaced controllers in which the controlled variable was proportional to the integral of error. The derivative control action resulted from work being carried out by the Taylor Instrument Companies on the control of part of the rayon making process [YU, 1999].

The Foxboro Company initially dealt with the transfer of lag by the addition of a device which they called an 'Impulsator'. It was applied an impulse to the control

valve which was proportional to rate of change of error. The device was only available with the potentiometric Stabilog that is the pneumatic controller which operated with a thermocouple input. The addition of derivative action to the standard Stabilog, called 'hyper reset' by Foxboro, was the work of George A. Philbrick, was developed during 1937-38 [Bennett, 2001].

Importance of the PID controller had been demonstrated in several difficult applications and by 1940 two of the leading instrument companies were offering pneumatic controllers for sale; but much needed still to be done before it could become widely used in industry. There were three main problems to be solved. The first how to find appropriate settings for the controller, providing a simple means for adjustment in the field was useless if there no was no easier way of finding the best settings. The second was to persuade designers to produce plants which were controllable. And third was to make the operation of the controller less dependent on complex and fragile mechanical linkages.

The first problem was quickly solved by J.B. Ziegler and N.B. Nichols of Engineering Sales and Engineering Research Department of Taylor Instrument Company in 1942. They published well known paper 'Optimum Setting for Automatic Controllers'.

A second less well known paper, by Ziegler and Nichols appeared one year later in which they commented that too often in process plants when the plants is run it does not work as expected. The engineers release that some factor has been neglected but cannot identify what is missing. The missing thing can call as 'controllability' the ability of the process to achieve and maintain the desired equilibrium value.

The third problem was more difficult to deal with, particularly as engineering effort and resources were diverted to the war effort. After the end of the war the leading companies, Foxboro and Taylor made minor changes to the existing designs, improving the mechanics and the methods for adjusting the controller parameters. The Foxboro 'Model 40 Stabilog', which is appeared in 1948, was a result of more substantial design changes. The Foxboro 'Model 58 Consotrol' range which appeared in the early 1950s was the result of major redesign in that incorporated a clever force balance arrangement. This was not first force-balance type controller for

example the Leeds and Northrup Company's pneumatic Micromax of 1944 used a force-balance arrangement but this was the first such instrument from the leading pneumatic controller company.

Several companies had produced electronic controllers since the late 1930s, and A.J. Young writing in 1955, described six electronic PID controller produced by six different company. Electronic instrument were capable of performing all the functions previously only available with pneumatic instruments and that these included in addition to PID, the ability to carry out; addition, multiplication, squaring, and other mathematical operation. The instrument manufacturers were fully aware of the possibilities of transistor, and new products using transistor were being developed. In 1995, C.E. Mathewson, using frequency response analysis of pneumatic and electronic components, sought to demonstrate that the elimination of time lags possible with electronic control gave improved performance at both low and high frequencies and also that electronic controller could be much more easily connected to digital read out and logging system. By this time the digital operation was just beginning to be used in process control [Bennett, 2001].

2.2 The Feedback Principles

The feedback idea is simple and powerful. It has had a profound influence on technology. Assume for simplicity that the process is such that the process variable increases when the manipulated variable is increased. The principle of feedback can then be expressed as follows: The manipulated variable is increased when the process variable is smaller than the setpoint and is decreased when the process variable is larger than the setpoint.

This type of feedback is called *negative feedback* because the manipulated variable moves in opposite direction to the process variable. The feedback principle can be illustrated by the block diagram shown in Fig. 2.1, where r and y denote the setpoint and measured output respectively, e is the error between setpoint and the measured output and u is the control input.

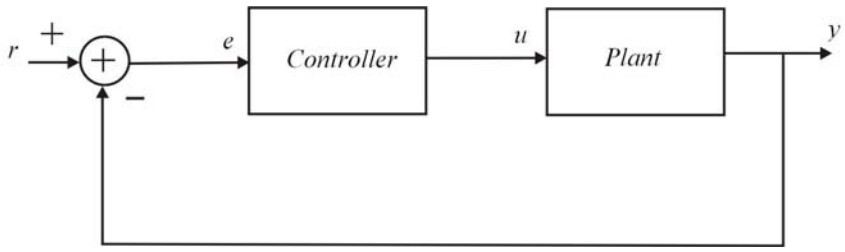


Figure 2.1 Block diagram of a simple feedback system.

The reason why feedback systems are of interest is that feedback makes the process variable close to the setpoint in spite of disturbances and variation of the plant characteristics [Aström, Hagglund, 1995].

2.3 On-Off Control

The feedback can be arranged in many different ways. A simple feedback mechanism can be described mathematically in on-off control as follows:

$$u = \begin{cases} u_{\max} & if \quad e > 0 \\ u_{\min} & if \quad e < 0 \end{cases} \tag{2.1}$$

where u_{max} and u_{min} denote maximum and minimum control inputs, respectively.

This control law implies that maximum corrective action is always used. The manipulated physical variable has its largest value when the error is positive and its smallest value when the error is negative. This type of feedback is called *on-off control.* It is simple and there are no tuning parameters to choose. On-off control often succeeds in keeping the process variable close to the setpoint, but it will typically result in a system where the variables oscillate. Notice that in Eq. (2.1) the control variable is not denned when the error is zero. It is common to have some modifications either by introducing hysteresis or a dead zone as shown in Fig. 2.2 [Aström, Hagglund, 1995].

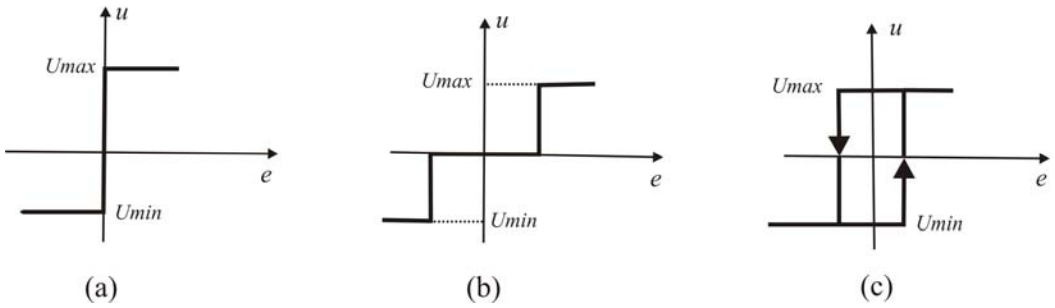


Figure 2.2 Controller characteristics for ideal on-off control (a) relay control, (b) dead-zone control, (c) hysteresis control.

2.4 Proportional Control

On-off control often introduces to oscillations; system overreacts because a small change in the error will make the manipulated variable change over the full range. This effect is avoided in proportional control where the characteristic of the controller is proportional to the control error for small errors as shown in Fig. 2.3 [Paraskevopoulos, 1996].

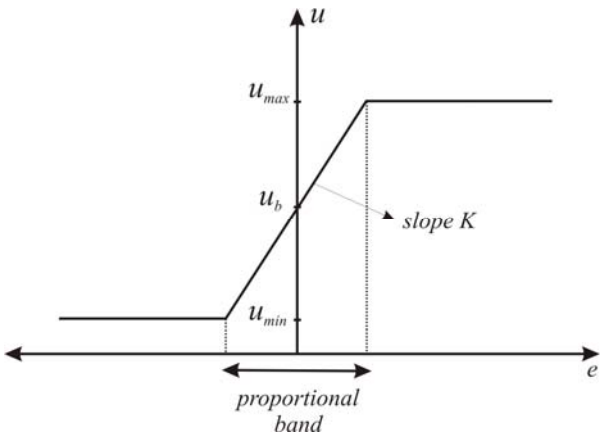


Figure 2.3 Characteristic of proportional control

To describe the characteristic of a proportional controller we must of course give the limits u_{max} and u_{min} of the control variable. The linear range can be specified either by giving the slope of the characteristic (controller gain, K_p) or by giving the

range where the characteristic is linear (proportional band, P_b). This range is normally centered around the setpoint. The proportional band and the controller gain are related as:

$$u_{\max} - u_{\min} = P_b.K_p \tag{2.2}$$

$$K_p = \frac{100}{P_b} \tag{2.3}$$

Notice that a proportional controller acts like an on-off controller for large errors.

For continuous-time systems, the proportional control is described as

$$u(t) = K_p e(t) \tag{2.4}$$

where the controller gain is:

$$G_c(s) = K_p \tag{2.5}$$

For discrete-time systems, the control input is described as:

$$u(k) = K_p e(k) \tag{2.6}$$

where the proportional controller is

$$G_c(z) = K_p \tag{2.7}$$

2.5 Integral Control

The main function of the integral action is to make sure that the process output agrees with the setpoint in steady state. With proportional control, there is normally a control error in steady state. With integral action, a small positive error will always lead to an increasing control signal, and a negative error will give a decreasing control signal no matter how small the error is [Aström, Hagglund, 1995].

For continuous-time systems, the integral control is described from time t_0 to time t as:

$$u(t) = \frac{K_p}{T_i} \int_{t_0}^{t} e(t)dt \tag{2.8}$$

where the integral controller in transfer function form is:

$$G_c(s) = \frac{K_p}{T_i s} \tag{2.9}$$

The constant T_i is called the integration time constant or reset in Eqs. (2.8)-(2.9). In the case of discrete-time systems, the integral equation can be approximated by the difference equation [Paraskevopoulos, 1996]:

$$\frac{u(k) - u(k-1)}{T} = \frac{K_p}{T_i} e(k) \tag{2.10}$$

where T is the sampling instant, $u(k\text{-}1)$ is the control input applied at a sample before.

The present control input and controller transfer function are defined as:

$$u(k) = u(k-1) + \frac{K_p T}{T_i} e(k) \tag{2.11}$$

$$G_c(z) = \frac{K_p T}{T_i (1-z)^{-1}} = \frac{K_p T z}{T_i (z-1)} \tag{2.12}$$

2.6 Derivative Control

The purpose of the derivative action is to improve the closed-loop stability and performance. Because of the process dynamics, it will take some time before a change in the control variable is noticeable in the process output. Thus, the control system will be late in correcting for an error [Aström, Hagglund, 1995].

For continuous-time systems, the derivative control and controller transfer function is described as:

$$u(t) = K_p T_d \frac{de(t)}{dt} \tag{2.13}$$

$$G_c(s) = K_p T_d s \tag{2.14}$$

where the constant T_d is called the derivative or rate time constant. In the case of discrete-time systems, the differential equation can be approximated by the difference equation [Paraskevopoulos, 1996]:

$$u(k) = K_p T_d \left(\frac{e(k) - e(k-1)}{T} \right) \tag{2.15}$$

The discrete time derivative controller is given as:

$$G_c(z) = K_p T_d \left(\frac{1 - z^{-1}}{T} \right) = \frac{K_p T_d}{T} \left(\frac{z-1}{z} \right) \tag{2.16}$$

2.7 Proportional + Integral (PI) Control

PI control is the sum of proportional and integral control. This combination provides stability with elimination of offset, making it the most common controller used in the fluid-processing industries. It is used almost universally, even in those applications where other controllers are better suited. The mathematical representations of a PI control and controller transfer function are continuous-time is:

$$u(t) = K_p \left(e(t) + \frac{1}{T_i} \int_{t_0}^{t} e(t) dt \right) \tag{2.17}$$

$$G_c(s) = K_p \left(1 + \frac{1}{T_i s} \right) \tag{2.18}$$

The discrete-time representations of the control input and controller are:

$$u(k) = K_p\left(e(k) + \frac{T}{T_i}\sum_{i=0}^{k-1} e(i)\right) \tag{2.19}$$

$$G_c(z) = K_p\left(1 + \frac{T}{T_i}\left(\frac{z}{z-1}\right)\right) \tag{2.20}$$

2.8 Proportional + Integral + Derivative (PID) Control

Combining proportional, integral and derivative control actions, produces classical PID controller which finds extensive application in industrial control. For the continuous-time case, the controller in its basic form is described as [Paraskevopoulos, 1996]:

$$u(t) = K_p\left(e(t) + \frac{1}{T_i}\int_{t_0}^{t} e(t)dt + T_d\frac{de(t)}{dt}\right) \tag{2.21}$$

The controller transfer function is:

$$G_c(s) = K_p\left(1 + \frac{1}{T_i s} + T_d s\right) \tag{2.22}$$

In the case of discrete-time systems, the PID control input and controller can be described in its simplest form by the difference equation:

$$u(k) = K_p\left(e(k) + \frac{T}{T_i}\sum_{i=0}^{k-1} e(i) + \frac{T_d}{T}\left[e(k) - e(k-1)\right]\right) \tag{2.23}$$

$$G_c(z) = K_p\left[1 + \frac{T}{T_i}\left(\frac{z}{z-1}\right) + \frac{T_d}{T}\left(\frac{z-1}{z}\right)\right] \tag{2.24}$$

The discrete PID controller can be simplified as:

$$G_c(z) = K\left(\frac{z^2 - az + b}{z(z-1)}\right) \tag{2.25}$$

where the constant coefficients, *K*, *a* and *b* are:

$$K = K_p\left(\frac{TT_i + T_dT_i + T^2}{T_iT}\right)$$

$$a = \frac{T_iT - T_dT_i}{TT_i + T_dT_i + T^2}$$

$$b = \frac{T_dT_i}{TT_i + T_dT_i + T^2}$$

3 SELF-TUNING CONTROL

The aim of self-tuning systems is to automate in some way certain activities of the control system and signal processing engineer. The principal tasks involved in control system and signal processing engineering consist of the following as summarized in Fig. 3.1 [Wellstead, Zarrop, 1991]:

1. modelling of a system,
2. design of a controller,
3. implementation of the controller.

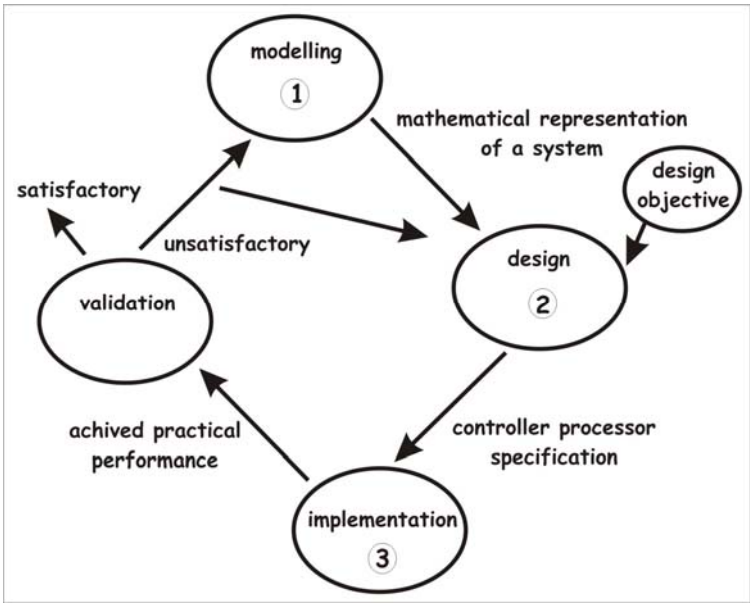


Figure 3.1 The three stages of control system.

The mathematical model is constructed in stage 1 and the controller is designed in the second stage, which the design is based on the developed model. In stage 3, the specification is then implemented and validated against the design objective. The validation phase of this sequence is vital to success and is one of the

most important arguments in favour of self-tuning. To be specific, in conventional off-line design the validation phase often proves unsatisfactory. This leads to a time consuming repetition of he whole sequence of modeling, design and implementation. A central advantage of self-tuning is that the sequence is performed on-line, usually in real time, in such a manner that the validation process is much faster [Wellstead, Zarrop, 1991].

The three stages of modeling, design and implementation in self-tuning are shown again in Fig. 3.2. The structure is associated with the system identifier, control synthesizer and controller blocks of a self-tuning controller. Note that the validation stage is not shown explicitly, since it is associated with a qualitative assessment of performance. It is not an algorithmic element of the self-tuning system [Wellstead, Zarrop, 1991].

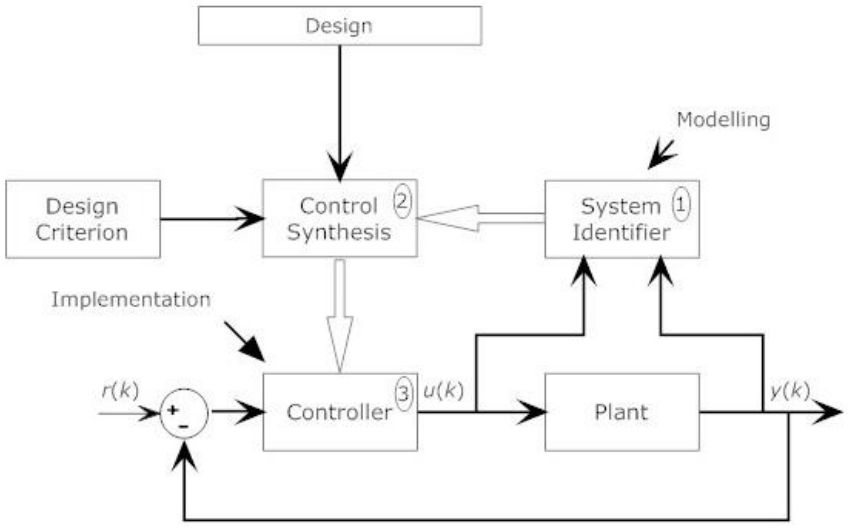


Figure 3.2 Self-tuning controller structure

3.1 System Identification for Self-Tuning

In adaptive control the task of identification is just as important as the role of control synthesis. Identification for adaptive control has, of course, its own specification, which for most cases involves estimation of parameters using the least squares method [Bobal, Böhm, Fessl, Machacek, 2005].

3.1.1 Least Squares Algorithm

The construction which is used in the self-tuning control system consists of generally discrete time transfer function with undetermined parameters. Control system determines these parameters using input output relation of physical systems. Least Squares identification is the most familiar method to determine the parameters of a system.

Transfer function model of a system with control input sequence $u(k)$ and with output $y(k)$ subject to disturbances from a measurable source $v(k)$, drift $\mathcal{D}(k)$ and random noise $e(k)$ [Wellstead, Zarrop, 1991]:

$$Ay(k) = Bu(k-1) + Dv(k) + \mathcal{D}(k) + Ce(k) \tag{3.1}$$

where the polynomials A, B, C, D are defined as:

$$A = 1 + a_1 z^{-1} + \cdots + a_n z^{-n_a}$$

$$B = b_0 + b_1 z^{-1} + \cdots + b_n z^{-n_b}$$

$$C = 1 + c_1 z^{-1} + \cdots + c_{n_c} z^{-n_c}$$

$$D = d_0 + d_1 z^{-1} + \cdots + d_n z^{-n_d}$$

The system defined in Eq. (3.1) is shown in Fig. 3.1.

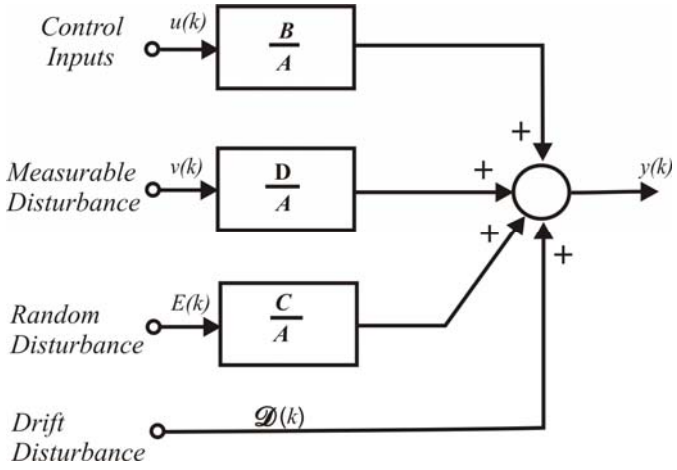


Figure 3.3 Transfer function of a system

When the system to be controlled is known then the coefficients of polynomials *A, B, C, D* should be calculated from continuous system parameters. If the system is unknown, however, then the polynomial coefficients are treated as parameters to be determined by measurement or estimation using [Wellstead, Zarrop, 1991]:

$$y(k) = x^T(k)\theta(k-1) + e(k) \tag{3.2}$$

where $\theta(k-1)$ is the vector of unknown parameters and $x(k)$ is a regression vector as:

$$\theta^T = [-a_1,...,-a_{n_a}, b_0,...,b_{n_b}, d_0,...,d_{n_d}, d_0,...,d_{n_d}, c_1,...,c_{n_c}] \tag{3.3}$$

$$\begin{aligned} x^T(k) = [&y(k-1),...y(k-n_a), u(k-1),...u(k-n_b-1), \\ &v(k),...v(k-n_d),1,k,...,k^{n_d}, e(k-1), e(k-2,...e(k-n_c))] \end{aligned} \tag{3.4}$$

The regression vector contains the values of $e(k\text{-}1)$, $e(k\text{-}2),\ldots,e(k\text{-}n_c)$ which in general will be unknown, since they are the past value of the unobservable white noise disturbance $e(k)$.

Assume a model of the system to determine data vector θ of true system parameters:

$$y(k) = x^T(k)\hat{\theta}(k-1) + \hat{e}(k) \tag{3.5}$$

where $\hat{\theta}$ is a vector of adjustable model parameters and $\hat{e}(k)$ is the corresponding modeling error. In the rest of the work $(k\text{-}1)$ in $\hat{\theta}$ is omitted for simplicity. The purpose is to select $\hat{\theta}$ so that overall modeling error is minimized in some sense:

$$\hat{e}(k) = e(k) + x^T(k)(\theta - \hat{\theta}) \tag{3.6}$$

so that $\hat{e}(k)$ depends on $\hat{\theta}$ and, in some cases, the 'minimized' modeling errors will be equal to the white noise sequence corrupting the system output data.

Assume that the system given in Eq. (3.2) has been running for sufficient time to form N consecutive data vectors. The data obtained in this way allows the model Eq. (3.5) to be expressed in the vector/matrix form:

$$\begin{bmatrix} y(1) \\ y(2) \\ \vdots \\ \vdots \\ y(N) \end{bmatrix} = \begin{bmatrix} x^T(1) \\ x^T(2) \\ \vdots \\ \vdots \\ x^T(N) \end{bmatrix} \hat{\theta} + \begin{bmatrix} \hat{e}(1) \\ \hat{e}(2) \\ \vdots \\ \vdots \\ \hat{e}(N) \end{bmatrix} \tag{3.7}$$

To be able to estimate the parameters uniquely the number N of equations in Equation (3.7) must not be less than m, where m is the number of unknown parameters in the vector θ. In the noise-free ease ($e(k)$ =0), the equation can *be* solved as a set of linear equations in $N=\boldsymbol{m}$ unknowns, where $m=n_a+(n_b+1)+(n_d+1)+(n_d+1)$ (assuming $n_c=0$). The resulting modeling errors are identically zero. When noise is present (and, in practice, even in nominally noise-free systems) N is much larger than m and an alternative procedure is used to reduce estimation errors induced by the noise. The technique most widely used in this connection is *linear least squares* [Wellstead, Zarrop, 1991].

Equation (3.7) is re-written as:

$$y = X\hat{\theta} + \hat{e} \tag{3.8}$$

where

$$y^T = [y(1),..., y(N)]$$
$$\hat{e}^T = [\hat{e}(1),..., \hat{e}(N)]$$

$$X = \begin{bmatrix} x^T(1) \\ x^T(2) \\ \vdots \\ \vdots \\ x^T(N) \end{bmatrix}$$

Rearranging Eq. (3.8) in terms of the error vector $\hat{e}$:

$$\hat{e} = y - X\hat{\theta} \tag{3.9}$$

A cost index J is selected which minimizes the sum of squares of errors as:

$$J = \sum_{t=1}^{N} \hat{e}^2(t) = \hat{e}^T \hat{e} \tag{3.10}$$

To find the least squares estimate, one can rewrite Eq. (3.10) in terms of the data vectors and parameter vector as:

$$\begin{aligned} J &= (y - X\hat{\theta})^T (y - X\hat{\theta}) \\ &= y^T y - \hat{\theta}^T X^T y - y^T X\hat{\theta} + \hat{\theta}^T X^T X\hat{\theta} \end{aligned} \tag{3.11}$$

For the cost index J to be minimized the derivative of J with respect to $\hat{\theta}$ is set to zero:

$$\frac{\partial J}{\partial \hat{\theta}} = -2X^T y + 2X^T X\hat{\theta} = 0 \tag{3.12}$$

Equation (3.12) yields:

$$X^T X\hat{e} = X^T y \tag{3.13}$$

If the second derivative of J is positive definite, unique solution is obtained as:

$$\frac{\partial J^2}{\partial \hat{\theta}^2} = 2(X^T X) \tag{3.14}$$

Hence, the least squares estimator for the parameter vector is obtained

$$\hat{\theta} = [X^T X]^{-1}[X^T y] \tag{3.15}$$

The resulting modeling error $\hat{e}$ is denoted by

$$\eta^T = [\eta(1),...,\eta(N)] \tag{3.16}$$

whose components are called *residuals* [Wellstead, Zarrop, 1991].

3.1.2 Recursive Least Squares Algorithm

Equation (3.15) cannot be used to calculate the parameter estimates of the process model for self-tuning controllers, since it represents one-shot parameter calculation. It is necessary to use its recursive version, which performs the identification in real time. Here, newly measured values are only used to correct the original estimates. This reduces the complexity of the calculation and thus the demands placed on the computer technology used. Recursive algorithms allow one to monitor changes in the characteristics (parameters) of the process in real time and therefore form the basis for self-tuning controllers [Bobal, Böhm, Fessl, Machacek, 2005].

In recursive estimation, a supposed model from the previous sampling period $\hat{\theta}(k-1)$ is used for assessment of $\hat{y}(k)$ system output in the given sampling period. Estimated system output is compared with the real system output $y(k)$ and on the basis of the obtained difference an error signal $\varepsilon(k)$ is generated. Now, so-called mechanism of updating, on the basis of error signal, correct values of supposed parameters of the system $\hat{\theta}(k-1)$ on $\hat{\theta}(k)$. Scheme of the recursive estimation is

shown in Fig. 3.4 [Wellstead, Zarrop, 1991]. The covariance matrix $P(k)$ is defined as:

$$P(t) = [X^T(k)X(k)]^{-1} \tag{3.17}$$

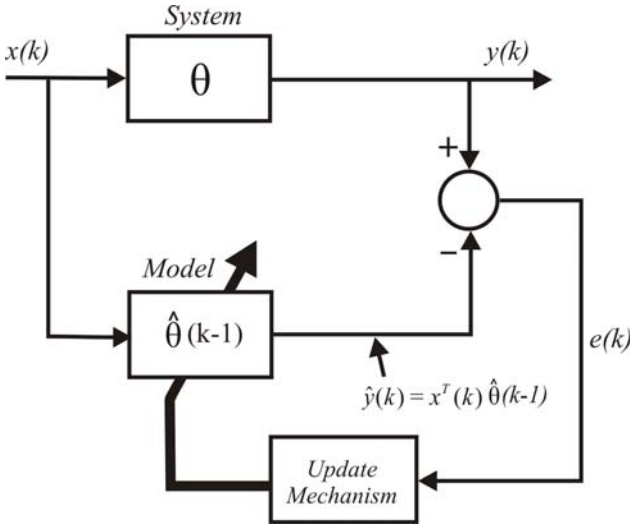


Figure 3.4 Scheme of recursive least squares method

$$P(k+1) = P(k)\left[I_m - \frac{x(k+1)x^T(k+1)P(k)}{1+x^T(k+1)P(k)x(k+1)} \right] \tag{3.18}$$

The modeling error is obtained as:

$$\varepsilon(k+1) = y(k+1) - x^T(k+1)\hat{\theta}(k) \tag{3.19}$$

After simple mathematical transformation a vector of assessed unknown parameters is obtained in the form of

$$\hat{\theta}(k+1) = \hat{\theta}(k) + P(k+1)x(k+1)\varepsilon(k+1) \tag{3.20}$$

Application of Recursive Least Squares (RLS) method demands supposition of starting values of system parameters $\hat{\theta}(0)$ and covariance matrix, $p(0)$.

To choose starting value of the parameters there are several recommendations [Wellstead, Zarrop, 1991]:

i) Existence of previous knowledge about the system like the possibility of placing suitable mathematical model which suggest the choice of starting system parameters.

ii) The technique which is useful when there is no other primary information about the system is using a supposition that the system is integrator of the first order with unity gain. In this way the starting values of the system parameters should be chosen in the form of

$$a_1 = -1, \quad b_0 = T, \quad a_i = 0 \ (i \neq 1), \quad b_i = 0 \ (i \neq 1) \tag{3.21}$$

Generally looking, the choice of the starting values of the system parameters is not crucial for achieving stationary values of the system parameters, but is important for their convergence. Still, the error in estimation of parameters depends on these accepted values [Krneta, Antic, Stojanovic, 2005].

The choice of the starting value of covariance matrix, $P(k)$, also presents an important step when applying RLS algorithm. If there is no primary knowledge about the system, the large value of covariance matrix $P(0)$ is recommended. However, if the starting values similar to exact values of the system parameters can be accepted, then it is enough to choose the little starting value of the covariance matrix [Krneta, Antic, Stojanovic, 2005].

In order to show the influence of the starting value of covariance matrix the following expression can be looked at:

$$P(k) = \left[P^{-1}(0) + \sum_{i=1}^{t} x(i)x^{T}(i) \right]^{-1} \tag{3.22}$$

It can be seen from Eq. (3.22) that the choice of a large value of $P(0)$ much less influences on $P(k)$ than the data value x(i). Vice versa, $P(0)$ if has a small value its influence is much greater. A standard choice of the matrix $P(0)$ can be presented in the form:

$$P(0) = rI_m, \tag{3.23}$$

where I_m is a unit matrix of the order m, and m is total number of the system parameters. The constant r is usually chosen in the range of 100-10000 for the large value of covariance matrix $P(0)$, while it is chosen to be 1-10 for the small starting value of the covariance matrix, $P(0)$ [Krneta, Antic, Stojanovic, 2005].

Steps in Recursive Least Squares algorithm:

[Step **1**] Form $x(k+1)$ using the new data.

[Step **2**] Form $\varepsilon(k+1)$ from Eq. 3.19.

[Step **3**] Form $P(k+1)$ from Eq. 3.18

[Step **4**] Update $\hat{\theta}(k+1)$ from Eq. 3.20

[Step **5**] Wait for next time step to elapse and loop back to [Step **1**]

3.2 Controller Tuning Methods

3.2.1 Ziegler-Nichols Tuning Methods

The Ziegler-Nichols tuning rules were developed to give closed loop systems with good attenuation of load disturbances [Siemens 2005]. The design criterion was quarter amplitude decay ratio, which means that the amplitude of an oscillation

should be reduced by a factor of four over a whole period of response [Yücelen, Kaymakçı, Kurtulan, 2001].

3.2.1.1 Ziegler-Nichols Process Reaction Method

This is the earliest design method for the PID controller. It was originally developed in 1942, so its no exactly state-of-the-art [Goodwin, Salgado, 2000]. Process reaction method is an experimental open-loop tuning method and is only applicable to open-loop stable systems. This method can be viewed as a traditional method based on modeling and control and is not suitable for use with plants which incorporate integral term (causes to ramp) [Yücelen, Kaymakçı, Kurtulan, 2001].

A linearized quantitative version of a simple plant can be obtained with an open loop experiment, using the following procedure:

1- With the plant in open loop, take the plant manually to a normal operating point. Say that the plant output settles at $y(t) = y_0$ for a constant plant input $u(t) = u_0$.
2- At an initial time, t_0, apply a step change to the plant input, from u_0 to u_∞ (this should be in the range of 10 to 20% of full scale) as shown in Fig. 3.5.
3- Record the plant output until it settles to the new operating point. Draw a tangential line curve at the inflection point of the response shown below. This curve is known as the *process reaction curve*. (m.s.t. stands for maximum slope tangent.), where t_0 is the step input application time, t_1 is the time t_0 that tangent line crosses the time axis and t_2 is the time t_0 that the tangent line crosses the steady-state output line.
4- Compute the model parameters and the transfer function as [Yaacob, Mohamed, 1998]:

$$K_0 = \frac{y_\infty - y_0}{u_\infty - u_0}; \qquad \tau_0 = t_1 - t_0; \qquad v_0 = t_2 - t_1 \tag{3.24}$$

$$G_0(s) = \frac{K_0 e^{-s\tau}}{v_0 s + 1} \qquad x \overset{\Delta}{=} \frac{\tau}{v_0} \tag{3.25}$$

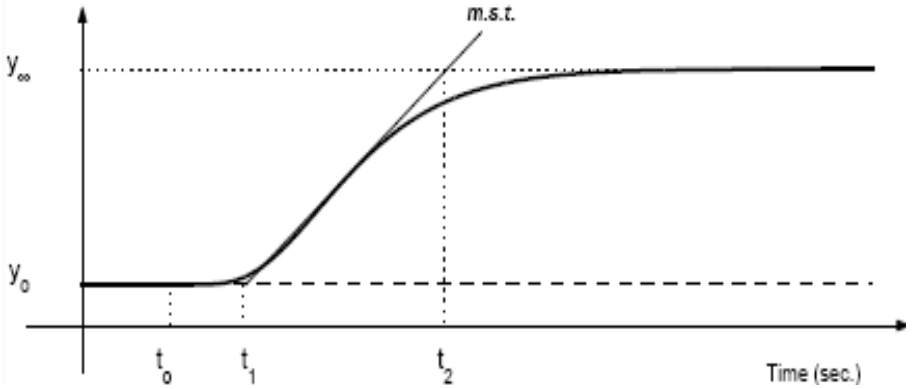


Figure 3.5 Z-N process reaction method.

Table 3.1 PID controller parameters obtained from the Z-N PRC method.

Controller	K_p	T_i	T_d
P	$\frac{v_0}{K_0 \tau_0}$	-	-
PI	$\frac{0.9 v_0}{K_0 \tau_0}$	$3\tau_0$	-
PID	$\frac{1.2 v_0}{K_0 \tau_0}$	$2\tau_0$	$0.5\tau_0$

3.2.1.2 Ziegler-Nichols oscillation method

In the oscillation method the key idea is to determine the point where the Nyquist curve of the open loop system intersects the negative real axis. This is achieved by increasing the gain of a proportional controller until the closed loop system reaches

the stability limit. The gain and the corresponding period are then determined and the PID coefficients are then found.

This procedure is carried out through the following steps:

1- Set the true plant under proportional control, with a very small gain, K_p.
2- Apply step set point change and observe the output response. If the output response is not in sustained oscillation mode, increase proportional gain K_p and repeat the procedure up to sustained oscillation is obtained at the output.
3- Record the controller critical gain K_p that causes sustained oscillation at the output and the oscillation period of the controller output, T_c
4- Adjust the controller parameters according to the rules presented in Table 3.2 [Yaacob, Mohamed, 1998].

Table 3.2 PID controller parameters obtained from the Z-N oscillation method.

Controller	K_p	T_i	T_d
P	$0.50K_c$	-	-
PI	$0.45K_c$	$\frac{T_c}{1.2}$	-
PID	$0.60K_c$	$0.5T_c$	$\frac{T_c}{8}$

3.2.2 Cohen-Coon Reaction Curve Method

Cohen and Coon carried out further studies to find controller settings which, based on the same model, lead to a weaker dependence on the ratio of delay to time constant. The model parameters defined in Section 3.2.1.1 are used to calculate the controller setting using the rules given in Table 3.3.

Table 3.3 PID controller parameters obtained from the Cohen-Coon method.

Controller	K_p	T_i	T_d
P	$\frac{v_0}{K_0\tau_0}\left[1+\frac{\tau_0}{3v_0}\right]$	-	-
PI	$\frac{v_0}{K_0\tau_0}\left[0.9+\frac{\tau_0}{12v_0}\right]$	$\frac{\tau_0[30v_0+3\tau_0]}{9v_0+20\tau_0}$	-
PID	$\frac{v_0}{K_0\tau_0}\left[\frac{4}{3}+\frac{\tau_0}{4v_0}\right]$	$\frac{\tau_0[32v_0+6\tau_0]}{13v_0+8\tau_0}$	$\frac{4v_0\tau_0}{11v_0+2\tau_0}$

3.2.3 Chien-Hrones-Reswick Method

Chien-Hrones-Reswich (CHR) method is the modified version of the Ziegler-Nichols method. This method was developed in 1952 by Chien-Hrones-Reswich provides a better way to select a compensator for process control applications. The controller settings are presented in Table 3.4.

Table 3.4 Chien-Hrones-Reswick method with 0% overshoot.

0% overshoot			
Controller	K_p	T_i	T_d
P	$\frac{0.3v_0}{K_0\tau_0}$	-	-
PI	$\frac{0.35v_0}{K_0\tau_0}$	$1.2v_0$	-
PID	$\frac{0.6v_0}{K_0\tau_0}$	v_0	$0.5\tau_0$

Table 3.5 Chien-Hrones-Reswick method with 20% overshoot.

20% overshoot			
Controller	K_p	T_i	T_d
P	$\frac{0.7v_0}{K_0\tau_0}$	-	-
PI	$\frac{0.6v_0}{K_0\tau_0}$	v_0	-
PID	$\frac{0.95v_0}{K_0\tau_0}$	$1.4v_0$	$0.47\tau_0$

3.2.4 Pole Assignment Self-Tuning Control

The aim of pole assignment control is to exactly match the closed loop characteristic equation of a feedback system to some desired form. This is used in controller design where the performance criterion for the control system can be expressed in the classical control terms of frequency response or transient response [Wellstead, Zarrop, 1991]. Diagram of pole-assignment control is shown in Fig. 3.6.

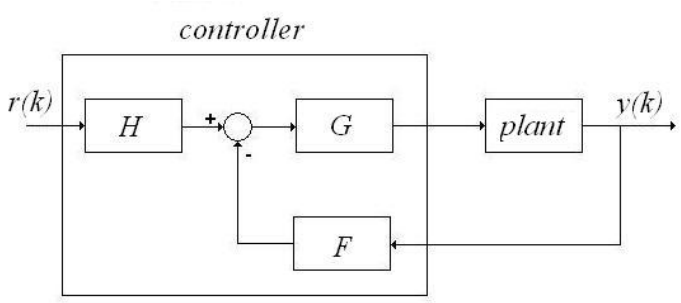


Figure 3.6 Design of a feedback controller.

The design of a feedback controller has two main aims. The first is to modify in some way the dynamic response of a system. The second is to reduce the sensitivity of a system output to disturbances. Additionally, and linked to these aims,

is the further objective of reducing the overall sensitivity of the closed system to parameter variations. For example, the controller polynomials *F, G, H* are to be designed so as to ensure that the system output *y*(*k*) tracks changes in the reference signal *r*(*k*) in an acceptably fast way. In addition, it is required that in the steady state (when *r*(*k*) is constant) the output *y*(*k*) is equal to the reference set point.

3.2.4.1 Pole Assignment Control

In general, the control objective for a system requires the output y(*k*) to follow a reference signal *r*(*k*) in some predetermined way and to reject random disturbances which may corrupt the output.

Consider a plant defined by the equation:

$$Ay(k) = Bu(k\text{-}1) + Ce(k) \tag{3.26}$$

The control law is in the form as [Wellstead, Zarrop, 1991]:

$$Fu(k) = Hr(k) - Gy(k) \tag{3.27}$$

Combining Eq. (3.26) and Eq. (3.27) yields the closed loop to be:

$$(FA + z^{-1}BG)y(k) = z^{-1}BHr(k) + CFe(k) \tag{3.28}$$

The closed loop poles are then assigned to their desired locations, specified by **T** a chosen polynomial, by selecting *F* and *G* according to the polynomial identity [Wellstead, Zarrop, 1991]:

$$FA + z^{-1}BG = \mathbf{T}C \tag{3.29}$$

where the polynomials *F, G, H* are:

$$\left.\begin{aligned} F &= 1 + f_1 z^{-1} + \ldots + f_{n_f} z^{-n_f} && n_f = n_b \\ G &= g_0 + g_1 z^{-1} + \ldots + g_{n_g} z^{-n_g} && n_g = n_a - 1 \quad (n_a \neq 0) \\ H &= h_0 + h_1 z^{-1} + \ldots + h_{n_h} z^{-n_h} \\ T &= 1 + t_1 z^{-1} + \cdots + t_{n_t} z^{-n_t} && n_t \leq n_a + n_b - n_c \end{aligned}\right\} \tag{3.30}$$

It is assumed that the polynomials *A* and *B* have no common zeroes.

$$y(k) = \frac{HB}{TC} r(k-1) + \frac{F}{T} e(k) \tag{3.31}$$

where the noise polynomial *C* has been cancelled in the disturbance term. Note that this requires that *C* is inverse stable, a weak requirement.

The precompensator (polynomial *H*) is selected to achieve both low frequency gain matching and the cancellation of *C* from the servo pole set. The simplest choice is

$$H = C\left[\frac{T}{B}\right]_{z=1} \tag{3.32}$$

yielding the closed loop equation to be:

$$y(k) = \left[\frac{T}{B}\right]_{z=1}\left[\frac{B}{T}\right] r(k-1) + \frac{F}{T} e(k) \tag{3.33}$$

Certain performance requirements are fed into the synthesis block. These consist of following controller design information:

(a) the desired closed loop pole set, specified by the polynomial **T**

(b) the form of the controller, e.g. whether it is a servo system, a regulator or a combination of both; and

(c) whether the controller is to be incremental or nonincremental in nature.

In addition, further information must be supplied concerning the configuration requirements of the self-tuner, including;

(d) the sample rate to be used;

(e) the degrees of the system model polynomials n_a, n_b, n_c, etc.;

(f) the delay "k" in the system, if known

Provided with this information, the self-tuning system can be set up to go through the following cycle of adaptation as shown in Fig. 3.7:

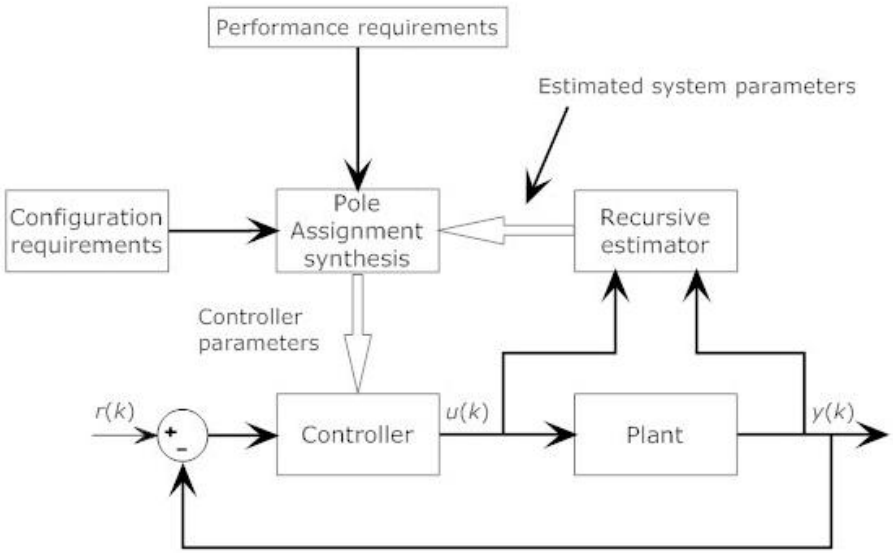


Figure 3.7 Self-tuning pole assignment system.

At each sample interval *t* the following sequence of action is taken:

Step (i) Data capture

The system output $y(k)$, reference signal $r(k)$ and control input $u(k)$ are measured:

Step (ii) Estimator update

The data acquired in (i) is used together with past data and the previous control signal to update the parameter estimates in a model of the system using an appropriate recursive estimator.

Step (iii) Controller synthesis

The updated parameters from (ii) are used in a pole assignment identity to synthesize the parameters of the desired controller.

Step (iv) Control calculation ($u(k)$)

The controller parameters synthesized in (iii) are used in a controller to calculate and input the next control signal $u(k)$.

At the end of the cycle the control computer waits until the end of sample interval *k* and then repeats the cycle for interval *k*+1, and so on. The steps in the self-tuning cycle are computed sequentially. The timing sequence is shown in Fig. 3.8 for self-tuning control. Below figure illustrates this in terms of a timing and sequence diagram. The total computation time must be less than the sample interval and is generally assumed to be much less. In any event the computation time introduces an additional time delay into the control loop. In view of the problems introduced by partial time delays, some users delay outputting the controller signal $u(k)$ until the end of the sample interval in which it was calculated. This ensures that the additional delay is one complete sample interval [Wellstead, Zarrop, 1991].

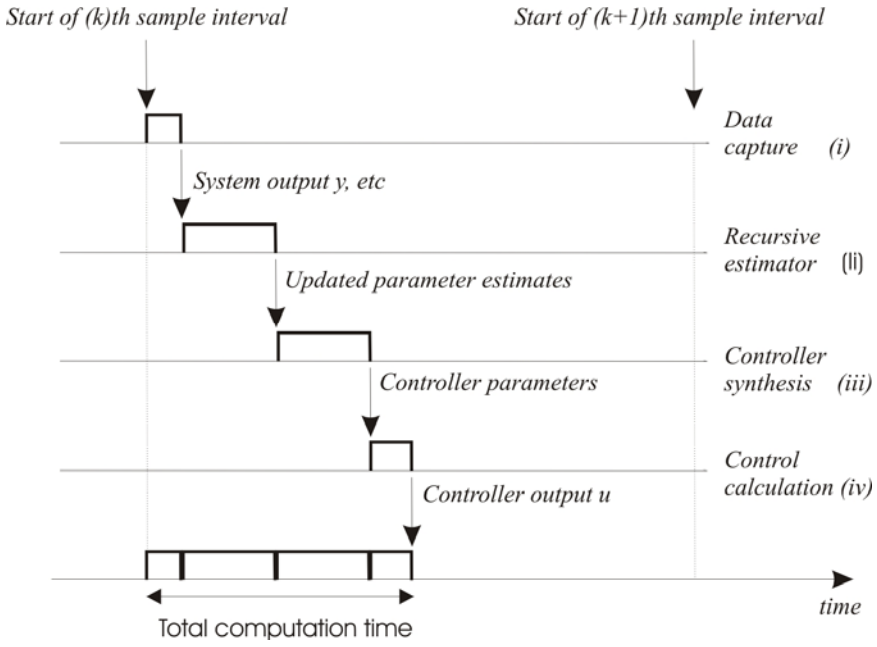


Figure 3.8 Timing and sequence diagram for self-tuning.

4 POLE ASSIGNMENT SELF-TUNING PID CONTROL

Some researchers have suggested an implicit pole-assignment self-tuning algorithm, in which the controller parameters are estimated directly without the need to solve a polynomial identity**.** However, Wellstead et al. [Wellstead, Zarrop, 1991] argue that in simulation, the computational savings achieved by using implicit algorithms are often offset by slow convergence phase systems if the closed loop poles are not selected properly. The objectives of the pole-assignment controller are that the controller should not cancel the zeros of the plant, and only the poles of the plant are to be placed to the desired location specified by the tailoring polynomial [Kirecci, Eker, Dulger, 2003].

4.1 Plant Model

Consider the SISO system described by the **A**uto **R**egressive **M**oving **A**verage with **X**egenous input (ARMAX) model:

$$A(z^{-1})y(k) = z^{-d}B(z^{-1})u(k) + C(z^{-1})e(k) \tag{4.1}$$

where $e(k)$ is the uncorrelated zero mean white noise sequence. The integer d represents the time delay between the input and output of the plant. The backward shift operator is defined by $z^{-1}y(k) = y(k-1)$, and a differential operator defined by $\Delta = 1 - z^{-1}$. The plant polynomials $A(z^{-1}), B(z^{-1})$ and $C(z^{-1})$ are defined:

$$\left.\begin{aligned} A &= 1 + a_1 z^{-1} + \cdots + a_{n_a} z^{-n_a} \\ B &= b_0 + b_1 z^{-1} + \cdots + b_{n_b} z^{-n_b} \\ C &= 1 + c_1 z^{-1} + \cdots + c_{n_c} z^{-n_c} \end{aligned}\right\} \tag{4.2}$$

where n_a, n_b, n_c represent orders of the polynomials; therefore the total number of model parameters is obtained as $n_T = n_a + n_b + 1 + n_c$ [Kara, T. and Eker, İ. 2003].

ARMAX model is not entirely suitable for adaptive control, if its parameters (coefficients of the polynomials *A, B, C*) are the subject of identification using measured data. There is a problem of identifying coefficients of the polynomial $C(z^{-1})$ because the fictitious noise $e(k)$ cannot be measured. Although there are identification procedures (Extended Least Squares Method) enabling $C(z^{-1})$ to be identified, their convergence is not guaranteed generally and usually is too slow. Therefore most adaptive controller designs are based on the regression **A**uto **R**egressive with **X**egenous input (ARX) model as shown in Fig. 4.1 [Bobal, Böhm, Fessl, Machacek, 2005]:

$$y(k) = -\sum_{i=1}^{n_a} a_i y(k-i) + \sum_{i=1}^{n_b} b_i u(k-i) + e(k) \tag{4.3}$$

or

$$A(z^{-1})y(t) = z^{-d} B(z^{-1})u(t) + e(t) \tag{4.4}$$

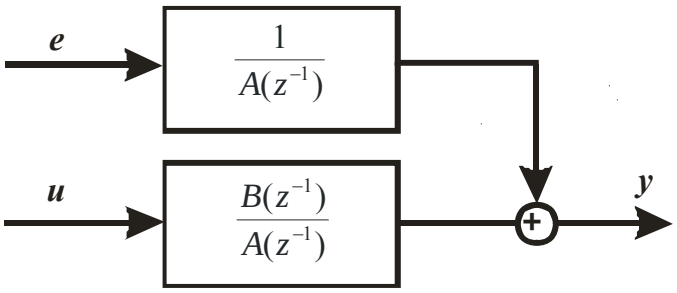


Figure 4.1 Block diagram of ARX model

Assumptions:

1. The degrees n_a and n_b are known.
2. Parameters a_i and b_i are unknown.

3. The time delay of the system t_d may be unknown, but the following relation is satisfied:

$$0 \le d \le t_d \le d + n_b \tag{4.5}$$

4. There exists a partial feedback gain $k_c(\ge 0)$ such that $A'(z^{-1})$ is asymptotically stable where

$$A'(z^{-1}) = A(z^{-1}) + z^{-(d+1)} k_c B(z^{-1}) \tag{4.6}$$

One can set $k_c=0$ in the case where $A(z^{-1})$ *is a Hurwitz polynomial (stable polynomial).*

5. Reference $\{r(k)\}$ is given by piecewise constants.
6. There exists a positive definite matrix R that satisfies:

$$R = \lim_{N\to\infty} \frac{1}{N} \sum_{k=1}^{N} X(k-1) X^T(k-1) \tag{4.7}$$

This assumption is so-called persistently spanning condition, which is required to guarantee the convergence of identified parameters [Yamamato, Kaneda, Tanaka, 1995].

4.2 Controller Design

Under above assumptions, output of a self-tuning pole assignment control is described as:

$$y(k) = X^T(k-1)\hat{\theta} + \hat{e}(k) \tag{4.8}$$

where the data vector X and estimated parameter vector $\hat{\theta}$ are:

$$X^T(k-1) = \left[-y(k-1), -y(k-2), \cdots, -y(k-n_a), u(k-1), u(k-2), \cdots, u(k-1-n_b)\right] \tag{4.9}$$

$$\hat{\theta}^T = [-\hat{a}_1, \hat{a}_2, ..., -\hat{a}_{n_a}, \hat{b}_0, \hat{b}_1, ..., \hat{b}_{n_b}] \tag{4.10}$$

The general closed loop block diagram is shown in Fig. 4.2, where the plant and controller are respectively given:

$$\left.\begin{aligned} G_P(z) &= \frac{y(z)}{u(z)} = \frac{z^{-d}\hat{B}(z^{-1})}{\hat{A}(z^{-1})} \\ G_R(z) &= \frac{u(z)}{e(z)} = \frac{\hat{G}(z^{-1})}{\hat{F}(z^{-1})} \end{aligned}\right\} \tag{4.11}$$

The polynomials are in the form of:

$$\hat{A} = 1 + \hat{a}_1 z^{-1} + \cdots + \hat{a}_{n_a} z^{-n_a}$$

$$\hat{B} = \hat{b}_0 + \hat{b}_1 z^{-1} + \cdots + \hat{b}_{n_b} z^{-n_b}$$

$$\hat{F} = 1 + \hat{f}_1 z^{-1} + \ldots + \hat{f}_{n_f} z^{-n_f}$$

$$\hat{G} = \hat{g}_0 + \hat{g}_1 z^{-1} + \ldots + \hat{g}_{n_g} z^{-n_g}$$

where $\hat{a}_i, \hat{b}_i, \hat{f}_i, \hat{g}_i$ are real constant coefficients to be estimated on-line.

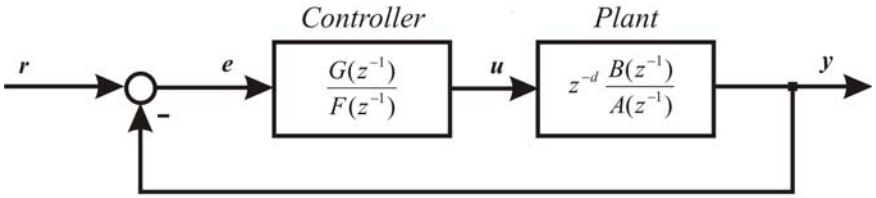


Figure 4.2 Block diagram of control loop.

The closed-loop transfer function is:

$$G_W(z) = \frac{z^{-d}\hat{B}\hat{G}}{\hat{A}\hat{F} + z^{-d}\hat{B}\hat{G}} \tag{4.12}$$

The desired closed loop pole set is defined by the zeroes of

$$T = 1 + t_1 z^{-1} + \cdots + t_{n_t} z^{-n_t} \tag{4.13}$$

Then the controller coefficients which assign the actual pole set to the desired set are given by the solution of the polynomial given in Eq. (4.13) as:

$$T = \hat{A}\hat{F} + z^{-d}\hat{B}\hat{G} \tag{4.14}$$

where the orders are

$$\left.\begin{aligned} n_f &= n_b \\ n_g &= n_a - 1 \quad (n_a \neq 0) \\ n_t &\leq n_a + n_b \end{aligned}\right\} \tag{4.15}$$

As an example, Eq. (4.14) becomes with $n_a = 3, n_b = 2, n_t = 1$:

$$\begin{aligned} &(1 + \hat{f}_1 z^{-1} + \hat{f}_2 z^{-2})(1 + \hat{a}_1 z^{-1} + \hat{a}_2 z^{-2} + \hat{a}_3 z^{-3}) \\ &+ z^{-1}(\hat{b}_0 + \hat{b}_1 z^{-1} + \hat{b}_2 z^{-2})(\hat{g}_0 + \hat{g}_1 z^{-1} + \hat{g}_2 z^{-2}) = 1 + t_1 z^{-1} \end{aligned} \tag{4.16}$$

It can be represented in matrix form as:

$$\begin{bmatrix} 1 & 0 & \hat{b}_0 & 0 & 0 \\ \hat{a}_1 & 1 & \hat{b}_1 & \hat{b}_0 & 0 \\ \hat{a}_2 & \hat{a}_1 & \hat{b}_2 & \hat{b}_1 & \hat{b}_0 \\ \hat{a}_3 & \hat{a}_2 & 0 & \hat{b}_2 & \hat{b}_1 \\ 0 & \hat{a}_3 & 0 & 0 & \hat{b}_2 \end{bmatrix} \begin{bmatrix} \hat{f}_1 \\ \hat{f}_2 \\ \hat{g}_0 \\ \hat{g}_1 \\ \hat{g}_2 \end{bmatrix} = \begin{bmatrix} t_1 - \hat{a}_1 \\ -\hat{a}_2 \\ -\hat{a}_3 \\ 0 \\ 0 \end{bmatrix} \tag{4.17}$$

The matrix in Eq. (4.17) can be rearranged as:

$$\boldsymbol{M\theta_c = N} \tag{4.18}$$

The special banded structure of $\boldsymbol{M}$ is common to Sylvester matrices. Provided that $\hat{A},\hat{B}$ are coprime and of the correct degrees $(\hat{a}_3 \neq 0 \neq \hat{b}_2)$, the matrix $\boldsymbol{M}$ is invertible and $\boldsymbol{M^{-1}}$ can be obtained by standard methods to yield the vector of controller parameters $\boldsymbol{\theta_c}$:

$$\boldsymbol{\theta_c = M^{-1} N} \tag{4.19}$$

Equation (4.19) can be solved for the unknown coefficients of $\hat{F},\hat{G}$ by matrix inversion. However, more effective algorithms exist for solving such polynomial identities (or Diophantine equations) and these methods allow for situations in which $\hat{A},\hat{B}$ may not be coprime. In the same spirit, the values used for n_a, n_b in modeling and recursively estimating $\hat{A},\hat{B}$ may be too large. This again can cause common factors in the estimated values of $\hat{A},\hat{B}$ and consequent ill-conditioning of the matrix inversion method [Wellstead, Zarrop, 1991].

Using the diagram given in Fig. 4.2, it is written that:

$$\hat{G}(z^{-1})y(k) + \Delta\hat{F}(z^{-1})u(k) - \hat{G}(1)r(k) = 0 \tag{4.20}$$

$$\hat{G}(1)w(k) - \hat{G}(z^{-1})y(k) = \Delta u(k)\hat{F}(z^{-1}) \tag{4.21}$$

$$\hat{G}(z^{-1})e(k) = \Delta u(k)\hat{F}(z^{-1}) \tag{4.22}$$

First, introduce a new signal $v(k)$ is defined as

$$\hat{v}(k) := \hat{x}(k) / \hat{z}(k) \tag{4.23}$$

where $x(k)$ and $z(k)$ are defined as

$$\hat{x}(k) := \hat{x}(k-1) + \hat{G}(1)w(k) - \hat{G}(z^{-1})y(k) \tag{4.24}$$

$$\hat{z}(k) := \frac{1}{\hat{F}(z^{-1})}\hat{x}(k) \tag{4.25}$$

Next, substituting (4.23)-(4.25) into (4.20) yields

$$\hat{G}(z^{-1})y(k) + \hat{v}(k)\cdot\Delta u(k) - \hat{G}(k)r(k) = 0 \tag{4.26}$$

$$\hat{l}_i(k) := \frac{\hat{g}_i}{\hat{v}(k)} \qquad (i = 0,1,2) \tag{4.27}$$

Yamamato et al. [Yamamoto, Fujii, Kaneda, 1996] considered a practical scheme based on the proposed self-tuning PID control. $v(k)$ is replaced with $F(1:k)$ in the steady state. Then the following equation recommended:

$$\hat{l}_i(k) := \frac{\hat{g}_i(k)}{\hat{F}(1:k)} \tag{4.28}$$

4.3 Design of $T(z^{-1})$

The control performance depends on the design of the characteristic polynomial $T(z^{-1})$. A method how to design $T(z^{-1})$ is described in [Yamamoto, Fujii, Kaneda, 1996] such that $T(z^{-1})$ can be designed using the rise time and damping property in the expected response. The polynomial $T(z^{-1})$ with coefficients is given as:

$$T = 1 + t_1 z^{-1} + t_2 z^{-2}$$

where the coefficients t_i , t_2 are positive constants. The coefficients t_i are calculated using,

$$t_1 = -2e^{-\frac{\rho}{2\mu}}\cos(\frac{\sqrt{4\mu-1}}{2\mu}\rho) \tag{4.29}$$

$$t_2 = e^{-\frac{\rho}{\mu}} \tag{4.30}$$

$$\rho = \frac{T}{\sigma} \tag{4.31}$$

$$\mu = 0.25(1-\delta) + 0.51\delta, \qquad (0 \le \delta \le 2.0) \tag{4.32}$$

where T, σ and μ denote the sampling interval, the rise time and the damping ratio, respectively. If $\delta = 0$ is set, the binominal model response and the Butterworth model response for $\delta = 1.0$. By varying the factor δ, the response shape can be adjusted arbitrarily, where it is recommended for δ to be between 0.0 and 2.0 practically [Yamamoto, Fujii, Kaneda, 1996].

4.4 Digital PID Implementation

There are three digital PID control implementation methods: (1) Forward rectangular, (2) Trapezoidal and (3) Backward rectangular methods.

Backward rectangular method is used in this research, since it is more suitable for practical use [Bobal, Böhm, Fessl, Machacek, 2005]. The digital PID in a backward rectangular form is given as:

$$u(k) = K_p\left\{e(k)\frac{T_0}{T_I}\sum_{i=1}^{k}e(i) + \frac{T_D}{T_0}\left[e(k)-e(k-1)\right]\right\} \tag{4.33}$$

It is either necessary to calculate integral or controller output value $u(k)$ from a previously recorded value $u(k\text{-}1)$ plus correction increment $\Delta u(k)$. Alternatively, for

a PID controller with digital output, just the increment (change) $\Delta u(k)$ may be calculated. Algorithms which calculate increment (change) $\Delta u(k)$ are referred to as *incremental* or *velocity algorithms*. By subtracting Equation (4.33), which we obtained from the backward rectangular method, for steps k and k-1, we obtain the recurrent relation [Bobal, Böhm, Fessl, Machacek, 2005]. The control input and the increment change are now:

$$u(k) = \Delta u(k) + u(k-1) \tag{4.34a}$$

$$\Delta u(k) = K_p \left\{ e(k) - e(k-1) + \frac{T_0}{T_i} e(k) + \frac{T_d}{T_0} \left[e(k) - 2e(k-1) + e(k-2) \right] \right\} \tag{4.34b}$$

Define a polynomial with real coefficients $L(z^{-1})$ as:

$$L(z^{-1}) = l_0 + l_1 z^{-1} + l_2 z^{-2} \tag{4.35}$$

The control input and its increment is simplified to as:

$$u(k) = l_0 e(k) + l_1 e(k-1) + l_2 e(k-2) + u(k-1) \tag{4.36}$$

$$\Delta u(k) = l_0 e(k) + l_1 e(k-1) + l_2 e(k-2) \tag{4.37}$$

Equation (4.36) (obtained from Eq. (4.33)) can also be written in the form as:

$$u(k) = K_p \left\{ e(k) - e(k-1) + \frac{T_0}{T_i} e(k) + \frac{T_d}{T_0} \left[e(k) - 2e(k-1) + e(k-2) \right] \right\} + u(k-1) \tag{4.38}$$

where the coefficients are:

$$\left.\begin{aligned} K_p &= -l_1 - 2l_2 \\ T_i &= \frac{T_0\left(-l_1 - 2l_2\right)}{l_0 + l_1 + l_2} \\ T_d &= \frac{T_0 l_2}{-l_1 - 2l_2} \end{aligned}\right\} \tag{4.39}$$

The control inputs for forward and trapezoid digital PID controllers are referred to [Bobal, Böhm, Fessl, Machacek, 2005].

5 EXPERIMENTAL APPLICATIONS

In this chapter, the applications for self-tuning pole assignment PID controller are implemented. All studies are performed in laboratory. A DC motor is used as a plant. The results of the present work are compared with that of obtained from conventional PID control methods, thus the advantages and disadvantages of the present self-tuning PID control are emphasized.

5.1 Hardware of the Experimental System

The hardware of experimental system mainly consists of a computer, a data acquisition card, a DC motor, a tachogenerator and a power amplifier. The data acquisition card (ADVANTECH PCL-1800, 330 kHz in speed, conversion time of 2.5µsec) is used to apply control signals yielded by control algorithm to 12V 2400 rpm DC motor. The speed data obtained from tachogenerator are transferred to the computer using data acquisition card. Experimental system picture is shown in Fig. 5.1 and the hardware of the experimental system is shown in Fig. 5.2.



Figure 5.1 Experimental system.

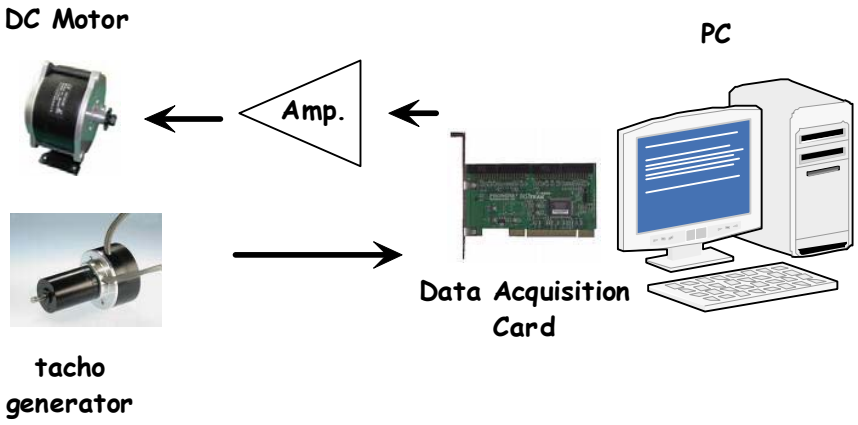


Figure 5.2 Hardware of the experimental system.

5.2 Software of the Experimental System

The software of the applications has mainly three parts:

- System Identification Algorithm.
- Pole Assignment Algorithm.
- PID Control Algorithm.

Steps in Self-tuning control algorithm:

[Step **1**] Set the desired polynomial $T(z^{-1})$ (4.29) – (4.32)

[Step **2**] Estimate $\hat{a}_i(k)$ and $\hat{b}_i(k)$ from (3.2)-(3.23)

[Step **3**] Calculate $F(z^{-1})$ and $Q(z^{-1})$ (4.13) – (4.19)

[Step **4**] Calculate $\hat{v}(t)$, $\hat{x}(t)$, $\hat{z}(t)$ from (4.23) – (4.25)

[Step **5**] Calculate $\hat{l}_i(t)$ from (4.27) or (4.28)

[Step **6**] Calculate PID parameters (K_p, T_i, T_d) from (4.39)

[Step **7**] Calculate $u(k)$ from (4.38)

[Step **8**] Return to [Step 2].

All control solutions and calculations are performed in the MATLAB environment, in Simulink of MATLAB software. As shown in Fig. 5.3 below, the control signal yielded from the control algorithm prepared in MATLAB are applied to the DC motor using analog output of the data acquisition card and the speed signal is read by the tachogenerator connected to the motor. This signal is transferred to MATLAB using analog input of the data acquisition card.

System identification algorithms are written in an S-Function with initial parameters and there is no need any algorithmic modification when the order of numerator or denominator of the plant model is changed.

Pole assignment algorithm and PID control algorithm are written in MATLAB Function. In the pole assignment algorithm, controller parameters are calculated. In the PID control algorithm, the parameters which calculated from pole assignment algorithm are used for calculating control signal according to discretizing integral component method.

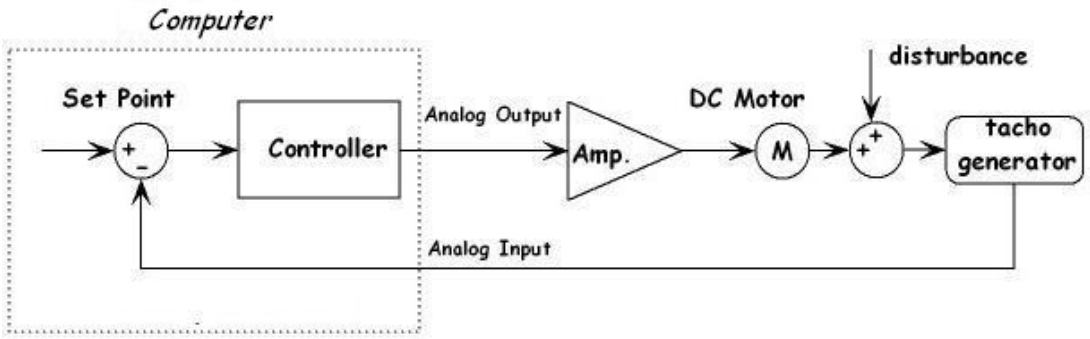


Figure 5.3 The block diagram of the experimental system.

5.3 Plant Model

DC motor is an electrical motor that basically used as torque transducer. The significance of DC motors is increasing in many industrial applications because of the following reasons: DC motors have high speed capabilities and low rotor inertia,

also their heat dissipation characteristics is better than the other types of motors. Therefore, DC motors have been widely used in many industrial applications. Moreover DC motor is a good actuator for varying load conditions where medium or less power is needed [Kim, 2009].

The mathematical model representing a dc motor can be given as:

$$v_a = R_a i_a + L_a\left(\frac{di_a}{dt}\right) + e_a \tag{5.1}$$

$$e_a = K_m \omega_m \tag{5.2}$$

$$T_m = K_m i_a \tag{5.3}$$

$$J_m\left(\frac{d\omega_m}{dt}\right) = T_m - T_{s1} - B_m\omega_m - T_f(\omega_m) \tag{5.4}$$

$$J_1\left(\frac{d\omega_1}{dt}\right) = T_{s1} - T_{s2} - B_1\omega_1 - T_f(\omega_1) \tag{5.5}$$

$$J_L\left(\frac{d\omega_L}{dt}\right) = T_{s2} - B_L\omega_L - T_d - T_f(\omega_L) \tag{5.6}$$

$$T_{s1} = k_1(\theta_m - \theta_1) + d_1(\omega_m - \omega_1) \tag{5.7}$$

$$T_{s2} = k_2(\theta_1 - \theta_L) + d_2(\omega_1 - \omega_L) \tag{5.8}$$

$$\frac{d\theta_m}{dt} = \omega_m, \quad \frac{d\theta_L}{dt} = \omega_L, \quad \frac{d\theta_1}{dt} = \omega_1, \tag{5.9}$$

where v_a is the armature voltage, i_a the armature current, R_a the armature resistance, L_a the armature inductance, e_a the back electromotive force, K_m the motor's torque constant, k_1, k_2 the spring constants, B_m, B_1 and B_L the viscous friction coefficients, d_1, d_2 the shaft inner damping coefficients, J_m, J_1, J_L the moments of inertia, ω_m, ω_1, ω_L the angular speeds, T_d the load torque, T_f the non-

linear frictional torque and T_{s1}, T_{s2} the load torques. The non-linear friction function, T_f can be given as [Seborg, Edgar, Mellichamp, 1989]:

$$T_f(\omega)=(\alpha_0+\alpha_1 e^{-\alpha_2|\omega|})\text{sgn}1(\omega)+(\alpha_3+\alpha_4 e^{-\alpha_5|\omega|})\text{sgn}\,2(\omega) \tag{5.10}$$

where the functions sgn1 and sgn2 are defined as:

$$\text{sgn}1(\omega)=\begin{cases}1 & \omega\geq 0\\ 0 & \omega<0\end{cases},\text{sgn}\,2(\omega)=\begin{cases}0 & \omega\geq 0\\ -1 & \omega<0\end{cases} \tag{5.11}$$

where $\alpha_i \in R$, $\alpha i>0,(i=0,.....,5)$ and $\alpha_0\neq\alpha_3, \alpha_1\neq\alpha_4, \alpha_2\neq\alpha_5$, R is the vector of real numbers. The schematic diagram of the dc motor is shown in Fig.5.4.

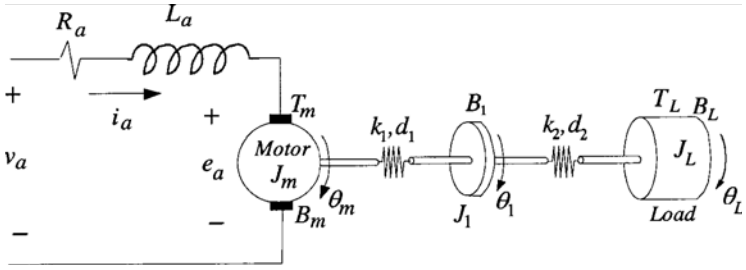


Figure 5.4 Schematic diagram of the electrical drive system.

The process reaction curve method is one of the widely used conventional approaches to predetermine the dynamic behavior of a system under load or no-load conditions [Eker, 2006]. Some dynamical properties of the systems can be obtained using the method such that rise time, settling time, time constant, time delay, and type of response can be determined using this method. A step input signal change is applied, and the output response is measured. The approximate plant model is obtained that is based on the process reaction curve method [Kim, 2009]:

$$G(s) = \frac{Ke^{-T_d s}}{\tau s + 1} \cong \frac{K}{(1+T_d s)(1+\tau s)} \tag{5.12}$$

where K is the steady-state gain, T_d is the time delay in second and τ is the time constant, in second.

In preliminary studies, the motor speeds at different input armature voltages are measured in open-loop conditions to obtain the tachogenerator characteristics. The tachogenerator has almost linear characteristics and its gain is calculated to be 2.15 volt/rad/s. A step input signal change is applied to the plant and the output response is measured as plotted in Fig. 5.5. A 7 volts input (armature voltage) corresponds to 1500 rpm speeds. The response is obtained using 3 msecs sampling period. A low-pass filter is used to filter the output speed signal from high frequency noise components. The rise time is observed to be about 0.3 s while delay time at the beginning is about 0.017 s. The steady-state gain is 0.875 and the time constant is 0.138 s.

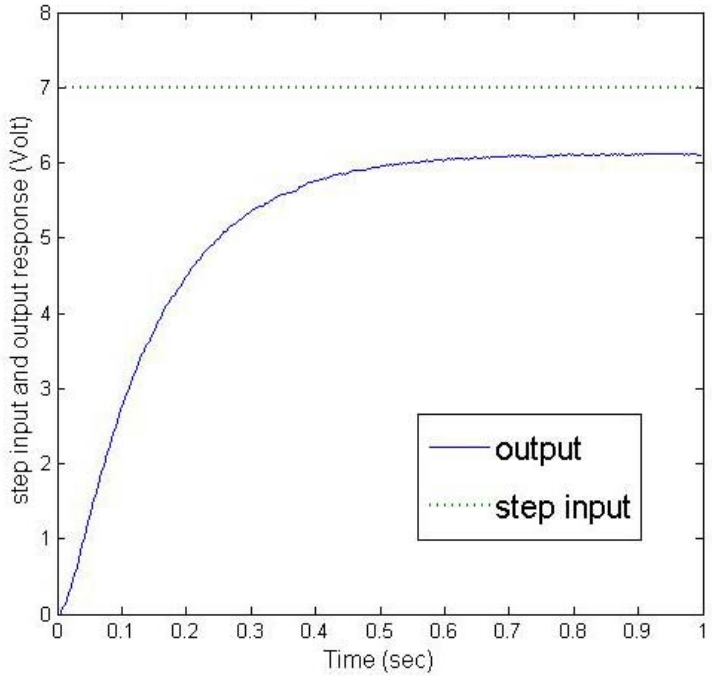


Figure 5.5 Step response of DC motor.

Using the approximate plant model given in Eq. (5.12), the plant gain coefficients are calculated and the nominal mathematical model in a transfer function form is approximated as:

$$G(s) = \frac{372.975}{s^2 + 66.070s + 426.257}$$

The model response is plotted in Fig.5.6.

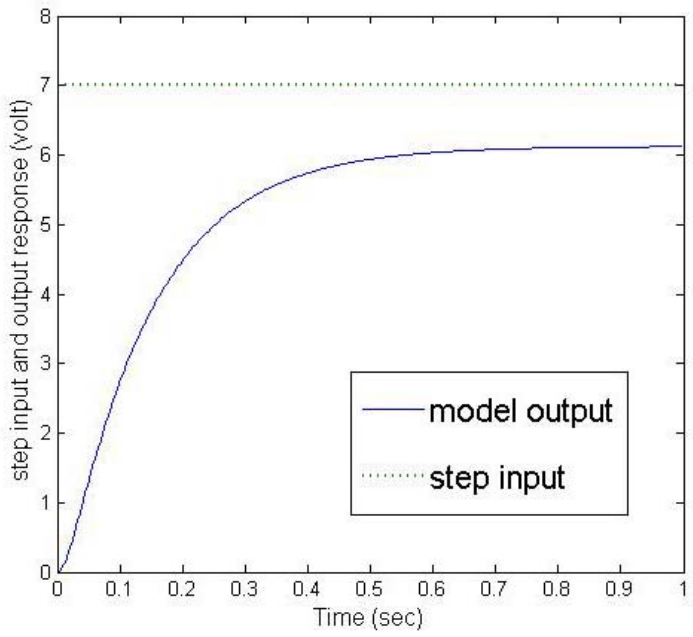


Figure 5.6 Step response of model.

The graph for the model validation is illustrated in Fig. 5.7 such that solid line presents real system output and dotted line is the approximated model response, and the speed error is illustrated in Fig. 5.8. The output speed error settles down after 0.5 sec and the error is about ±4 rpm at the steady-state. The mean error in Fig. 5.8 is

zero, that is, the steady-state modeling error is zero. The modeling error ± 8 rpm corresponds to ± 0.53 % (±8/1500) modeling error.

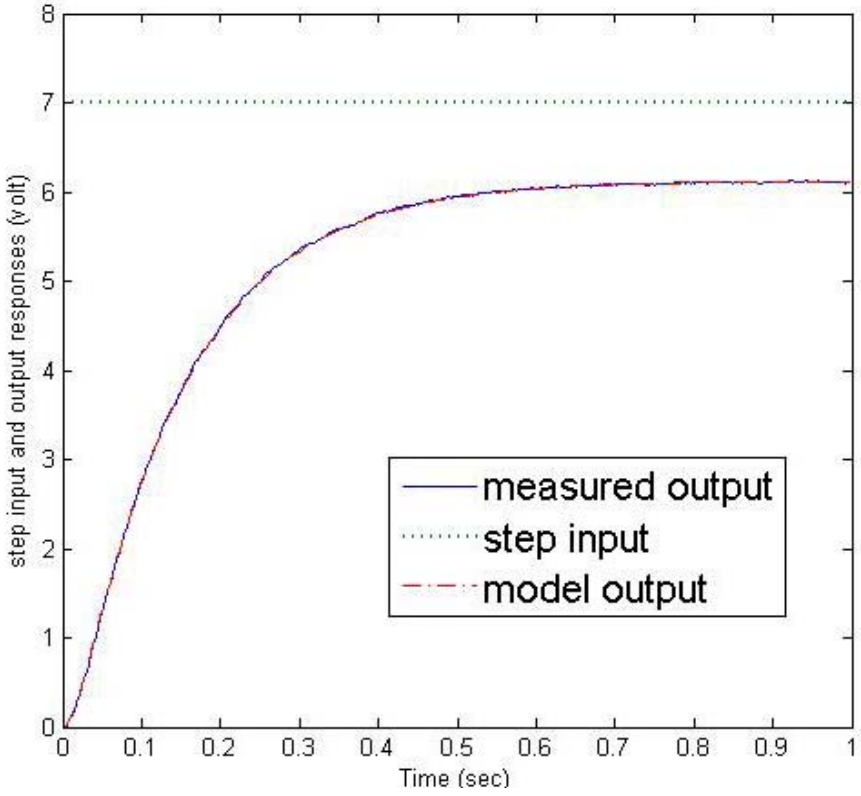


Figure 5.7 Step responses of measured output and model output.

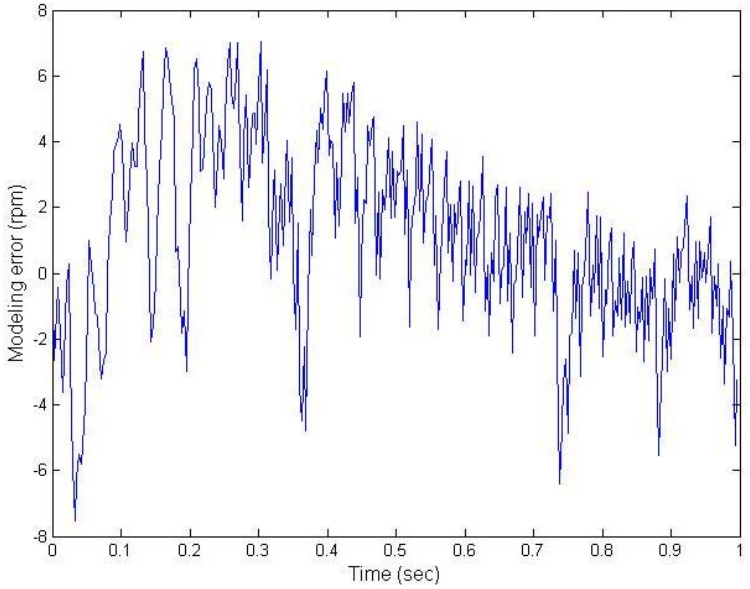


Figure 5.8 Modeling error.

5.4 Ziegler - Nichols Process Reaction Method Results

As described in Section 3.2.1.1. Ziegler-Nichols Process Reaction Method is the earliest method for the PID controller. This method can be viewed as traditional method based on modeling and control. The value of P, PI and PID controller parameters are calculated using the set values presented in Table 3.1 and the controller parameters are $K_p = 9.27$ for P controller, $K_p = 8.34$ and $T_i = 0.051$ for PI controller and $K_p = 11.12$ *and* $T_i = 0.034$ *and* $T_d = 0.0085$ for PID controller, respectively.

Responses to step reference changes are illustrated in Fig 5.9 for P control, Fig. 5.10 for PI control and Fig 5.11 for PID control systems. There is large steady-state error in P control system, while 15.3 % overshoot is obtained with PI control system.

All responses are illustrated in Fig. 5.12 and corresponding control signals (armature voltage variations) are shown in Fig. 5.13 for Z-N process reaction curve method. The best response is obtained from PID control system. The time specifications are summarized in Table 5.1.

P, PI and PID control system results are shown in Fig. 5.9, Fig. 5.10 and Fig. 5.11 respectively. 0-6V step signal (0-1500 rpm) is applied to the system as reference signal. It can be clearly seen that there is a large steady-state speed error for P control. PI and PID type control reduces the steady-state error to zero.

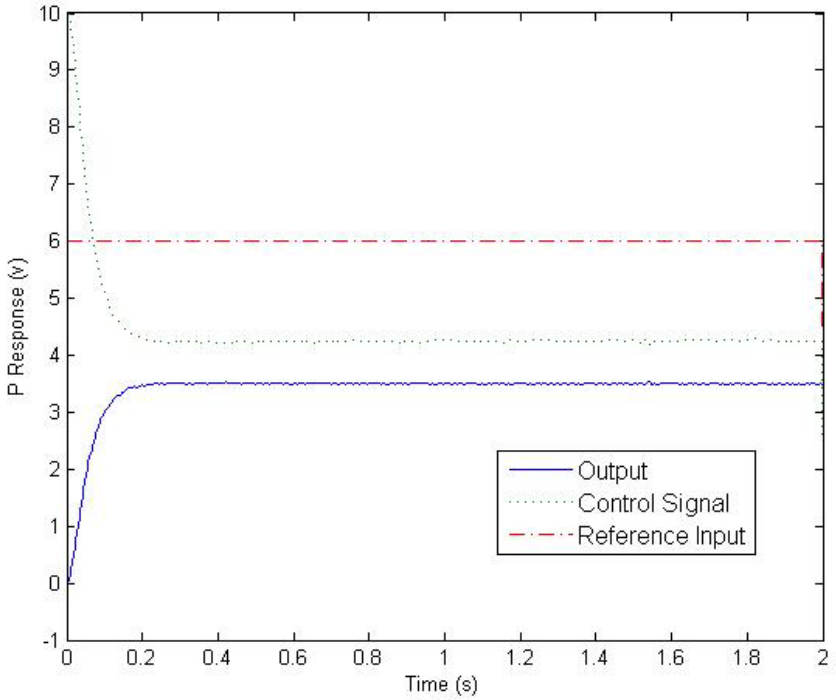


Figure 5.9 Responses to step setpoint change for P controller.

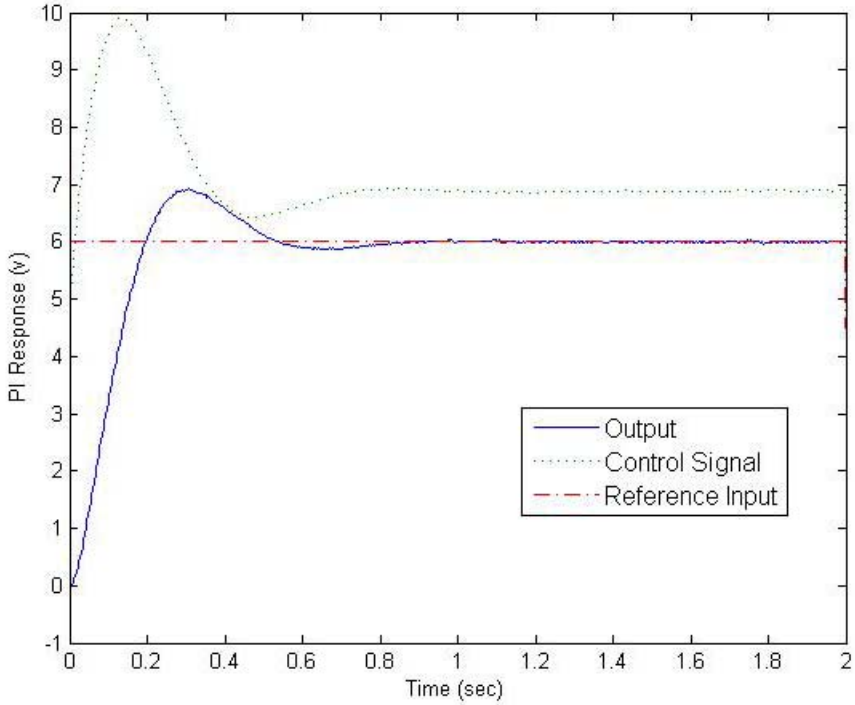


Figure 5.10 Responses to step setpoint change for PI controller.

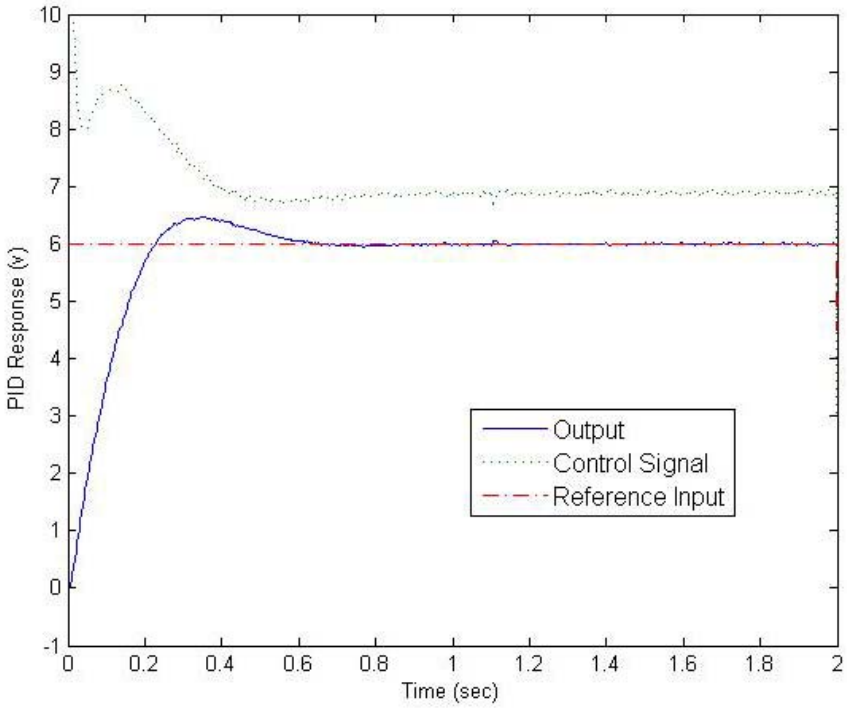


Figure 5.11 Responses to step setpoint change for PID controller.

Some quantitative performance indicators of the speed tracking quality are presented in Table 5.1 for Z-N P, PI and PID controllers. It can be seen from Fig. 5.12 and Table 5.1 that PID type controller provides the best response in the Z-N process reaction methods. PID response has smaller settling time, less overshoot and smaller output fluctuations in magnitude. Control signals of P, PI and PID control are shown in Fig. 5.13.

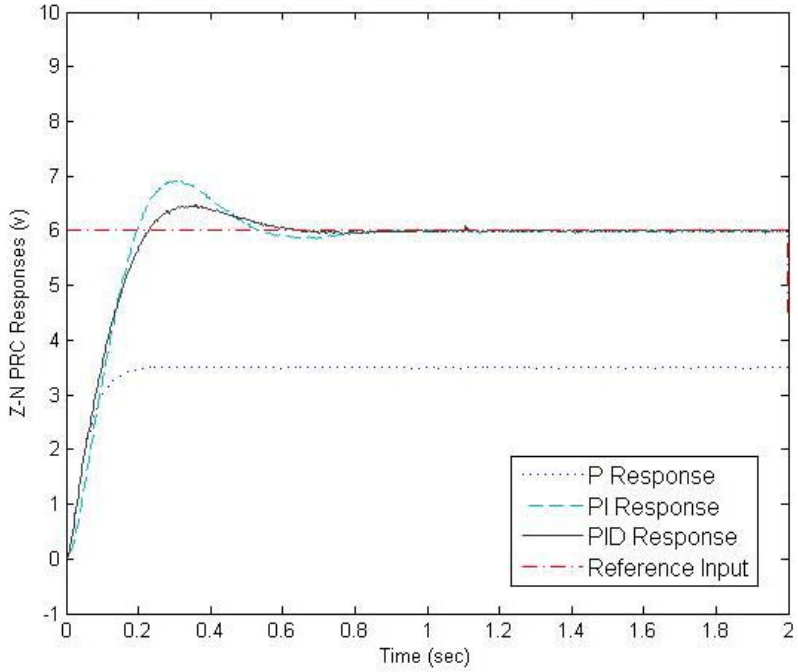


Figure 5.12 Response to step setpoint change for Z-N PRC method.

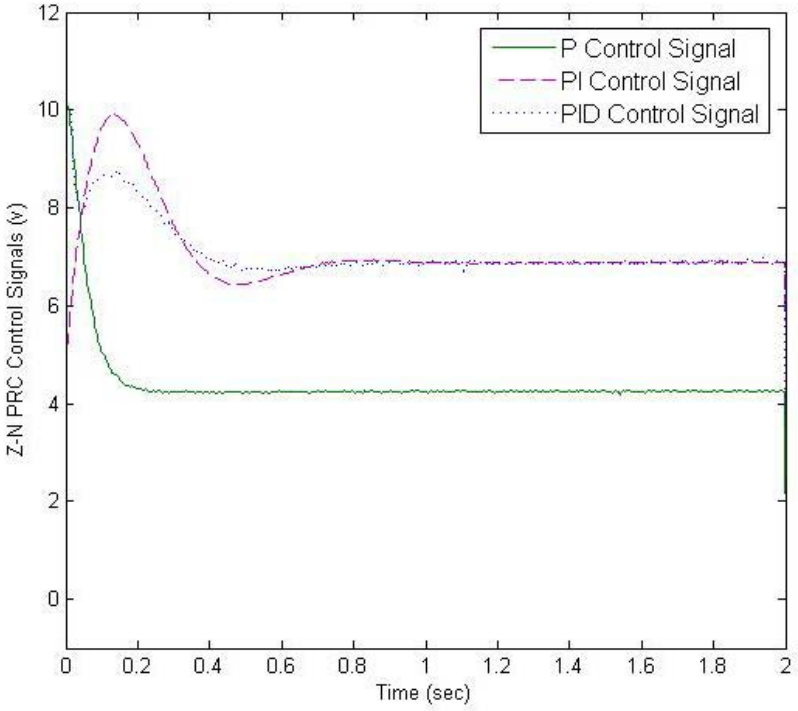


Figure 5.13 Control signals of Z-N PRC method.

Table 5.1 Time domain specifications of Z-N process reaction method.

Controller type	Rise Time (ms)	Settling Time (ms) (5%)	Overshoot (%)	Output deviations (rpm)
Z-N P	**95**	**200**	**-**	**± 2.3**
Z-N PI	**138**	**464**	**15.3**	**± 3.8**
Z-N PID	**155**	**450**	**7.7**	**± 3.5**

5.5 Cohen – Coon Reaction Curve Method Results

The values of P, PI and PID controller parameters are determined by Cohen-Coon process reaction method (Table 3.3.) and the controller parameters are calculated to be $K_p = 9.65$ for P controller, $K_p = 8.43$ *and* $T_i = 0.045$ for PI controller and $K_p = 12.6$ and $T_i = 0.039$ and $T_d = 0.006$ for PID controller, respectively.

As a reference, again 0-6V step signal is applied to the system. It can be also clearly seen that there is a large steady-state speed error for P control. PI and PID type control reduces the steady-state error to zero.

Some time domain specifications of the speed tracking quality are presented in Table 5.2 for Cohen-Coon P, PI and PID controllers. The corresponding control signals of P, PI and PID control are shown in Fig. 5.18. It can be seen from Fig. 5.17 and Table 5.2 that PID type controller provides the best response in the Cohen-Coon reaction curve method such that PID response has smaller rise time, smaller settling time, and smaller output fluctuations in magnitude.

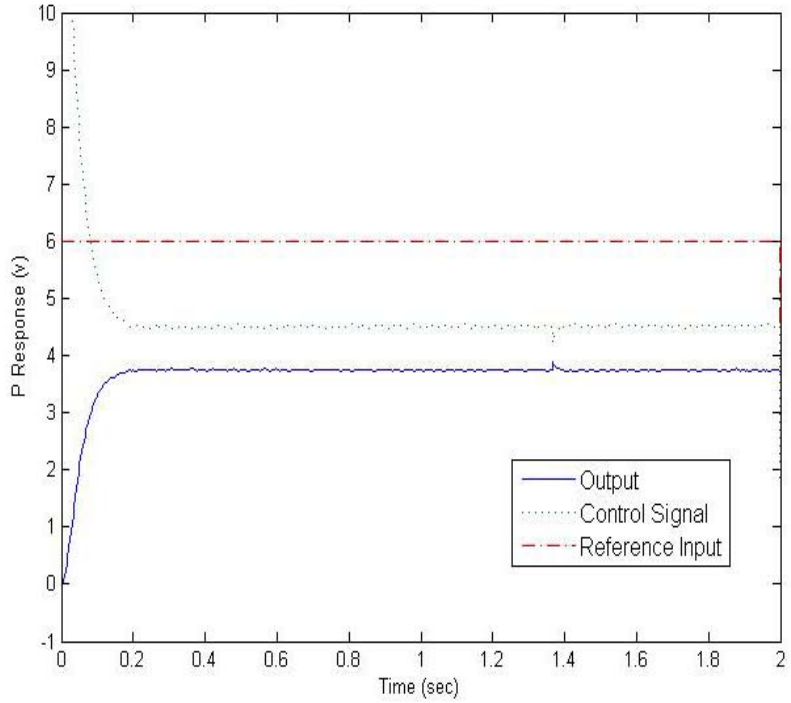


Figure 5.14 Responses to step setpoint change for P controller.

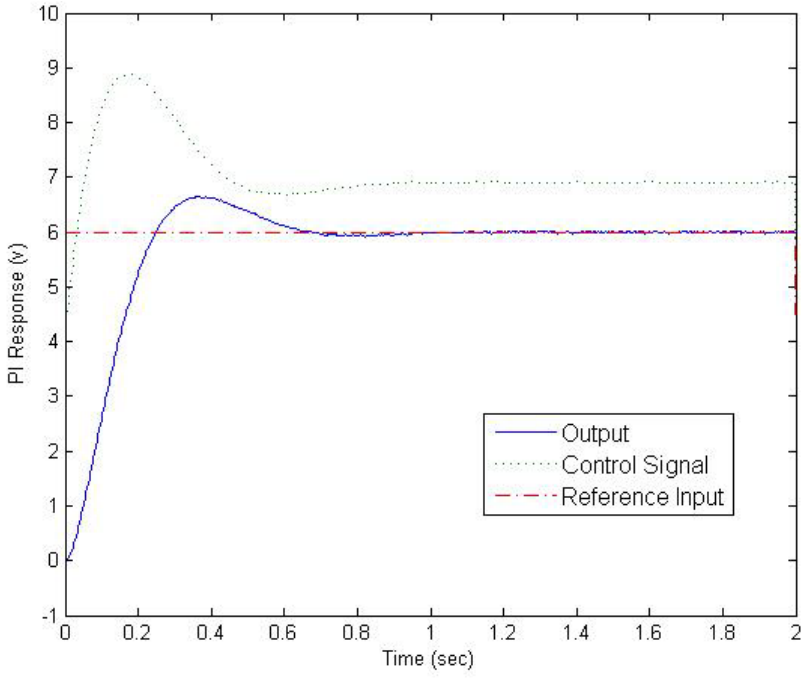


Figure 5.15 Responses to step setpoint change for PI controller.

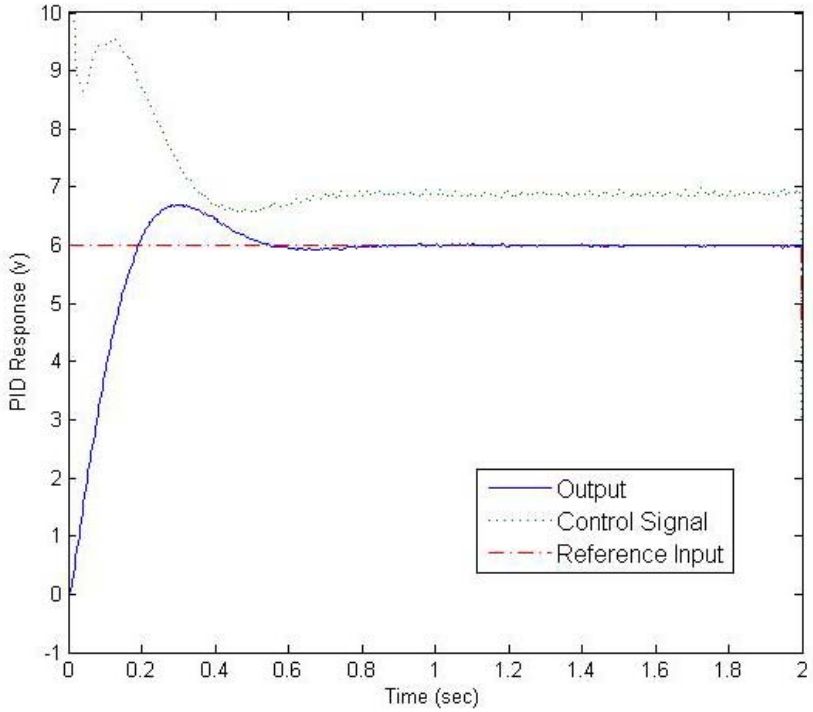


Figure 5.16 Responses to step setpoint change for PID controller.

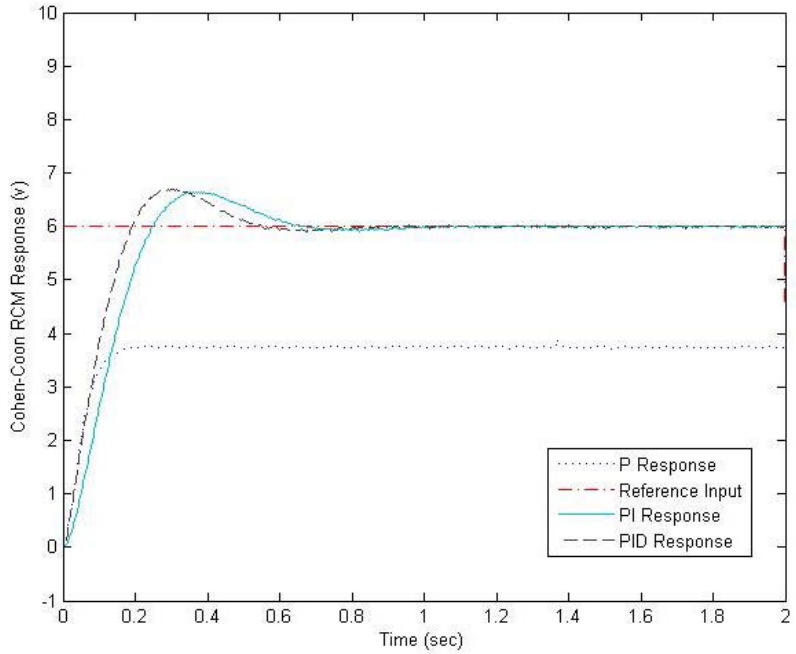


Figure 5.17 Responses to step setpoint change for Cohen-Coon RCM.

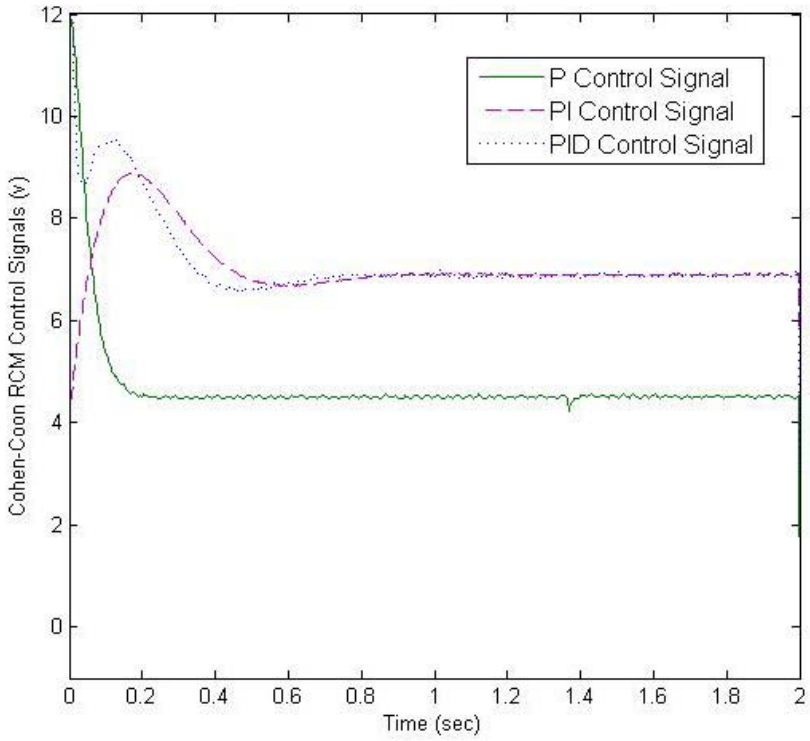


Figure 5.18 Control signals of Cohen-Coon RCM designed control system.

Table 5.2 Time domain specifications of Cohen-Coon process reaction method.

Controller type	Rise Time (ms)	Settling Time (ms) (5%)	Overshoot (%)	Output deviations (rpm)
Cohen-Coon P	**85**	**-**	**-**	**± 2.4**
Cohen-Coon PI	**168**	**525**	**10.8**	**± 3.7**
Cohen-Coon PID	**138**	**438**	**11.6**	**± 3.5**

According to the results for Z-N process reaction and Cohen-Coon reaction curve methods, it can be seen that Cohen-Coon PID control gives smaller rise time, smaller settling time but more overshoot than Z-N PID control, while Z-N PI has smaller rise time, smaller settling time but more overshoot than Cohen-Coon PI control.

5.6 Self-Tuning Pole Assignment Method Results

As described in Chapter 4, the algorithm given in Section 5.2, is applied for self-tuning PI and self-tuning PID control. ARX model is selected and Recursive Least Squares method (RLS) is used for system identification. For all self-tuning tests, sampling time is chosen to be 20 msecs. The sampling time is chosen to be suitable according to the total computation time illustrated in Fig. 3.12.

As design requirements, fast responses, less overshoot, smaller output fluctuations in magnitude, smaller rise time, low variations of control signal are aimed.

F and *G* polynomials are calculated which described in pole assignment algorithms, using *A* and *B* polynomials estimated by system identification algorithm. PID gains (K_p, T_i, T_d) are calculated from *F* and *G* polynomials.

Self-Tuning PI control responses are shown in Fig. 5.19, variations of identification parameters are shown in Fig. 5.20 and variations of self-tuning PI control gains (K_p, T_i) shown in Fig. 5.21. The system is assumed to be 2nd order in Self-Tuning PI Control. So, system identification polynomials are $\hat{A} = 1 + \hat{a}_1 z^{-1} + \hat{a}_2 z^{-2}$, $\hat{B} = \hat{b}_0 + \hat{b}_1 z^{-1}$, controller polynomials are $\hat{F} = 1 + \hat{f}_1 z^{-1}$, $\hat{G} = \hat{g}_0 + \hat{g}_1 z^{-1} + \hat{g}_2 z^{-2}$ and desired close loop polynomial is $T = 1 + t_1 z^{-1} + t_2 z^{-2}$.

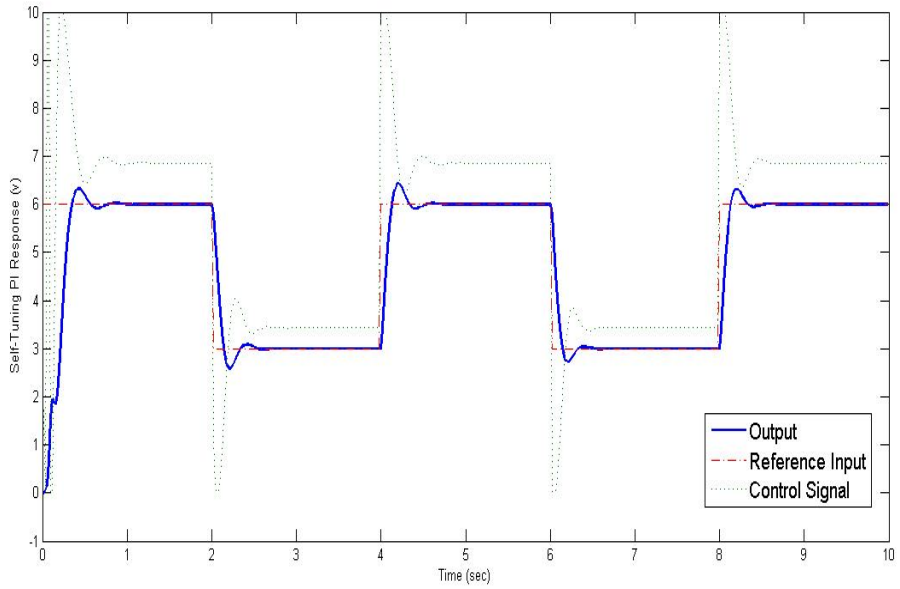


Figure 5.19 Response of the self-tuning PI control system.

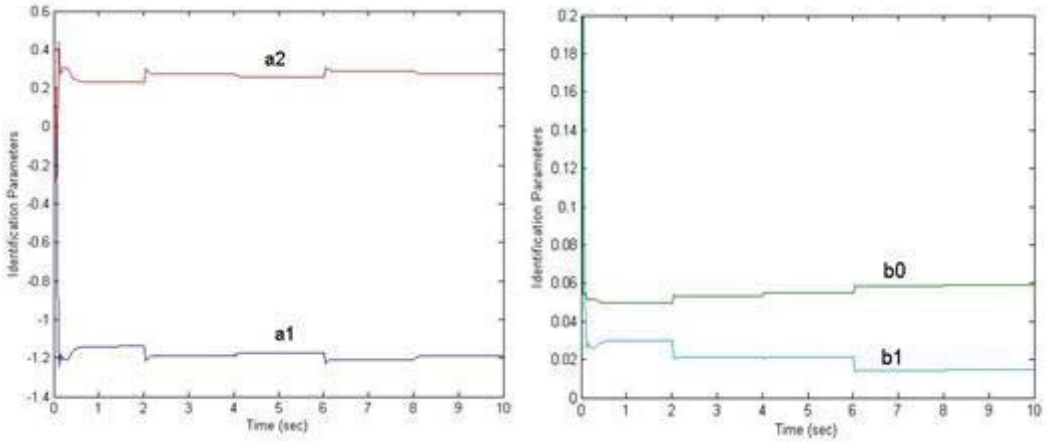


Figure 5.20 Variations in identification parameters for self-tuning PI control.

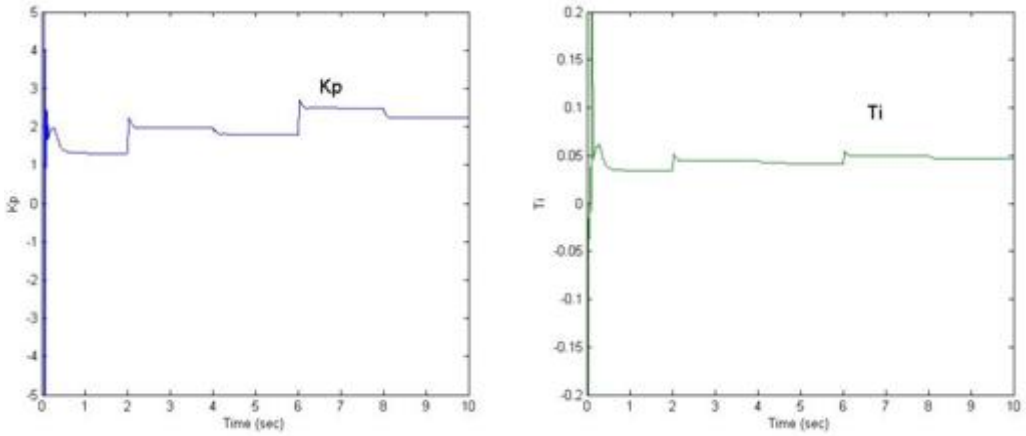


Figure 5.21 Variations of self-tuning PI controller gains.

The system is assumed to be as 3rd order in Self-tuning PID control. Self-tuning PID control responses are shown in Fig. 5.22, identification parameters is shown in Fig. 5.23 and Self-tuning PID control gains (K_p, T_i, T_d) are shown in Fig. 5.24. The system identification polynomials are $\hat{A} = 1 + \hat{a}_1 z^{-1} + \hat{a}_2 z^{-2} + \hat{a}_3 z^{-3}$, $\hat{B} = \hat{b}_0 + \hat{b}_1 z^{-1} + \hat{b}_2 z^{-2}$, controller polynomials are $\hat{F} = 1 + \hat{f}_1 z^{-1} + \hat{f}_2 z^{-2}$, $\hat{G} = \hat{g}_0 + \hat{g}_1 z^{-1} + \hat{g}_2 z^{-2}$ and desired close loop polynomial is $T = 1 + t_1 z^{-1} + t_2 z^{-2}$.

As a reference, 3-6V square wave signal is applied to the system. The estimated system parameters converge to the real system parameters, the responses of the self-tuning PI and self-tuning PID control get better in every sampling cycle, as shown in Figs. 5.19-5.22.

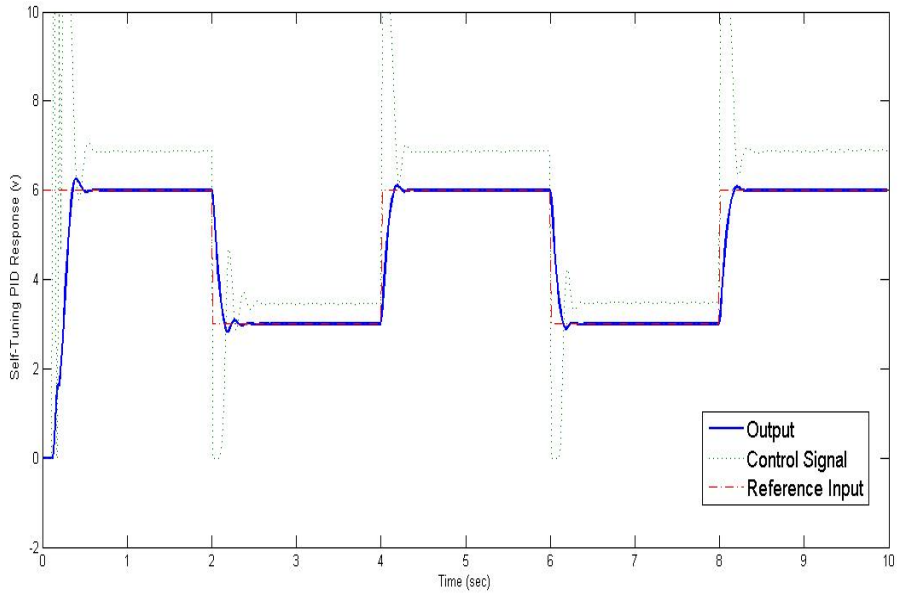


Figure 5.22 Responses of the self-tuning PID control system.

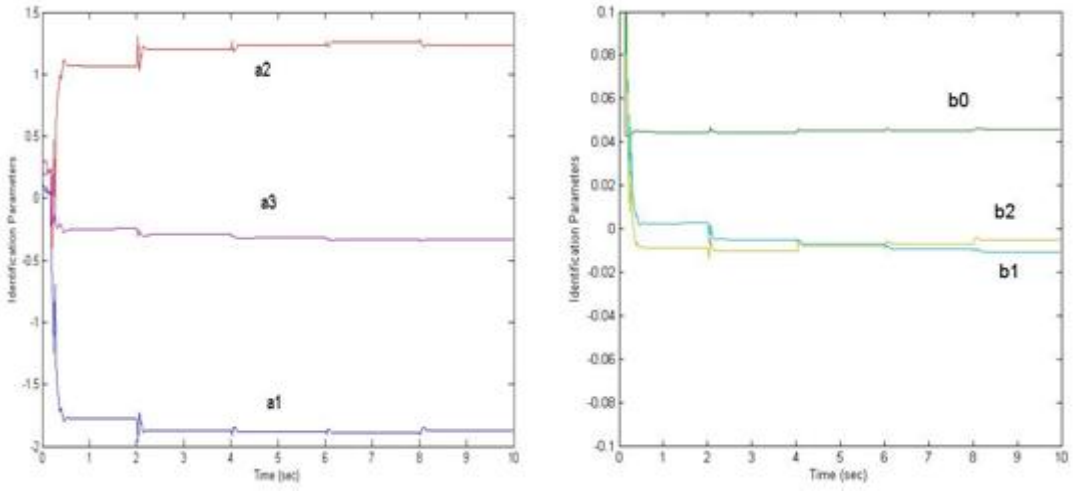


Figure 5.23 Variations of identification parameters for self-tuning PID control.

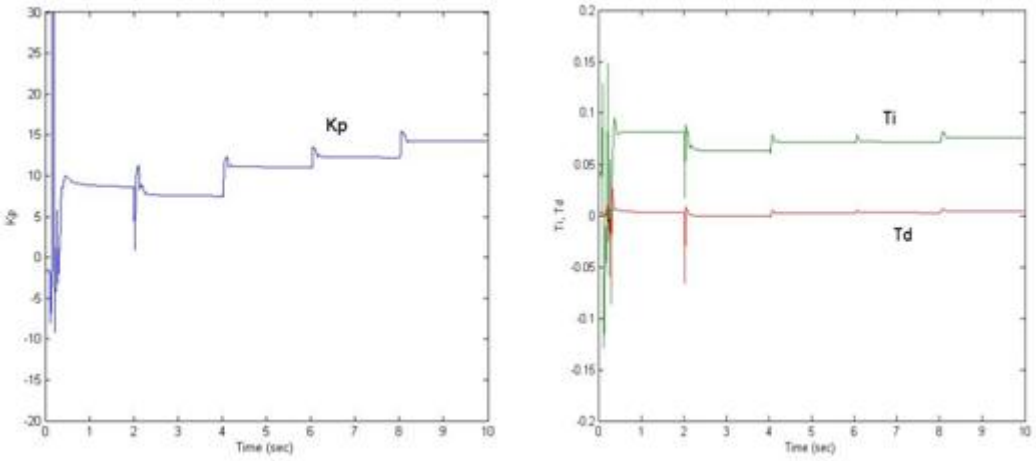


Figure 5.24 Variations of self-tuning PID controller gains.

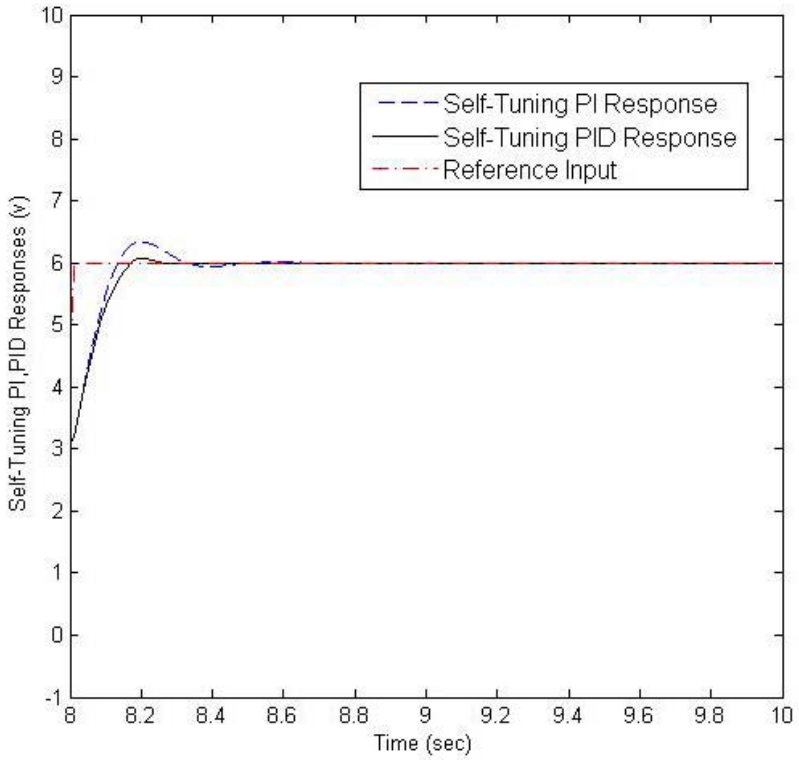


Figure 5.25 Responses to step setpoint change with self-tuning PI and self-tuning PID controllers.

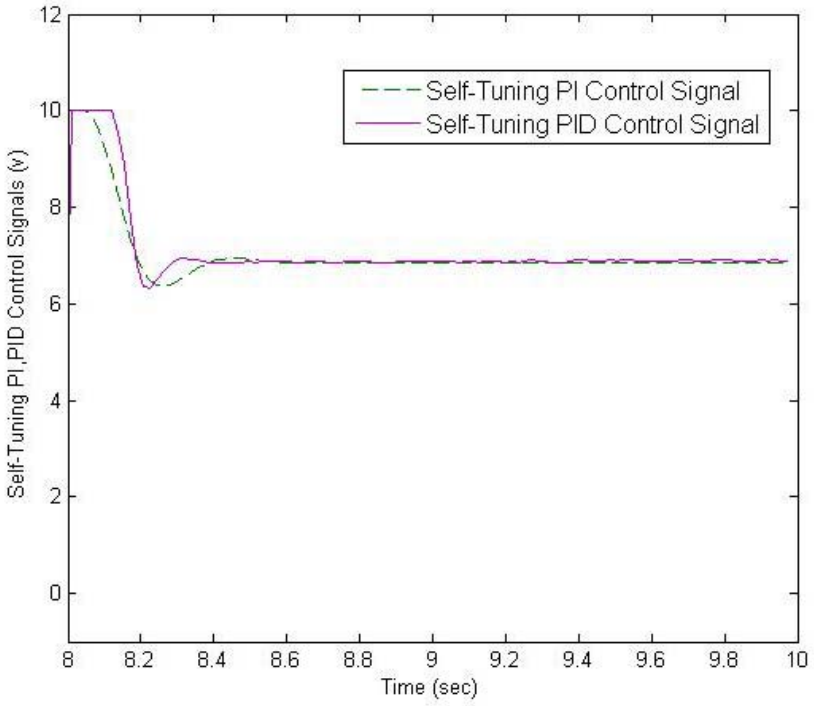


Figure 5.26 Control signals of self-tuning PI and self-tuning PID control system.

Some quantitative performance indicators of the speed tracking quality are presented in Table 5.3 for self-tuning PI, self-tuning PID and the corresponding control signals of self-tuning PI and self-tuning PID control are shown in Fig. 5.26. It can be seen from Fig. 5.25 and Table 5.3 that self-tuning PID type controller provided better response than self-tuning PI controller. Self-tuning PID control system has smaller settling time, less overshoot and smaller output fluctuations in magnitude.

Table 5.3 Time domain specifications of self-tuning PI and PID control system.

Controller type	Rise Time (ms)	Settling Time (ms) (5%)	Overshoot (%)	Output deviations (rpm)
Self-tuning PI	**96**	**281**	**10.5**	**± 2.8**
Self-tuning PID	**115**	**156**	**1.2**	**± 2.3**

To test the tracking performance of the system, a square wave command trajectory is applied to the system and the responses of PI, PID type controls are shown in Fig. 5.27 and Fig. 5.28 respectively. These figures confirm the fact that the system with the self-tuning controller has a better tracking performance than the system with conventional PI and PID controllers.

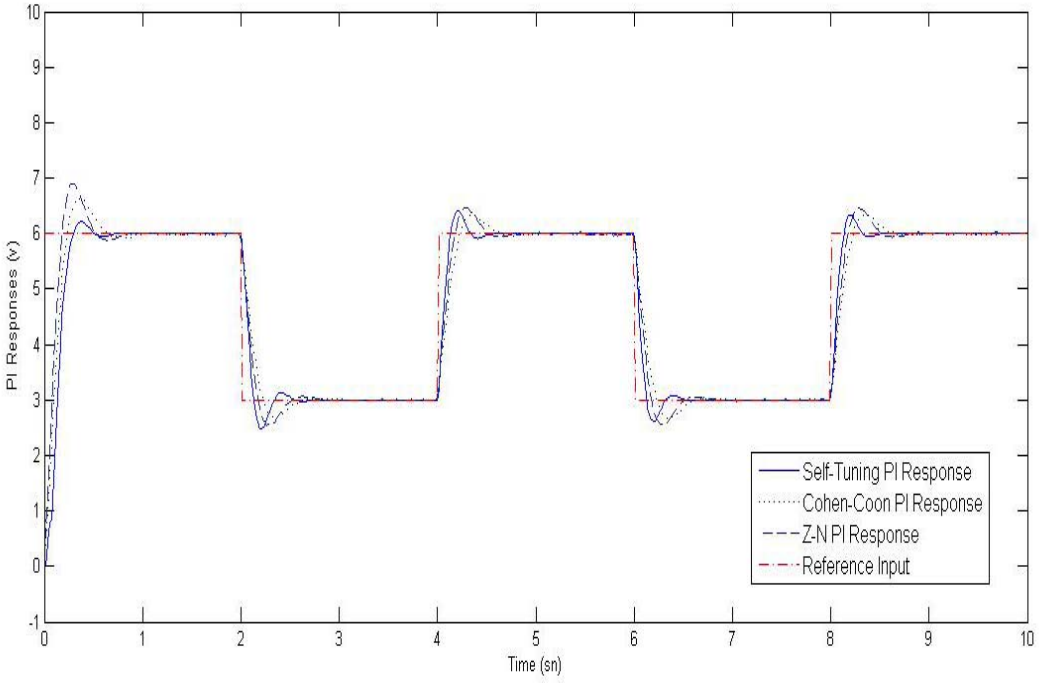


Figure 5.27 PI controller responses to step setpoint change for self-tuning, Z-N and Cohen-Coon system.

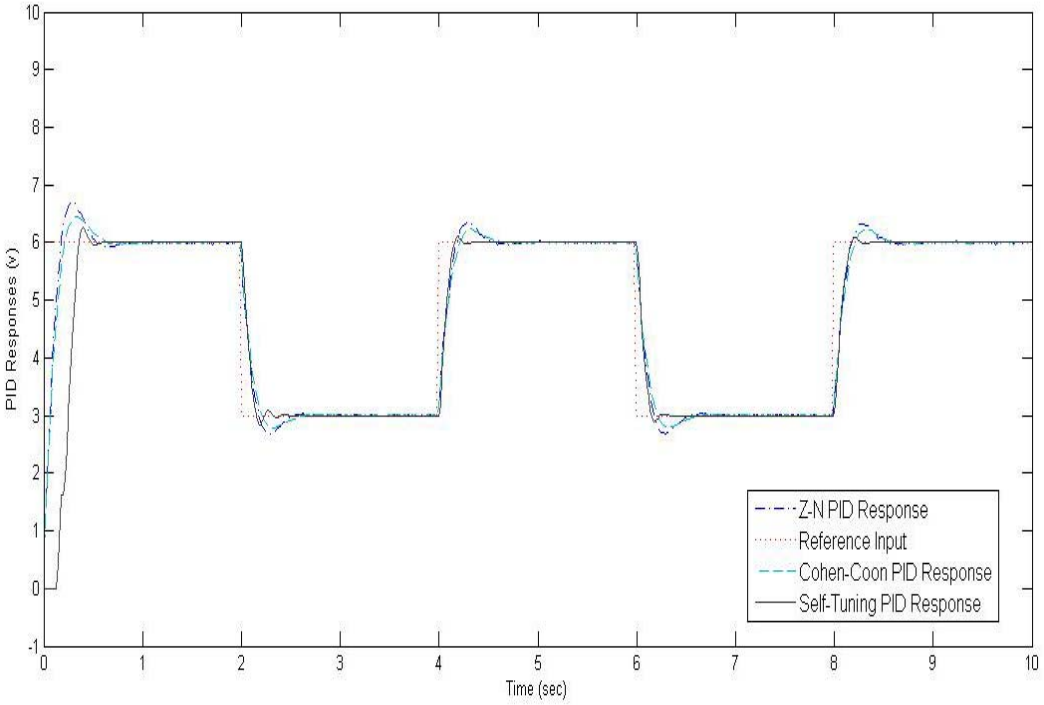


Figure 5.28 PID controller responses to step setpoint change for of self-tuning, Z-N and Cohen-Coon system.

The satisfactory responses are summarized such that Cohen-Coon PID, Z-N PID and self-tuning PID control responses are illustrated in Fig. 5.29 and the corresponding control signals are shown in Fig. 5.30.

The results of some quantitative performance indicators of the speed tracking quality are presented in Table 5.4 for self-tuning PID, Cohen-Coon PID and Z-N PID controllers. It is fact that self-tuning PID type controller has provided best response in tracking.

Table 5.4 Time domain specifications.

Controller type	Rise Time (ms)	Settling Time (ms) (5%)	Overshoot (%)	Output deviations (rpm)
Cohen-Coon PID	**142**	**433**	**5.4**	**± 2.6**
Z-N PID	**159**	**432**	**3.8**	**± 2.7**
Self-tuning PID	**115**	**156**	**1.2**	**± 2.3**

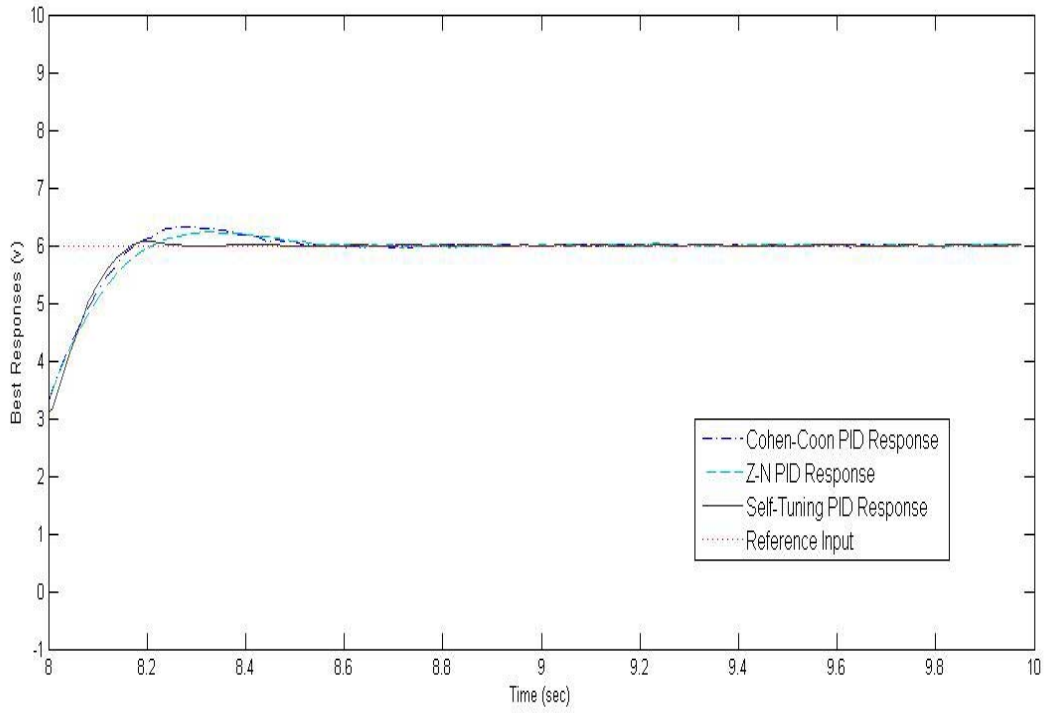


Figure 5.29 Responses of different control systems.

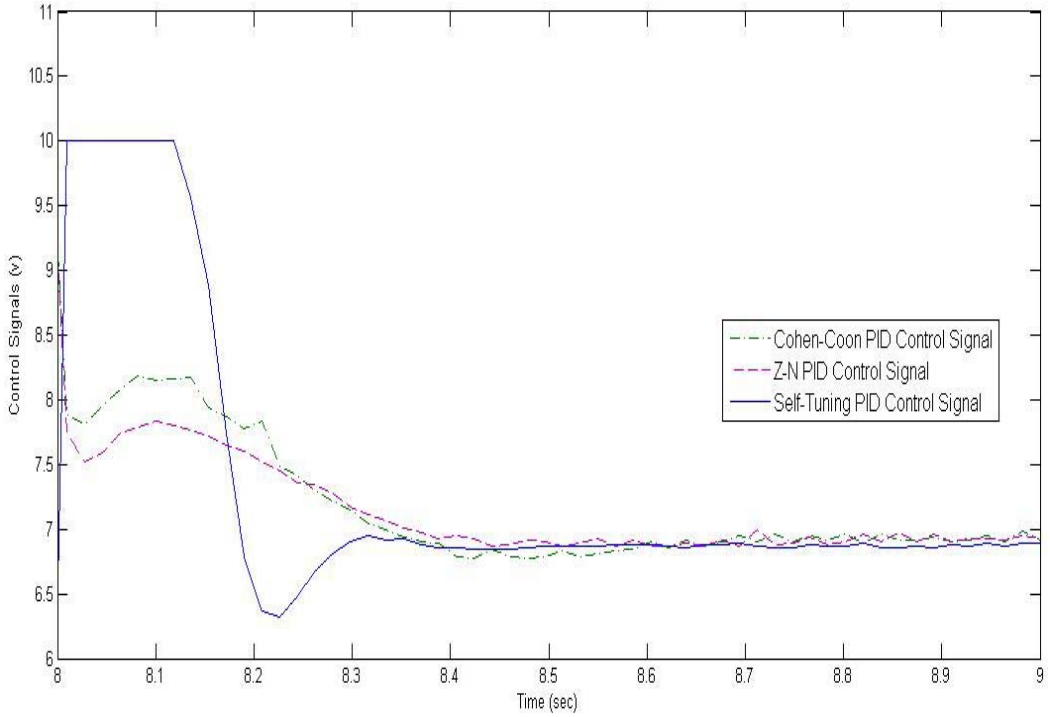


Figure 5.30 Control signals of the systems.

For the robustness and regulatory behavior of the system, ±0.35V square wave signal with a period of 2 secs (corresponds with ±91.87 rpm speed) was added at the measured output of the system as external load disturbance as shown in Fig 5.31 and the responses are illustrated in Fig. 5.32, corresponding control signals are illustrated in Fig. 5.33. Obviously, the control performance of the proposed self-tuning PID controller is better than conventional PID controllers such that the control system recovers the external disturbance in about 0.15 secs. On the other hand, the system with Z-N PID and Cohen-Coon PID controllers recover the external disturbance in about 0.45 secs. The control input of the self-tuning PID control system varies more slowly and in smaller amplitude compared with the others as illustrated in Fig 5.33. It is the fact that the smaller the variations in amplitude the better the control. The issue reduces maintenance of the actuators in industrial control systems. Some time domain specifications for the regulatory behavior are presented in Table 5.5.

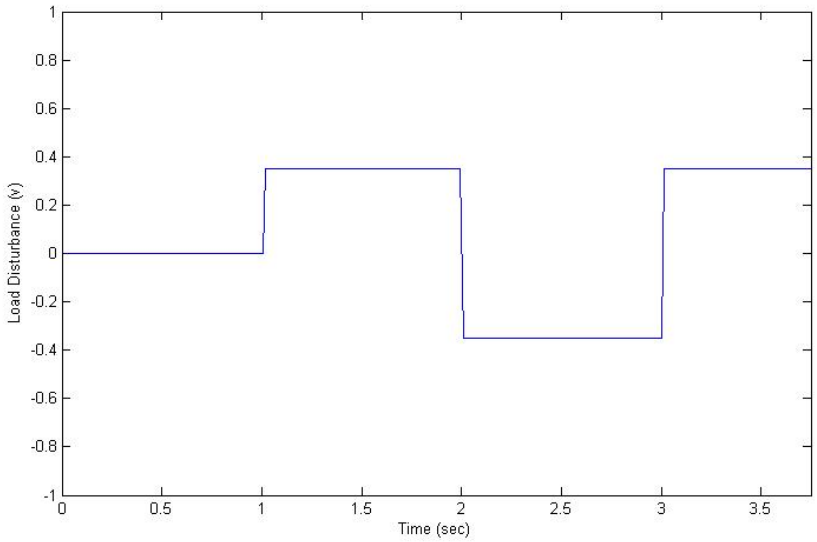


Figure 5.31 Load Disturbance

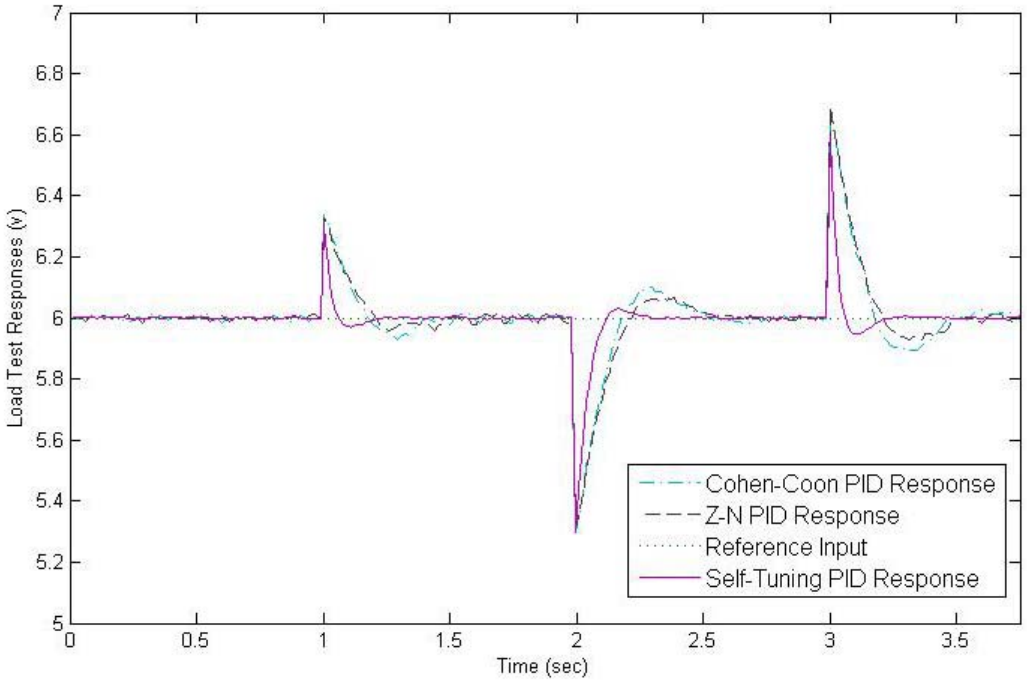


Figure 5.32 Output speeds to ±91.87 rpm square wave external load disturbance.

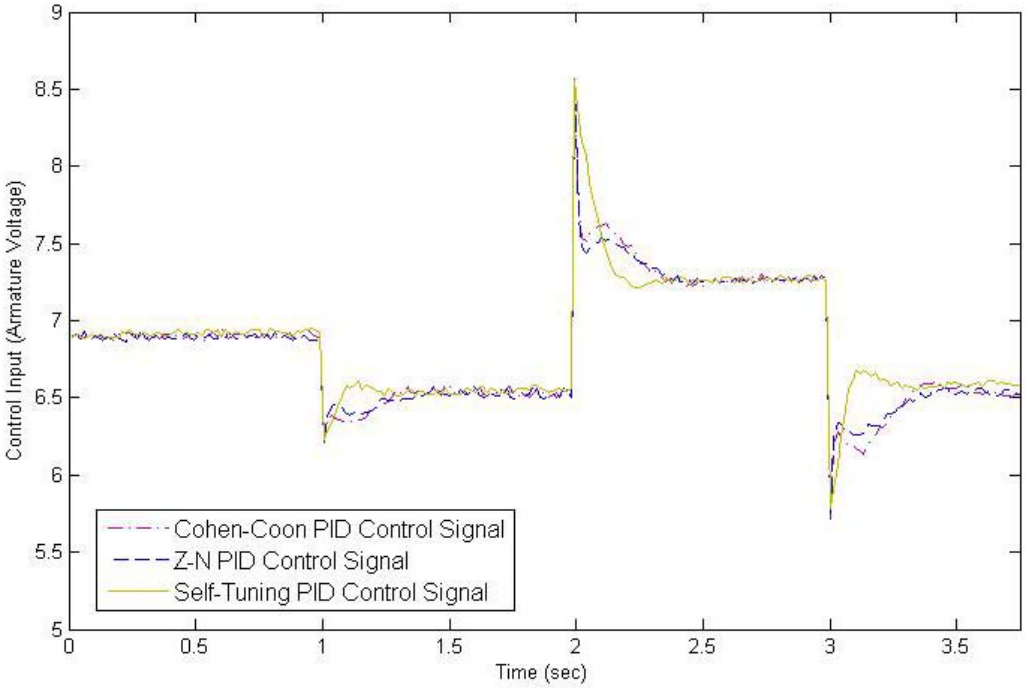


Figure 5.33 Control signals to ±91.87 rpm square wave external load disturbance.

Table 5.5 Time domain specifications for load test.

Controller type	Recovery Time (ms)	Output deviations (v)	Output deviations (rpm)
Cohen-Coon PID	**445**	**0,015**	**± 3.9**
Z-N PID	**475**	**0.012**	**±3.1**
Self-tuning PID	**150**	**0.008**	**± 2.1**

Based on the experimental results, it can be concluded that the overshoot was significantly reduced to desired level and the speed of the drive system was improved such that rise time and settling time improved compared with the Ziegler-Nichols and Cohen-Coon conventional PID control. The system with the self-tuning controller has a better tracking performance and regulatory behavior than the system with conventional PI and PID controllers.

6 CONCLUSIONS

In this thesis, a pole assignment self-tuning PID controller has been implemented and evaluated to control the electromechanical plant while the nominal system is assumed to be known. An approximated second-order plant model is used in the present design, since many of the industrial plants can be modeled using a second-order model. The process reaction curve method was used to predetermine the steady-state and transient behavior of the system such that the system was stable and did not give oscillatory response.

Classical Ziegler-Nichols open and closed-loop design and Cohen-Coon open-loop PID control design methods were reviewed and their basic futures are outlined. The control parameters to be set are presented in Table formats.

The idea behind self-tuning control was introduced and parameter identification using Least Square and Recursive Least Square methods based on ARMAX and ARX models was described.

Pole assignment control was explained and PID self-tuning control based on pole assignment approach was presented which is applicable to industrial control systems. It was also explained that the present pole assignment self-tuning control algorithm reduces a computational burden in comparison with the usual explicit pol-assignment self-tuning control algorithms, because it is not necessary to solve a Diophantine equation. And also, this control scheme has a feature that it enables us to construct the self-tuning control systems with taking account of the stability margin and the control performance.

The theoretical results were applied experimentally to an electromechanical plant in laboratory conditions. A data acquisition card was used to communicate between the control algorithm and the plant. The control algorithms were implemented in MATLAB and Simulink.

The experimental based plant model was obtained to calculate the approximate plant model. The model verification was performed such that the actual output response and approximated model output response were compared that the small modeling error was obtained. Conventional Ziegler-Nichols PID control and Cohen-Coon PID control parameters were calculated using the approximate model.

Pole-assignment self-tuning PID control was performed using the requirements and specifications already outlined. 20 msecs sampling period was used in self-tuning control and conventional control experiments.

The results were illustrated and time domain specifications were given in tables for comparison. The experiments were performed for tracking ability and regulatory behavior of the controllers. The overall results provided the advantages of the self-tuning PID control in the sense of fast response, less overshoot, smaller rise and settling times, and smaller output variations in magnitude.

Based on the experimental results and the time domain specifications presented, it can be concluded that the control performance of the electromechanical plant was significantly improved with the present self-tuning PID controller compared with the conventional PID control for both tracking and regulatory behaviors. Experimental results also confirm the fact that the self-tuning PID controllers are reasonable candidates to be used in industrial applications and these can be considered to be an alternative to conventional PID controllers, since it is simple to use and easy to understand and the computational task is not a problem any more because of rapidly increasing development in computer technology.

REFERENCES

ASTRÖM, K. J. and WITTENMARK, B., 1973, "On self-tuning regulators", Automatica, vol. 9, pp. 185-199.

ASTRÖM, K., HAGGLUND, T., 1995, "PID Controllers: Theory, Design, and Tuning", 2nd ed., Instrument Society of America.

BANYASZ, C. and KEVICZKY, L., 1982, "Direct methods for self-tuning PID regulators", in Proc. 6th IFAC Symposium on Identification and System Parameter Estimation, pp. 1249-1254.

BENNETT, S., 2001, "The past of PID controllers", Annual Reviews in Control, vol. 25, pp. 43-53.

BOBAL, V., BÖHM, J., FESSL, J., MACHACEK, J., 2005, "Digital Self-Tuning Controllers: Algorithms, Implementation and Application", Springer.

BÖHM, J., HALOUSKOVA, A., KARNY, M. and PETERKA, V., 1984, "Simple LQ self-tuning controllers", in Preprints of 9th IFAC World Congress, vol. 8, pp. 171–176.

CLARKE, D. W. and GAWTHROP, P. J., 1975, "Self-tuning controller", Proc. IEE, vol. 122, pp. 929–934.

CLARKE, D. W., MOHTADI, C. and TUFFS, P. S., 1987, "Generalized predictive control-Part I. the basic algorithm", Automatica, vol. 23, pp. 137-148.

DOS SANTOS C., L., "Tuning of PID controller for an automatic regulator voltage system using chaotic optimization approach", doi: 10.1016, Chaos, Solutions & Fractals.

DUMONT, G. A., HUZMEZAN, M., 2002, "Concepts, methods and techniques in adaptive control", IEEE-ACC American Control Conference, vol. 2, pp. 1137-1150.

EKER, İ., 2006, "Sliding mode control with PID sliding surface and experimental application to an electromechanical plant", ISA Transactions; vol. 45, No. 1, pp. 109-18.

FONG-CHWEE, T., SIRISENA, H. R., 1988, “Self-tuning PID controllers for dead time processes”, IEEE Transaction on Industrial Electronic, Vol. 35, No.1, pp. 119-125.

GAWTHROP, P. J., 1980, “Hybrid self-tuning control”, IEE-Proceedings D, vol. 127, pp. 229-236.

GAWTHROP, P. J., 1986, “Self-Tuning PID Controllers: Algorithm and implementation”, IEEE Transaction on Automatic Control, vol. 31, pp. 201-209.

GOODWIN GRAEBE, SALGADO, 2000, “Classical PID Control”, Prentice Hall.

KALMAN, R. E., 1958, “Design of a self optimizing control system”, Trans. ASME, vol. 80, pp. 481-492.

KARA, T. and EKER, İ., 2003, “Non-linear closed-loop direct identification of a DC motor with load for low speed two-directional operation”, Electrical Engineering, vol. 86, No.2, pp. 87-96.

KARNY, M., HALOUSKOVA, A., BÖHM, J., KULHAVI, R. and NEDOMA, P., 1985, “Design of linear quadratic adaptive control: Theory and algorithms for practice”, Kybernetika, vol. 21.

KIRECCI, A., EKER, İ., DULGER, L. C., 2003, “Self-tuning control as conventional method”, Electrical Engineering 85, pp. 101–107, Springer-Verlag.

KIM, J., 2009, “Identification of lateral tyre force dynamics using an extended Kalman filter from experimental road test data”, Control Engineering Practice, vol. 17, pp. 357–367.

KRNETA, R., ANTIC, S. and STOJANOVIC, D., 2005, “Recursive least squares method in parameters identification of dc motors models”, Electrical Energy, vol. 18, No. 3, pp. 467-478.

PARASKEVOPOULOS, P. N., 1996, “Digital Control Systems”, Prentice Hall.

PAZ A. R., 2001, “The Design of PID Controller”, Klipsch School of Electrical and Computer Engineering.,

PETERKA, V., 1970, “Adaptive digital regulation of noisy systems”, in Preprints of the 2nd IFAC Symposium on Identification and Process Parameter Estimation, p. 6.2, Prague: UTIA CSAV.

PETERKA, V., 1984, “Predictor-based self-tuning control”, Automatica, vol. 20, no. 1, pp. 39–50. Reprinted in: Adaptive Methods for Control System Design, Editor M.M. Gupta, IEEE Press, 1986.

REN, T. J., CHEN, T. C., TSAI, M. C., 2008, “Motion control for a two-wheeled vehicle using a self-tuning PID controller”, Control Engineering Practice, vol. 16, pp. 365-375.

SASTRY, S., BODSON, M., 1994, “Adaptive Control: Stability, Convergence, and Robustness”, Prentice Hall.

SEBORG, D. E., EDGAR, T. F. and MELLICHAMP, D. A., 1989, “Process Dynamics and Control”, Wiley, New York.

SIEMENS, Application Data, 2005, AD353-119, Rev 1.

WELLSTEAD, P. E., EDMUNDS, J. M., PRAGER, D. I. and ZANKER, P. M., 1979, “Pole zero assignment self-tuning regulator”, International Journal of Control, vol. 30, pp. 1-26.

WELLSTEAD, P.E., ZARROP, M. B., 1991, “Self-Tuning Systems: Control and Signal Processing”, Wiley.

YAACOB, S., MOHAMED, F. A., 1998, “Real time self-tuning controller for induction motor based on pole assignment method”, SICE, Chiba.

YAMAMATO, T., KANEDA, M. and TANAKA, K., 1995, “A simple explicit self-tuning controller based on pole-assignment scheme and its applications”, IEEE Transactions on Industrial Electronics, Control, and Instrumentation, vol. 2, pp. 950-955.

YAMAMOTO, T., FUJII, K. and KANEDA, M., 1996, “A self-tuning PID controller and its application for an ethylene cracking furnace”, IEEE Transaction on Industrial Electronic, vol. 1, pp. 275-281.

YU, C.C., 1999, Auto tuning of PID controllers, Springer, Great Britain.

YÜCELEN, T., KAYMAKÇI, Ö., KURTULAN, S., 2006, "Adaptive PI-D controller using Ziegler-Nichols based self-tuning method's parameters for programmable logic controllers", Proceeding of Symposium on Intelligent Manufacturing System, pp. 381-393.

ZHONG, S., CHUNPENG, K., DAWEI, X., 2007, "Research of PID parameter self-tuning applied in temperature control system", The Eighth International Conference on Electronic Measurement and Instruments, pp. 360-363.

BIOGRAPHY

I was born in Adana, Turkey, in 1984. I completed my high school education in Adana and graduated from ADANA Technical High School. I completed university in Technical Education Faculty, department of Electronics education from Gazi University, Ankara, Turkey in 2006.

My areas of interest include robotics, automation, control systems, computer programming.