THESIS REPORT
Master's Degree

S Y S T E M S
R E S E A R C H
C E N T E R

Supported by the National Science Foundation Engineering Research Center Program (NSFD CD 8803012), Industry and the University

Adaptive PID Control of Nonlinear Systems

by R. Ghanadan
Advisor: G.L. Blankenship

M.S. 90-10

ABSTRACT

Title of Thesis: ADAPTIVE PID CONTROL OF NONLINEAR SYSTEMS

Name of degree candidate: Reza Ghanadan

Degree and Year: Master of Science in Electrical Engineering, 1990

Thesis directed by: Gilmer L. Blankenship
Professor
Electrical Engineering

Adaptive proportional-integral-derivative (PID) controllers are used to control model-following nonlinear systems with second order dominant dynamics.

An Adaptive PID-based controller is designed which drives the difference (error) between a process response and a desired model output to zero. The design approach is based on the model reference adaptive control technique. Lyapunov stability theory is used to analyze asymptotic stability of the error system and to obtain controller tuning rules. The proposed design scheme and

the resulting tuning rules are computed and justified for nonlinear time-varying and/or uncertain plants where the dominant dynamics are of second order. The uncertain nonlinearities are assumed to be bounded by a known constant or a function of the plant output and its derivative. The disturbances in the system are dealt with in the design scheme and they are assumed to be bounded by a known parameter. It is shown that the error approaches zero asymptotically at a rate constrained by the desired model's characteristics. Simulation results for various nonlinearities show that the proposed tuning technique performs very well.

ADAPTIVE PID CONTROL OF NONLINEAR SYSTEMS

by

Reza Ghanadan

Thesis submitted to the Faculty of the Graduate School
of The University of Maryland in partial fulfillment
of the requirements for the degree of
Master of Science
1990

Advisory Committe:

Professor Gilmer L. Blankenship, Chairman/Advisor
Professor William S. Levine
Professor Thomas J. McAvoy

ACKNOWLEDGEMENTS

I am deeply grateful to my advisor, Professor Gilmer L. Blankenship, for his encouragement, guidance, and support during the past two years of my graduate research. It is both a privilege and pleasure to be his student.

I would like to thank the Systems Research Center for the financial support through a graduate fellowship and for providing an invigorating environment to explore new ideas. I thank the faculty and staff of the Department of the Electrical Engineering at the University of Maryland at College Park.

I also thank Dr. W. Dayawansa for valuable and stimulating discussions.

I am especially grateful to my wife, Tara, for her constant love and support. She deserves the highest praise for her love, caring, and endurance. My parents deserve the warmest gratitude for their ever present support and encouragement, which I can never thank them enough for that. I would also like to thank my parents-in-law, Lani and Emmett, for their support and encouragement.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

The objective of this thesis is to present an adaptive Proportional-Integral-Derivative (PID) controller tuning technique for nonlinear dynamical systems.

1.1 Background and Motivation

PID-based controller design is the predominant design method in the automatic control industry. PID controllers are used extensively in the chemical process industry, electronic and electrical systems, autopilots for aircrafts, missiles, and ships, industrial robots, etc. [Smi72,RR82,WR88,Bes89,CPP84]. Their popularity is due to their robustness in a wide range of operating conditions, the simplicity of their structure, as well as the familiarity of designers and operators with the PID algorithms. They are inexpensive to implement and reasonably sufficient for many industrial control systems needs.

Despite their popularity, PID controllers are usually poorly tuned in practice, with most of the tuning done manually through a lengthy trial and error procedure. This makes the tuning process difficult and time consuming for process engineers. In many situations the PID controller is replaced by a PI controller by switching off the derivative action [Tin89a,Tin89b,AH88]. This sacrifices performance and operating efficiency for an easier and faster tuning procedure since it is particularly difficult to tune the derivative action.

To address this problem, several techniques have been introduced to tune PID controllers [ZN42,ZN43,HW50], some of which will be reviewed in Chapter 2. If the performance requirements are not high, these *classical* PID tuning methods are sufficient for many control systems. As the demand on control performance and process economy increased, and systems with more complex structure must be controlled, efficient tuning methods are needed. Advances in industrial electronics and microprocessor technology have made possible the development of a wide range of PID *"autotuning"* methods [AH88,Fie62,N+84, AH84] and instruments. These systems have appeared in the market place [Har90,Mor87,Kom89].

These autotuners adjust the PID controller (i.e. generate PID parameters) *automatically* on demand from a process operator. They do not provide *continuous* on-line tuning, but they can be effectively used for initial tuning called *"pretuning"* and manual on line "retuning" in case it is needed. Although

PID controllers with autotuning capability can save time for process engineers and ensure a better control than ill-tuned standard PID controllers, they can not cope with changes in the dynamic characteristics of the process under control, structural perturbations, and environmental variations. Moreover, many control engineering problems require control in the presence of uncertainties, disturbances, and unforeseen changes in the process parameters and input signals. Since the PID parameters are fixed once they have been defined in the autotuning phase until the next retuning of the controller, changes and disturbances degrade the performance of the controlled process. In some cases it eventually destabilizes the plant.

Another important point to consider is that autotuners adjust the PID parameters only at one *nominal* operating condition (set point). As the process changes its operating conditions from the nominal one, any process nonlinearity eventually leads to degradation of the control performance. To overcome these problems, it is necessary to adjust the PID parameters *continuously* and *automatically* while the controller is running. This is called "adaptive" tuning of PID controller or *"adaptive PID controller"*.

There has been considerable interest recently in automating the tuning of the PID controller with adaptive techniques [Gaw87,AW89]. Adaptive PID controllers have the following advantages:

1. Adaptive PID tuning is *faster* than manual tuning.

2. The controller can be tuned more *accurately*, and hence, process economy is improved.

3. The tuning is *automatic* and is made *systematically* when needed.

4. The controller can be applied to systems with time-varying parameters, systems where the parameters change significantly over the range of operating conditions, or to cases where the system is partially known for which a model can not be measured with sufficient accuracy.

5. The controller can cope with disturbances in the system.

Several continuous-time and discrete-time self-tuning PID algorithms for *linear* systems have been successfully introduced in recent years [Mor87,And81, KS85,GO68,Haw83,P+83,Gaw82,Gaw86]. However, due to their linear structure, they operate successfully only within the *limited* region of control that the process nonlinearities can be effectively ignored. Since nonlinearities are an intrinsic part of every real world process, there is a need for adaptive PID algorithms that can handle systems with completely or partially known *nonlinearities* without losing their simple control structure. This is the motivation behind the work presented in this thesis on adaptive PID control for nonlinear systems.

1.2 Preview

This thesis is divided into five chapters. In Chapter 2, we review the basics of PID controllers. Several tuning methods such as Ziegler-Nichols open loop and closed loop techniques are briefly described. "Autotuning" and "self-tuning" techniques with their applications to design of PID controllers are explored. In section 2.7.2, it will be shown analytically that PID control is sufficient for systems with second order dominant dynamics. Some PID tuning techniques can be applied to higher order processes by approximating the process dynamics with a second order model over a given operating region either through on-line identification and estimation or off-line modeling techniques. The main drawback of these tuning methods is that they are developed for controller design of *linear* systems.

Our main results on PID controller design for *nonlinear* processes are introduced in Chapter 3. The design approach is based on a *model reference adaptive control* (MRAC) technique. The process model is assumed to be *partially* known. In the SISO case, systems with dominant dynamics of second order are considered. The disturbances in the system are assumed to be *bounded* by a known parameter. The *uncertain nonlinearities* in the system are assumed to be *bounded* by a *known* measurable function of the system output and its derivative. The MIMO case is simply a generalization of the SISO case where the system is composed of several *interactive* subprocesses each of which being

controlled by a PID controller.

A major weakness of traditional PID controllers is that they only employ the error signal $e = r - y$ within a scheme generally known as "system with error feedback." An adaptive PID controller, in contrast, employs *several* signal paths that are naturally available in an *adaptive loop.* The extra *degrees of freedom* can substantially increase the control capability of a PID controller whose P, I, and D terms function independently on various signal paths in the adaptive control loop. The design of such an adaptive PID controller requires the decision on the *architecture* of the controller (which term acts on what signal) and the *tuning* of each term. Both of these issues are considered in section 3.5.

Chapter 4 is devoted to the performance evaluation of the adaptive PID controller through simulation of several systems with nonlinear terms.

In Chapter 5, some directions for future work are suggested. Since PID controllers are very common in the control industry and very familiar to designers, this thesis aims to reduce the current gap between adaptive control theory and industrial control practice.

Chapter 2

Design of PID Controllers

2.1 Introduction

This chapter gives an introduction to PID control and reviews several conventional techniques for PID tuning. In section 2.2 we present the basics of standard PID algorithms. Implementation issues such as anti-windup, limitation of derivative gain and etc. are discussed in section 2.3. Section 2.4 describes different methods to determine a simple process model. These methods are based on either transient response of the process (i.e. the response of the system to pulses, steps or other test input signals) or frequency response function (i.e. the steady state response of the system to a sinusoidal input signal with changing frequency ω.)

Once a model has been obtained for the process dynamics, the PID parameters may be computed using some tuning methods. Several common PID tuning

techniques are reviewed in section 2.5. "Autotuning" is briefly discussed in section 2.6. In section 2.7, we give a short introduction to "self-tuning" regulators (STR) and review their application to PID controller design. Finally, some general notes concerning PID controllers are summarized in section 2.8.

2.2 PID Control: Preliminaries

The PID controller is the most common control algorithm in use today. According to Cegelec, a French control systems manufacturer, PID algorithms are applied to more than 90% of applications [Bou89]. Despite their wide applications, there are no industry-wide standard definitions for PID algorithms. According to Astrom [AH88], the "text book" version of the PID control algorithm has the form:

$$\begin{aligned} u(t) &= K[e(t) + 1/T_i \int_0^t e(\tau)d\tau + T_d \frac{de(t)}{dt}] \\ e(t) &= r(t) - y(t) \end{aligned} \tag{2.1}$$

where u is the control variable, e is the set point, and y is the measured value of the process output. The proportional gain K, integral time constant T_i, and derivative time constant T_d are the controller parameters. This version of PID algorithm is also called the "ideal noninteracting PID controller" or "ISA algorithm" [Ger87].

Other forms of PID algorithms used by manufacturers are [Ger87]:

- Interacting PID Controller:

$$u(t) = K' \left[e(t) + 1/T_i' \int_0^t e(\tau)d\tau\right] \left[1 + T_d' \frac{de(t)}{dt}\right] \tag{2.2}$$

- Ideal Parallel PID controller:

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau)d\tau + K_d \frac{de(t)}{dt} \tag{2.3}$$

For example, Foxboro and Fisher PID controllers use the noninteracting versions while Honeywell and Texas Instruments use the interacting algorithm [Ger87].

Different algorithms can be *manually* tuned more easily for different processes and work better in different situations. With manual tuning, choosing the best version for a process at hand depends on the specific control demands and objectives. Analytically, by redefining the PID coefficients, (2.1) is equivalent to (2.3). Also (2.2) can always be replaced by (2.1), but (2.2) is corresponding to (2.1) only if $T_d < 1/4\ T_i$. Alternative conventional PID algorithms and their relationship are discussed in detail in [Ger87,AH88].

As we see from the above equations, the majority of controllers operate on an *error* signal which is generated on-line by subtracting the process output y from the set point r. The resulting control system is shown in Figure (2.1) where l and n denote load disturbances and measurement noise respectively. The control system in this setting is called "system with error feedback" [AH88] and might be expected to meet several general requirements such as: closed loop stability,

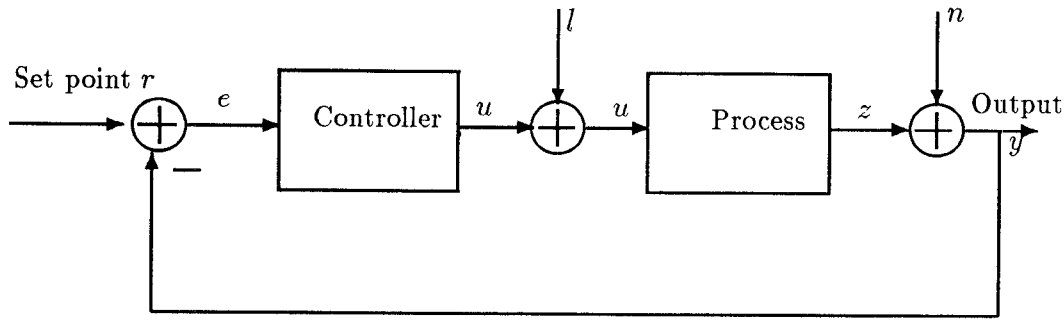

Figure 2.1: Control System with Error Feedback

good transient response to set point changes, load disturbance rejection, and noise rejection.

One discrete equivalent of (2.1) is derived by simply replacing the integral by a sum and the derivative by a finite difference such as:

$$u(n) = K \cdot [e(n) + \frac{h}{T_i} \sum_{j=0}^{n} e(j) + \frac{T_d}{h}(e(n) - e(n-1))] \qquad (2.4)$$

where h is the sampling period which is assumed to be small, and n is the current time index. Assuming $e(j) = 0 \ \forall j < 0$ and using the relationship:

$$\frac{1}{1 - q^{-1}} = 1 + q^{-1} + q^{-2} + \cdots$$

where q^{-1} is the backward shift operator defined by:

$$q^{-i}e(k) = e(k - i)$$

We can rewrite (2.4) as:

$$u(n) = [K + \frac{K \cdot h}{T_i} \cdot \frac{1}{1 - q^{-1}} + \frac{K \cdot T_d}{h}(1 - q^{-1})] \cdot e(n) \qquad (2.5)$$

Important considerations in computing the discrete equivalents of PID algorithm are choosing the numerical integration technique and the derivative estimation method. These considerations together with some modifications of (2.1) will be briefly discussed in section (2.3).

To simplify the writing, we use the following notation for PID algorithms:

$$u(t) = P(t) + I(t) + D(t) \tag{2.6}$$

where:

P : Proportional term

I : Integral term

D : Derivative term

Also:

$$P_s = K_{p_s} S(t) \ , \quad I_s = K_{i_s} \int_0^t S(\tau) d\tau \ , \quad D_s = K_{d_s} \frac{dS(t)}{dt} \tag{2.7}$$

In particular, 2.1 and 2.3 can be rewritten as:

$$u(t) \ = \ P_e \ + \ I_e \ + \ D_e \tag{2.8}$$

In the rest of this thesis, by "design" we mean the *structure* used in P, I, and D terms and by "tuning" we mean the *selection* of the numerical values of the parameters in P, I, and D terms.

A reader unfamiliar with the basics of PID controllers is recommended to read Chapter 2 in [AH88].

2.3 Implementation issues

To implement a PID control law on a digital computer we need to approximate the integral and the derivative terms. There are different ways to do this.

2.3.1 Integration routine:

One way to approximate the integral term I_s is to write it as [AH88]:

$$\frac{dI_s}{dt} = K_{i_s} \cdot S \tag{2.9}$$

and approximate the derivative by a finite difference:

$$\frac{I_s(n+1) - I_s(n)}{h} = K_{i_s} \cdot S(n)$$

hence, a recursive estimate follows:

$$I_s(n+1) = I_s(n) + K_{i_s} \cdot h \cdot S(n) \tag{2.10}$$

Another way is to use the trapezoid rule for integration: [Ise81,Smi72]

$$I_s(n) = K_{i_s} \cdot h \sum_{j=1}^{n} \frac{S(j) + S(j-1)}{2} = K_{i_s} \cdot h \left[\frac{S(0) + S(n)}{2} + \sum_{j=1}^{n-1} S(j)\right]$$

$$I_s(n+1) = K_{i_s} \cdot h \cdot \frac{S(n+1)}{2} + I_s(n) \tag{2.11}$$

2.3.2 Anti-Windup

A common modification in many PID controllers is the *"anti-windup"* feature of the integral term used in the algorithm. The purpose of this modification

is to prevent integrating the error when the control variable *saturates*. This feature can be thought of as the usual I term followed by a saturation element inserted into the algorithm [B+84]. It can also be described by (compare to (2.9)) [AW89]:

$$\frac{dI_e}{dt} = K_{i_s} \cdot e(t) + \frac{1}{T_\tau}(v(t) - u(t)) \tag{2.12}$$

where $v(t)$ is the output of the corresponding saturating actuator and T_τ is the "tracking time constant" usually chosen small (a fraction of T_i in (2.1)) so that the integrator can be reset quickly [AH88,AW89].

2.3.3 Derivative Approximations and Filtering

In most PID controllers such as those described by (2.1), (2.2), and (2.3), the derivative action operates only on the *measurement signal* y and not on the set point signal r. The purpose of this modification is to prevent drastic changes in the D term due to abrupt changes in the set point.

A more important modification is to filter the derivative action with a first order or a second order filter to lower the derivative noise. This feature *limits* the high frequency measurement noise amplification in the controller output. The control signal will be *less noisy* and the high frequency gain will stay within an appropriate bound. A derivative term in 2.1 with a first-order filter will then look like:

$$T_d/N \cdot \frac{dD}{dt} + D = -K \cdot T_d \cdot \frac{dy}{dt} \tag{2.13}$$

where N is the bound on the derivative gain, and T_d/N is the filter time constant. A typical value for N is 10 [AW89].

Respectively, modification to D_s in (2.7) is:

$$\tau \frac{dD_s}{dt} + D_s = K_{d_s} \frac{dS}{dt} \tag{2.14}$$

To approximate the filtered derivative term in (2.14) one can choose among the following realizations:

- Forward Difference method:

$$\tau \frac{D(n+1) - D(n)}{h} + D(n) = K_{d_s} \frac{y(n+1) - y(n)}{h}$$

$$D(n+1) = (1 - h/\tau) \cdot D(n) + \frac{K_{d_s}}{\tau}[y(n+1) - y(n)] \tag{2.15}$$

- Backward Difference method:

$$\tau \frac{D(n) - D(n-1)}{h} + D(n) = K_{d_s} \frac{y(n) - y(n-1)}{h}$$

$$D(n) = \frac{\tau}{\tau + h} D(n-1) + K_{d_s} \cdot h[y(n) - y(n-1)] \tag{2.16}$$

- Tustin's approximation method:

$$\frac{D(s)}{y(s)} = G(s) = \frac{K \cdot S}{\tau \cdot S + 1}$$

$$\frac{D(z)}{y(z)} = G(z) = G(s)\Big|_{s = \frac{2}{h}\frac{(1-z^{-1})}{(1+z^{-1})}}$$

$$G(z) = \frac{2K(1 - z^{-1})}{2\tau(1 - z^{-1}) + h(1 + z^{-1})}$$

$$2\tau D(n) - 2\tau D(n-1) + hD(n-1) + hD(n) = 2K[y(n) - y(n-1)]$$

$$D(n) = \frac{2\tau - h}{2\tau + h} D(n-1) + \frac{2K}{2\tau + h}[y(n) - y(n-1)] \qquad (2.17)$$

The recursions methods in (2.16) and (2.17) are always stable, however, (2.15) is unstable for small h (sampling time). Equation (2.16) gives sufficient accuracy if h is very small. Tustin's approximation gives the best results in general; and as h gets smaller, higher accuracy can be reached [Tza85].

2.3.4 Proportion Action Modification:

The system with error feedback described in section 2.2 works purely on error feedback (one degree of freedom), however, by separating the signal path for the set point and the process output in the proportional term of the control law (two degrees of freedom), the design flexibility increases which ultimately leads to a better control performance.

The desired modification is partially achieved if we modify the error e in the proportional part of equation 2.1 by:

$$e_p = \beta r - y \qquad (2.18)$$

where β is a constant design parameter. The closed loop transient response can then be improved by manipulating β.

More degrees of freedom for the control system means more parameters need

to be tuned. In conventional PID controllers, β is usually not subject to the "tuning" process. If we treat it like the other PID parameters K_p, K_i, and K_d, we must tune four parameters instead of three, which may not be easy.

In Chapter 3, we will see that there are more signal paths for an adaptive controller (higher degree of freedom) and a systematic procedure to tune the corresponding parameters, including β, is presented.

2.3.5 Noise Rejection, Sampling, and Prefiltering

As shown in Figure (2.1), in all industrial control systems, the process output z is always corrupted with noise. To estimate z, it is necessary to utilize a *noise-rejection* scheme in the control system. One effective scheme is the two step analog/digital filtering technique which is common in *digital implementation* of PID controllers. The analog filter, called a *prefilter* or *antialiasing filter*, is used before sampling the output z. Its purpose is to eliminate all *high frequency* signal components with frequency above half the sampling frequency. This is necessary because a high frequency disturbance appears, after sampling, as a low frequency signal (*aliasing* effect).

The *antialiasing filter* is usually a first-order lag or second-order Butterworth filter [AH88,Smi72]. For a first-order prefilter with time constant τ, a common rule-of-thumb is to choose $\tau = h/2$ where h is the sampling time [Smi72]. A

second order Butterworth prefilter:

$$B(s) = \frac{{\omega_c}^2}{s^2 + \sqrt{2}\,\omega_c s + {\omega_c}^2} \tag{2.19}$$

can also be used, where ω_c is the prefilter bandwidth. As an example [AH88], choosing:

$$\omega_c = \omega_s/8$$

where $\omega_s = 2\pi/h$ is the sampling frequency, results in a prefilter that attenuates signals by a factor of 16 at half the sampling frequency $\omega_s/2$.

2.4 Process Modeling

As mentioned in section 2.2, conventional PID controllers are tuned manually. There are, however, some simple tuning methods in which first, a simple process model is determined for the process to be controlled, and then the PID parameters are chosen. In this section we summarize several short cuts to develop a *simple parametric process* model for linear systems. In such systems, the shape of the output does not depend on the magnitude of the input.

2.4.1 Transient Response Methods:

Transient response techniques are usually based on open loop response of a given process to a test signal. The most convenient and commonly used test signal

is a step function. Many industrial processes have one of the open loop step responses shown in Figures (2.2), (2.3) , or (2.4).

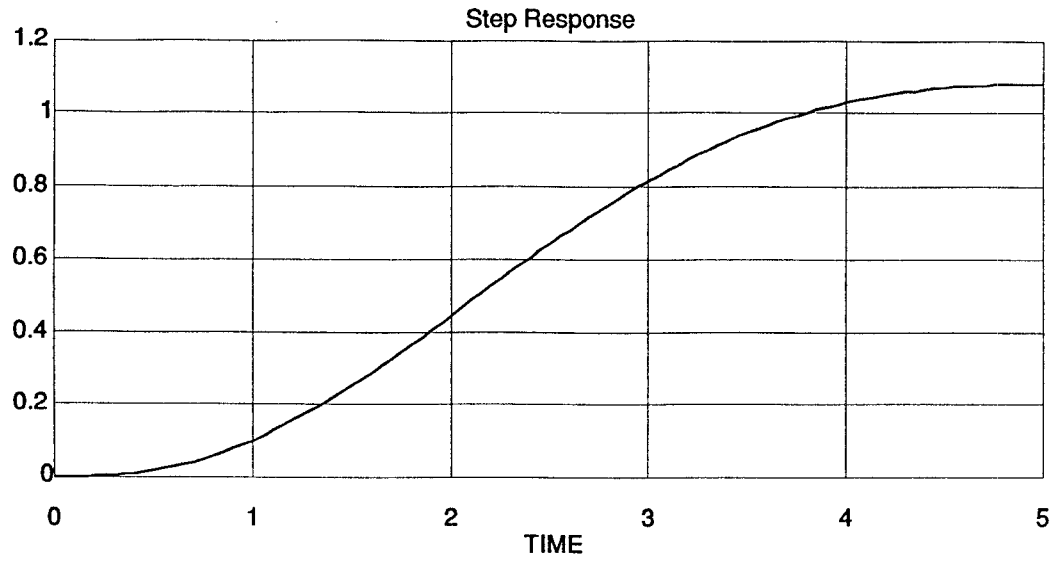

Figure 2.2: Step Response of a Self-Regulating Process

Self-regulating systems are those that have a step response shown in Figure (2.2) [Smi72], where, after a step input, the process reaches a new level. Temperature control processes are examples of this type. A simple model for such a system can be described by the transfer function (first-order lag plus deadtime):

$$G(s) = \frac{K}{1+\tau s} \cdot e^{-sL} \tag{2.20}$$

$$m = K \cdot \frac{L}{\tau}$$

where K is the process gain, m is the unit reaction rate, τ is the time constant, and L is the deadtime (time delay). This model describes the process behavior at the time scale L, where L approximates high order time constants in the

process.

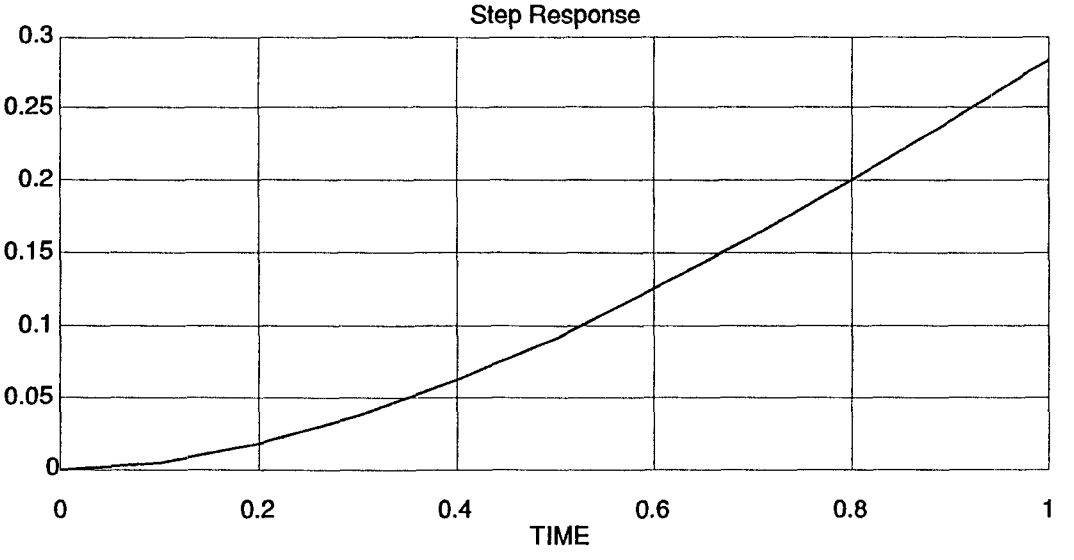

Figure 2.3: Step Response of a Nonself-Regulating Process

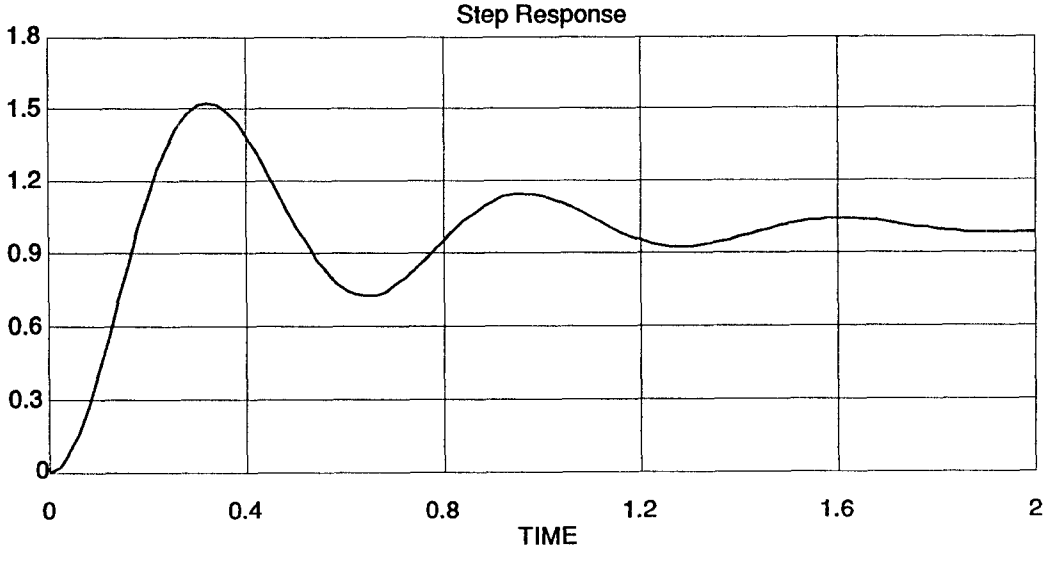

Figure 2.4: Step Response of an Oscillatory Process

Determination of three parameters K, τ, and L is based on the graphical construction shown in Figure (2.5) where a tangent to the step response is drawn.

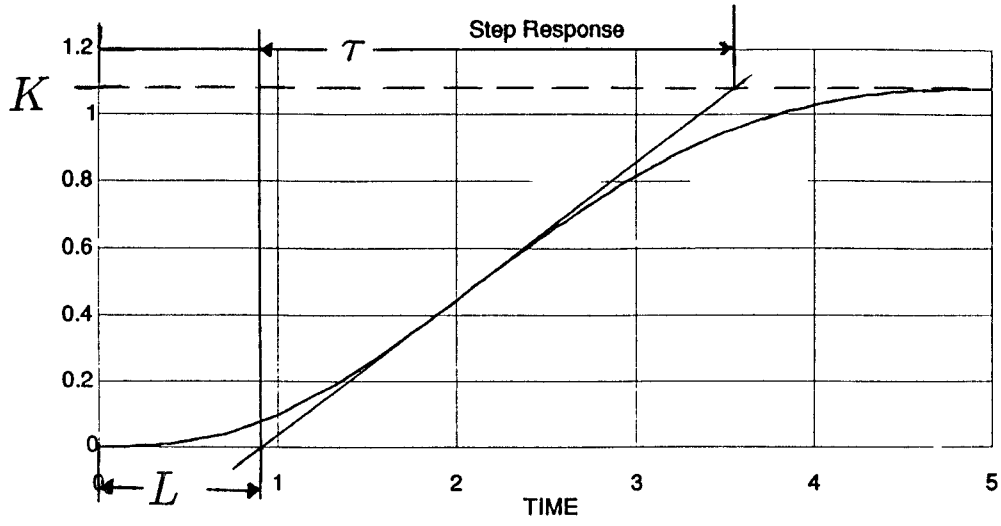

Figure 2.5: Determination of K, L, and τ for Self-Regulating Processes

Nonself-regulating processes have a step response of the type shown in Figure (2.3) where a step input to the process causes their outputs to increase or decrease indefinitely. Level control processes are of this type and they usually contain an integral term plus several time constants [Smi72]. The dynamics of these processes can be approximated by the transfer function:

$$G(s) = \frac{K}{s} \cdot e^{-Ls} \tag{2.21}$$

$$R = K \cdot L$$

where L, the apparent time delay, approximates the process time constants, R

is the unit reaction rate, and K is the gain. These two parameters can also easily be obtained graphically from the step response of the process as shown in Figure (2.6). There are alternative ways to find these parameters which are described in [AH88,Smi72].

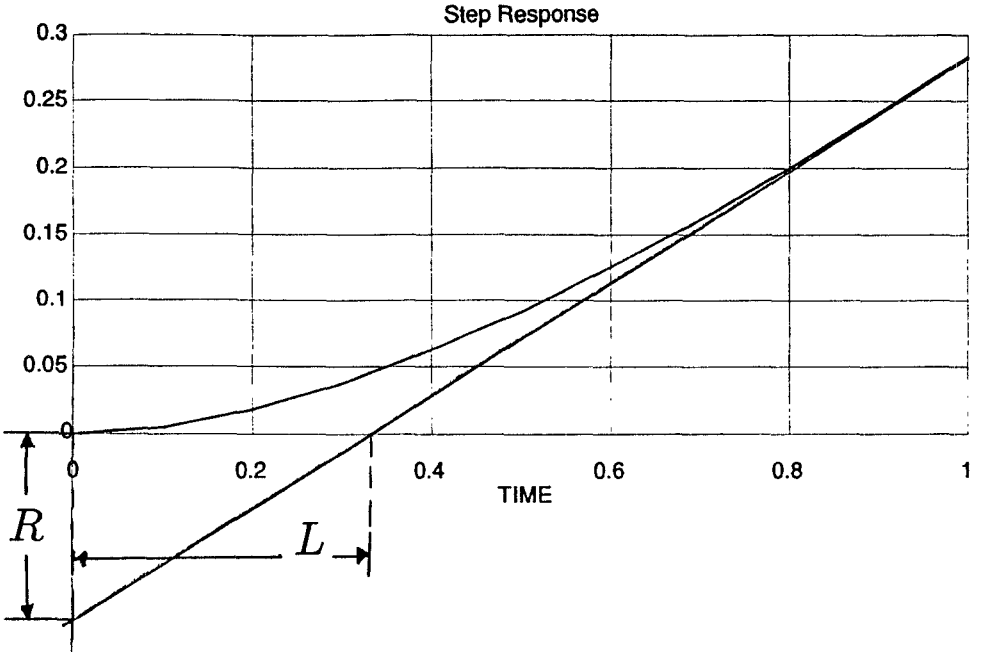

Figure 2.6: Determination of K and L for Nonself-Regulating Processes

Oscillatory systems have a step response of the form shown in Figure (2.4), and their dynamics can be approximated by the transfer function:

$$G(s) = \frac{K \cdot \omega^2}{s^2 + 2\zeta\omega s + \omega^2} \tag{2.22}$$

where the model parameters, natural frequency ω, gain K, and relative frequency ζ, can be approximated graphically from Figure (2.7) and the following

relationships [AH88]:

$$\begin{cases} \zeta &= \left[\sqrt{1 + (2\pi/\ln d)^2}\right]^{-1} \\ \omega &= \frac{2\pi}{T_p \cdot \sqrt{1-\zeta^2}} \end{cases} \tag{2.23}$$

where d is the damping and T_p is the period of oscillation as shown in Figure (2.7).

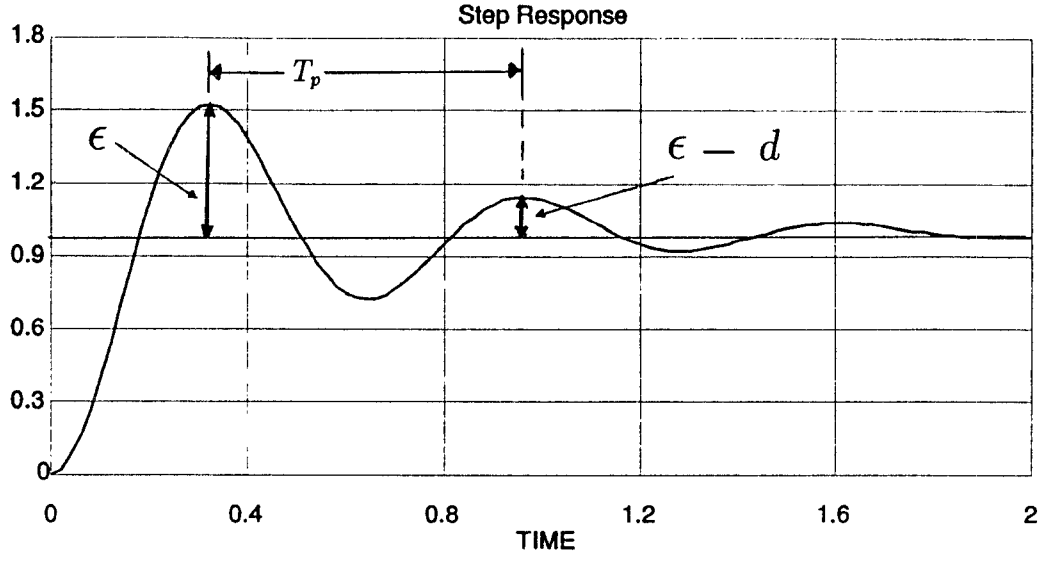

Figure 2.7: Determination of d and T_p for Oscillatory Processes

An alternative form of (2.22) which includes an apparent time delay L is:

$$G(s) = \frac{K \cdot \omega^2}{s^2 + 2\zeta\omega s + \omega^2} \cdot e^{-sL} \tag{2.24}$$

where L can be determined in the same way it was for the model in (2.20).

2.4.2 Frequency Response Methods:

In section 2.4.1 the process dynamics were approximated through a *transfer function* $G(s)$, here we use only those *parts* of the *Nyquist curve* that give the essential dynamical characteristics of the process sufficient for tuning the PID controller. The basic idea in most of these methods is to bring the process into self-oscillation (a limit cycle) using feedback at some appropriate frequency. The frequency of this limit cycle is called the *"crossover frequency"* (ω_c) and is the lowest frequency where the Nyquist curve of the open loop system intersects the negative real axis:

$$arg\ G(i\omega_c) \ = \ -\pi \tag{2.25}$$

where $G(s)$ is the open loop transfer function of the given process. The period $T_c \ = 2\pi/\omega_c$ is called the *"ultimate period"*. Another important term is the *"ultimate gain"* K_c which brings the system to the stability limit under pure proportional control. The relation between the ultimate period T_c and the ultimate gain K_c is given by:

$$K_c \ = \ \frac{1}{\left|G\left(i \cdot \frac{2\pi}{T_c}\right)\right|} \tag{2.26}$$

As shown in Figure (2.8), the point on the Nyquist curve corresponding to ω_c is called the *critical point.*

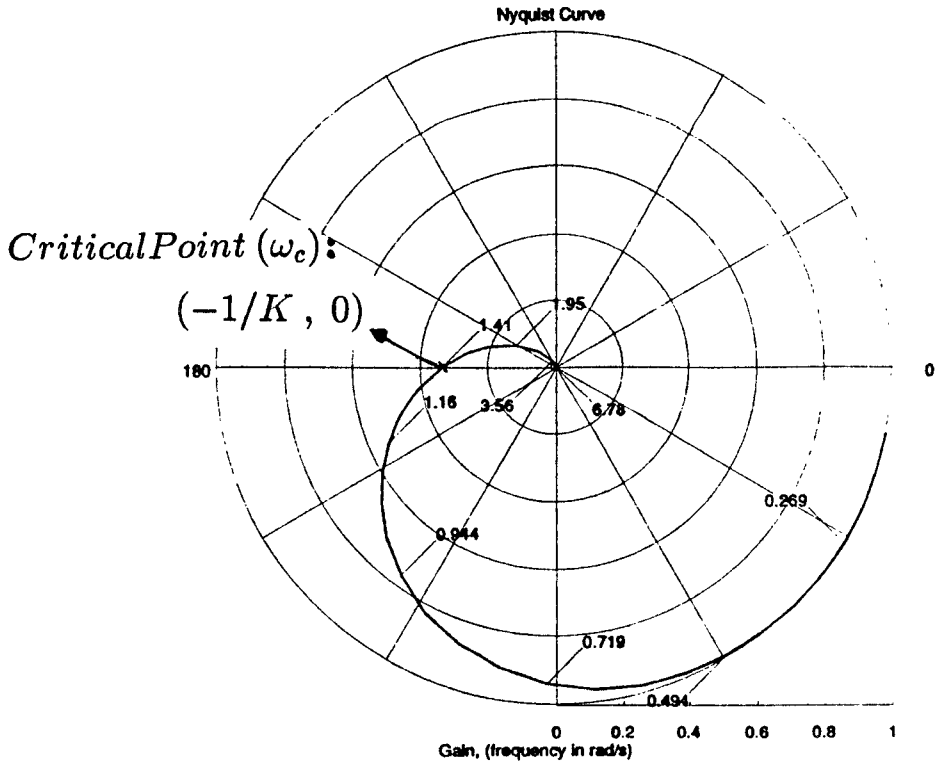

Figure 2.8: Nyquist Plot-$\omega_c \approx 1.41$

There are several ways to determine the critical point and the ultimate period T_c. Two of the most common methods are briefly reviewed below. The reader is referred to [ZN42,ZN43,HW50,Fie62,N+84,AH84,War88a,Smi72,AH88] for more details. Ziegler and Nichols [ZN42,ZN43] have suggested an experimental technique to determine the ultimate gain and period based on the observation that many systems may be brought to a stability boundary under pure proportional feedback by choosing sufficiently *high gain* K, see Figure (2.9).

At this boundary, the process input u and output y are *sinusoids* with a phase shift of -180 degrees. The frequency of the oscillation is the crossover frequency ω_c, and the gain K that brings the system to the stability boundary

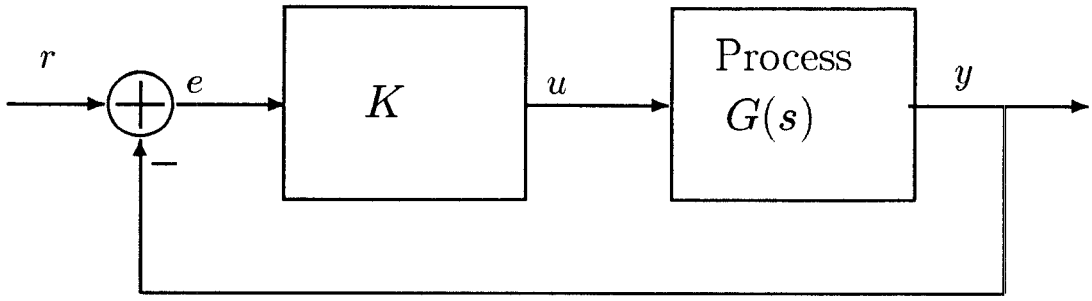

Figure 2.9: Ziegler and Nichols Stability Experiment

is the ultimate gain K_c. Notice that the approach is *manual.*

Hagglund and Astrom [AH84] have provided a method to approximate the critical point on the Nyquist curve which is based on a relay feedback experiment. As shown in Figure (2.10), a limit cycle oscillation is forced on the process using a relay described by:

$$u = \begin{cases} \delta & \text{if } e > 0 \\ -\delta & \text{if } e < 0 \end{cases} \tag{2.27}$$

where δ is the relay amplitude. The condition for oscillation in Figure (2.10) is that the linear process $G(s)$ has a Nyquist curve that intersects the negative real axis.

The ultimate period T_c is the same as the period of the limit cycle oscillation, and the ultimate gain K_c is the relay gain given by:

$$K_c = \frac{4\delta}{\pi d} \tag{2.28}$$

where d is the amplitude of the oscillation in the error signal e.

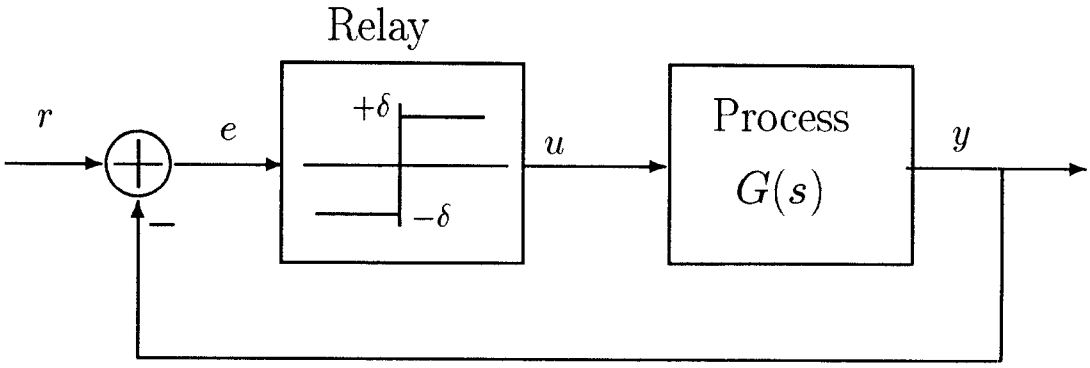

Figure 2.10: System with Relay Feedback

2.5 Conventional PID Tuning Techniques:

In this section we describe the well-known Ziegler-Nichols classical tuning methods for PID controllers presented in 1942. With some modifications, these techniques are still very common in control industry. See [ZN42,ZN43,AH88] for more details. These methods are based on many experiments performed on typical industrial plants.

In the Ziegler-Nichols *step response* method, first a simple process model is determined using the techniques described in the previous section. Then, the *"best"* PID parameters are directly computed from the relationships given by (2.29) and Table (2.1) bellow:

$$\begin{cases} T_c &= 4 \cdot L \\ K_c &= \frac{2}{m \cdot L} \end{cases} \tag{2.29}$$

Controller	K	T_i	T_d	T_p
P	$0.5K_c$			T_c
PI	$0.45K_c$	$0.83\ T_c$		$1.4\ T_c$
PID	$0.6K_c$	$0.5\ T_c$	$0.125\ T_c$	$0.85\ T_c$

Table 2.1: Ziegler-Nichols PID Tuning Rules

As we see in this table, PID parameters K, T_i, and T_d for the algorithm in (2.1) are given as functions of the ultimate period T_c and the ultimate gain K_c which are estimated from the relationships in (2.29). Recall that the two process model parameters, delay L and unit reaction rate m, were obtained from a graphical construction of the open-loop process step response. (see section 2.4.1 for details). In the Table (2.1), T_p is approximately the period of the dominant dynamics of the closed-loop system.

In the Ziegler-Nichols *frequency response* method, the parameters K_c and T_c are first determined from the *stability experiment* described in section 2.4.2. Then, from Table (2.1), the PID controller parameters K, T_i, and T_d are computed. Since in many situations it is not practical to perform the Ziegler-Nichols stability experiment on a process, other experiments such as the relay excitation technique described in section 2.4.2 may be used.

Note that the Ziegler-Nichols tuning technique results in an amplitude margin of 1.5 and phase margin of 25 degree which gives *quarter amplitude damping* (i.e. $d = 0.25$ in Figure (2.7)) and *relative damping* $\zeta = 0.22$ in equation (2.22). These characteristics lead to good process responses to load disturbances. Other

objectives can be met by modifying Table (2.1). For example, Table (2.2) gives the recommended PID parameters to achieve an amplitude margin of at least 2 and phase margin of at least 45 degrees [AH88].

Controller	K	T_i	T_d
PID	$0.35K_c$	$0.77\ T_c$	$0.19\ T_c$

Table 2.2: Modification to the Ziegler-Nichols PID Tuning Rules

The recommended PID parameters can be derived based on the idea of transferring one or more points of the Nyquist curve of the open-loop system to a desired position. In particular, the Ziegler-Nichols tuning method described above corresponds to moving the *critical point* of the open-loop Nyquist curve to the point $-0.6-0.28i$ of the closed-loop Nyquist curve. See [AH88] for details.

2.6 Autotuning

The purpose of a PID *auto-tuner* is to adjust the controller parameters *automatically* either on demand from a plant operator or an external signal. Broad acceptance of microprocess-based controllers has made possible PID controllers with auto-tuning capabilities. Some approaches to auto-tuning are discussed in [AH88,N+84,AH84]. A number of commercial auto-tuners are now on the market [Mor87,Har90,Kom89,Haw83]. They are mainly based on two different avenues:

- Process identification combined with computation of PID parameters through some design methodologies.

- Pattern recognition approach combined with some method to determine controller parameters values.

In the first method, the process dynamics are identified automatically through a modeling technique like those discussed in section 2.4. One of the most common approaches is the one presented by [AH84]. A block diagram of this autotuner is shown in Figure (2.11).

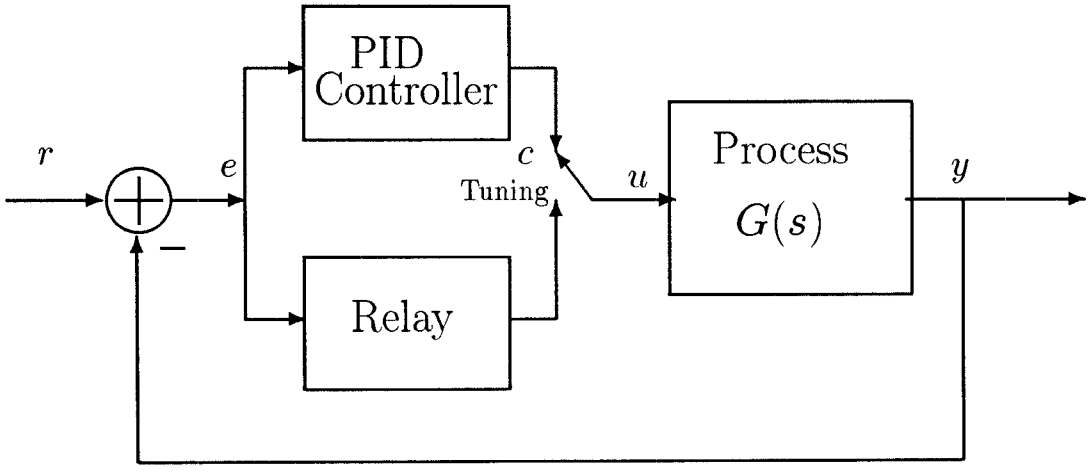

Figure 2.11: Block Diagram of an Autotuner Based on the Astrom and Hagglund Relay Experiment

In the tuning phase, the control loop is a relay feedback loop which looks like the one shown in Figure (2.10). As discussed in section 2.4, a periodic oscillation will be reached in this phase, where the period of the oscillation is simply the ultimate period T_c, and the relay gain (at the critical frequency)

$4\delta/\pi d$ is the ultimate gain (see section 2.4.2 for details). The period T_c may simply be determined automatically by measuring the times between the zero-crossings, and the amplitude d can be found by measuring the peak-to-peak values of the process output. After this identification procedure, the next step is to compute the values of the PID parameters using Table (2.1) or some modified rules programmed within the auto-tuner.

When the tuning is complete, the process is automatically switched to the PID controller (position C). The new PID parameters will remain fixed and are in effect till the next time the operator asks for tuning option.

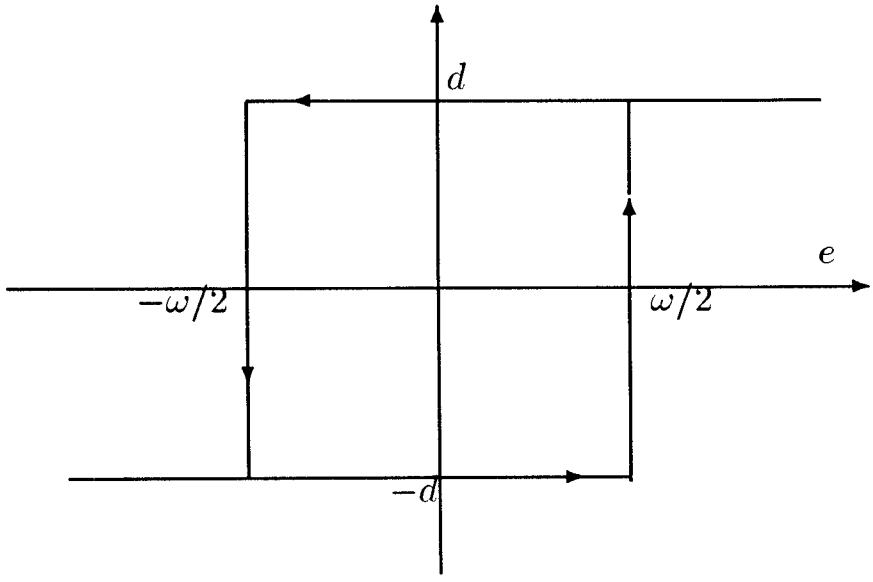

Figure 2.12: A Relay with Hysteresis

In order to reduce the effects of the *measurement noise* while estimating K_c and T_c the relay in Figure (2.11) is often replaced with a *relay with hysteresis* shown in Figure (2.12). The width w can automatically be set based on the

measurement noise level in the system.

Another common approach is to employ a recursive parameter estimator for process identification and apply a design scheme to the *estimated* parametric model in order to determine the PID parameters. If the identification procedure is performed *on-line*, the auto-tuner can easily be upgraded to an *adaptive tuner* where the PID parameters are updated continuously. A *self-tuning PID controller* based on this procedure is described in the next section.

The second methodology is based on a heuristic logic developed by engineers over many years, and it is based on the assumption that the disturbances can be approximated by steps and short pulses. One scheme, used in the Foxboro Exact controller, is the result of an expert system based on pattern recognition approach described by [Bri77]. In this method, a step change is applied to the system after some initializations. The auto-tuner then monitors the resulting transient error response and calculates key parameters: *damping* d, overshoot η, and the period of oscillation T_p defined as in Figure (2.13) where:

$$\begin{cases} d &= \frac{e3-e2}{e1-e2} \\ \eta &= \left|\frac{e2}{e1}\right| \end{cases} \tag{2.30}$$

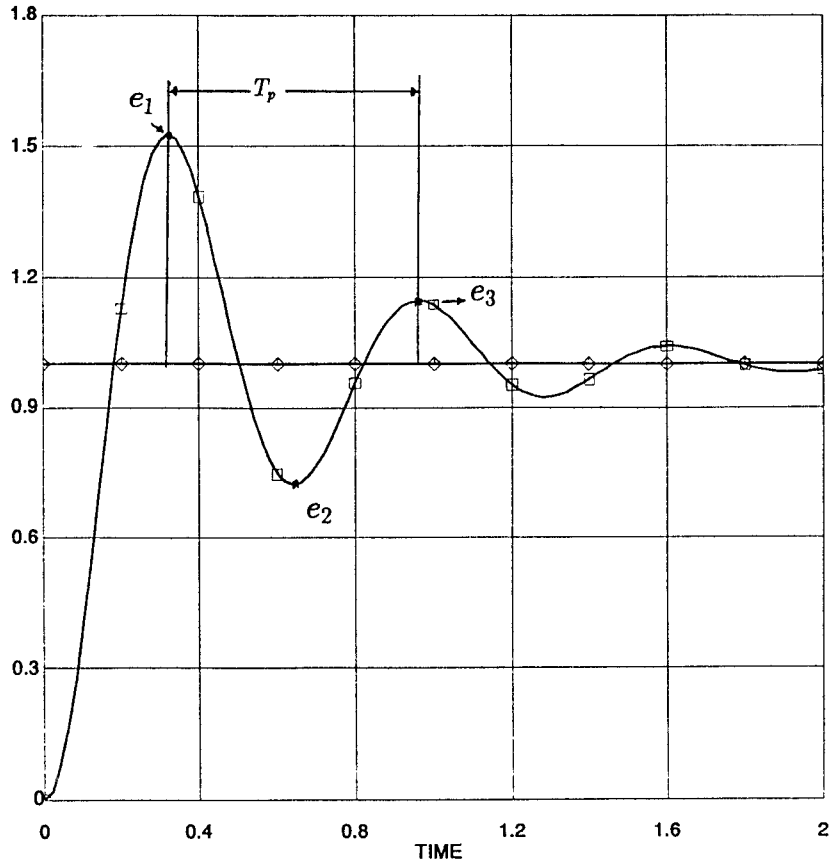

Figure 2.13: Determination of Damping d, Overshoot η, and Period T_p

The operator can specify the maximum overshoot η and the maximum damping d typically in the range $[0, 1]$ and $[0.1, 1]$ respectively [AH88]. The key parameters are then used in deciding how to adjust PID parameters through empirical rules developed by Foxboro partially described in [KM84]. For example, from Table (2.1) the simple Ziegler-Nichols tuning method suggests:

$$\begin{cases} \frac{T_i}{T_p} = \frac{0.5}{0.85} \\ \frac{T_d}{T_p} = \left|\frac{0.125}{0.85}\right| \end{cases} \tag{2.31}$$

where T_i and T_d are the respective PID parameters in equation (2.1). In practice, however, some modifications to the relations in (2.31) may be made, based on

process time delays, time constants, etc. After the tuning is done, the controller operates with fixed parameters until a disturbance with sufficient magnitude occurs. A tolerance is often introduced in the auto-tuner that determines when to trigger the tuning mode. The tolerance is typically twice the *noise band* preset by the operator [War88b].

2.7 Self-tuning PID Controller Design

There are basically two approaches to adaptive control: model reference adaptive control (MRAC) and self-tuning regulators (STR) [AW89,Ast83,Ast87,SB89, Cha87,Nar86]. The idea behind adaptive control is to adjust the controller parameters *automatically* and *continuously*, based on some on-line input/output measurements of the process under control. MRAC will be reviewed in the next chapter. In section 2.7.1, we briefly discuss STR. Applications of STR to PID controller design will be discussed in section 2.7.2.

2.7.1 Self-tuning Regulators (STR):

STR, as originally proposed by Kalman [Kal58] and later developed by Astrom and Wittenmark in 1973 [Ast80], is an approach to adaptive control in which it is assumed that there exists a constant but unknown *linear* parametric model underlying the process dynamics. It is, however, customary to apply

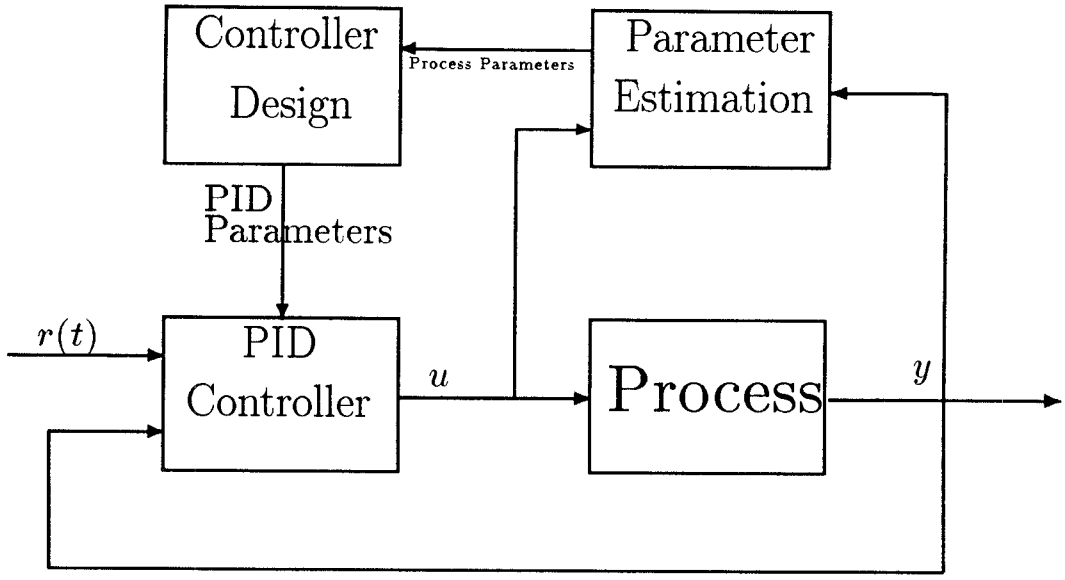

Figure 2.14: Self-Tuning Regulators

STR methodology to linear time-varying processes under the important condition that the changes in the process parameters are *slow* relative to the process dynamical response.

As shown in Figure (2.14), process parameters are estimated on-line using a recursive parameter estimator. The estimated values are used as the process parameters in a design scheme which computes the controller parameters.

Therefore, the block referred to as "controller design" gives the on-line solution to the design problem for a process with *known* parameters and is called *underlying design problem* [AW89]. The underlying design problem specifies the characteristics of the closed-loop system under the assumption that all the process parameters are known.

STR based on least squares estimation and minimum variance regulation

are investigated in [Ast80]. It should be noted here that for the parameter estimates to converge to the true values, the input signal to the process should be sufficiently "rich" and the closed-loop system must be stable [AW89]. Conditions for convergence in parameter estimation is therefore very important to the design problem.

Some modifications to STR have been proposed in [CG75,CG79,Gaw77] and implementation issues such as microprocessor-based STR have been explored in [CG81,GO68,Gaw82,Nom88,Gaw88,Moh88,War88b]. The following books provide details in theory, design, stability, algorithms, and implementation issues for STR: [War88a,Gaw87,HB81,AW89,Cha87,SB89,Nar86,GS84]. In the next section, we apply the STR formulation of Figure (2.14) to tune PID controllers.

2.7.2 Self-tuning PID Controller

A self-tuning PID controller, like any other STR, has an identification algorithm such as recursive least square estimator that provides the latest updates of the process parameters (see Figure (2.14)). Estimation techniques for STR are explored in detail in [Cla81,AW89,SB89,GS84]. Various identification and estimation schemes are proposed in many books such as: [AM79,Lju87,LS83,Nor86, Leo87,Ise81]. A well known identification technique used is the least-square estimator. The details are not considered here and may be found elsewhere [AW89,Cla81].

The estimated parameters can then be utilized within the STR underlying design problem, discussed in previous section, subject to the fact that the controller has a PID structure. A common approach to the design problem is the pole placement technique [OK84,Wit79]. The idea is to find the controller parameters such that the closed loop poles have the desired locations.

Assume that the process in Figure (2.14) is represented by the discrete-time ARMA model:

$$A \cdot y(t) \;=\; q^{-k}\, B \cdot u(t) \;+\; v \tag{2.32}$$

where v is a disturbance, $k > 1$ is the delay, and:

$$\begin{aligned} A &= 1 + a_1 q^{-1} + a_2 q^{-2} + \cdots + a_{n_a} q^{-n_a} \\ B &= b_1 + b_2 q^{-1} + b_3 q^{-2} + \cdots + b_{n_b} q^{-(n_b-1)} \end{aligned}$$

and q^{-1} is the backward shift operator (delay):

$$q^{-i} y(t) \;=\; y(t-i)$$

For the controller structure, we rewrite equation (2.5) as:

$$\begin{aligned} R \cdot u(t) &= S \cdot e(t) \\ e(t) &= r(t) - y(t) \end{aligned} \tag{2.33}$$

where:

$$\begin{cases} R &= 1 - q^{-1} \\ S &= k_1 + k_2 q^{-1} + k_3 q^{-2} \end{cases}$$

$r(t)$ is the set point, $e(t)$ is the error, and:

$$\begin{cases} k_1 &= K + \frac{K \cdot h}{T_i} + \frac{K \cdot T_d}{h} \\ k_2 &= -K - 2\,\frac{K \cdot T_d}{h} \\ k_3 &= \frac{K \cdot T_d}{h} \end{cases} \tag{2.34}$$

Applying an input of the form (2.33) to the process described by (2.32) gives:

$$\begin{aligned} A \cdot R \cdot y(t) &= q^{-k}\, B \cdot R \cdot u(t) + R \cdot v \\ &= q^{-k}\, B \cdot S \cdot r(t) - q^{-k}\, B \cdot S \cdot y(t) + R \cdot v \end{aligned}$$

Hence, the closed loop system is described by:

$$y(t) = \frac{B\,S}{T}\, r(t-k) + \frac{R}{T}\, v \tag{2.35}$$

where: $T = A\,R + q^{-k}\,B\,S$ is the closed loop characteristic polynomial. Assuming that the *desired* poles for the closed loop response are selected and given by T_m (desired closed loop characteristic polynomial), we need to find the S polynomial, i.e. k_1,k_2, and k_3 in (2.34), such that the following hold:

$$A\,R + q^{-k}\,B\,S = T_m \tag{2.36}$$

Since there are three unknowns (k_1, k_2, and k_3) and $k \geq 1$, T_m needs to be approximated at best by a 3rd order polynomial (assuming the constant terms have been equated by simple manipulations). Hence, the following must hold:

$$max\{\, 1 + n_a\,,\; k + n_b + 2\,\} \leq 3 \tag{2.37}$$

where n_a and n_b are orders of polynomials A and B.

Since $k \geq 1$, for (2.36) to hold, we must have:

$$\begin{cases} n_b &= 0 \\ k &= 1 \\ n_a &\leq 2 \end{cases} \tag{2.38}$$

Therefore, the proposed self-tuning PID scheme can completely cope with processes with the following dynamics:

$$y(t) = \frac{b_1}{1 + a_1 q^{-1} + a_2 q^{-2}} \cdot u(t-1) \tag{2.39}$$

with $b_1 \neq 0$. Conditions (2.38) and the form of the equation (2.39) simply state that processes of 1st and 2nd order with small delay are good candidates for PID controllers. PID controllers are also good for processes with first or second order *dominant dynamics* since their dynamics can be approximated by a second order model.

With assumption (2.38), equation (2.36) can be solved *automatically* for three PID parameters, i.e. K, T_i, and T_d in (2.34), on-line, based on recent estimates of process parameters in (2.39), i.e. b_1, a_1, and a_2. Note that in (2.34), h (sampling time) is known and is usually specified as a design parameter. The overall estimation and design computation and data-processing for the process given by (2.39) is fairly simple and can easily be programmed in microprocessors.

2.8 Conclusion

In this chapter we reviewed the basics of the PID control and discussed several modifications of the standard PID algorithm that are common in practice such as integral anti-windup, filtering, digital implementation , and proportional and derivative modifications.

Several conventional PID tuning techniques based on transient response and frequency response were presented. In tuning methods based on transient response, first a simple process model for the plant is obtained from the open loop step response of the process. Then, the PID parameters are determined simply from some recommended mostly empirical rules. In techniques based on frequency response methods, only the essential parts of the Nyquist curve of the open loop system are used. After identifying these parts, the PID parameters are computed again using some given recommended rules.

We also reviewed autotuning and stated that its main purpose is to tune the PID controller automatically on demand from an operator. We noted that autotuners do not provide continuous on-line tuning and consequently, the tuning phase is separated from the control phase. Upon any change in dynamic characteristics of the controlled process, environmental variations, operating point, or any structural perturbations, the process need to be retuned (i.e. switched

to the tuning phase). We mentioned that autotuners simply combine the techniques for identifying the process dynamics and the methods for determining the PID parameters.

Finally, we explored an application of adaptive control theory to the PID controller design. The idea of self-tuning regulators (STR) was introduced and from this, an adaptive PID design scheme which is based on pole placement technique was obtained. The methods described in this chapter are successful only over the limited operating region of the control where the nonlinearities in the process are not dominating and can effectively be ignored. Over this operating region, the process was assumed to be linear and time-invariant. Also, as it was discussed in section 2.7, PID control is adequate for systems where the dynamics are approximately of first or second order. If possible, approximations need to be made for higher order dynamics to incorporate them within a second order model over a selected operating region.

Chapter 3

Adaptive PID Design for Nonlinear Processes

3.1 Introduction:

In this chapter we develop a design scheme that employs simple PID controllers and *adaptive* methodologies to control *nonlinear processes.* The approach is based on a *model reference adaptive control* (MRAC) method responding to *bounded disturbances.*

Section 3.2 contains a brief discussion of the MRAC problem. In section 3.3, we describe the *form* of the nonlinear processes that we deal with in our approach. The statement of the design problem and some notation are introduced in section 3.4. The main results and adaptive laws are stated and derived in section 3.5. Some topics of interest such as the rate of convergence of the

algorithm and the choice of an adaptive gain vector are discussed in section 3.6.

In section 3.7, some modifications are made, the effect of integral action with *anti-windup* capability in the stability of the adaptive system is investigated, and the continuous version of the control law is presented. Section 3.8 generalizes our SISO result to MIMO case where several PID controllers need to be tuned at the same time in an *interactive* multiple loop environment.

3.2 Model Reference Adaptive Control:

In section 2.7, we stated that the "self-tuning regulator" (STR) is one of the main approaches to adaptive control. MRAC is another main approach to adaptive control which is based on *model-following*. That is, the controller parameters are adjusted so that the closed-loop behavior will be close to that of a prescribed model.

The basic ideas of MRAC are illustrated in Figure (3.1). In this figure, y_p is the process output to be controlled. The desired performance is expressed in terms of a *reference model* which gives the desired response y_m to an input command signal u_c. The error $y_m - y_p$ is formed on-line and continuously monitored through the block called "adjustment mechanism". This block generates the controller parameters based on the error in the adaptive system. Hence, some *updating rules* need to be specified within this block.

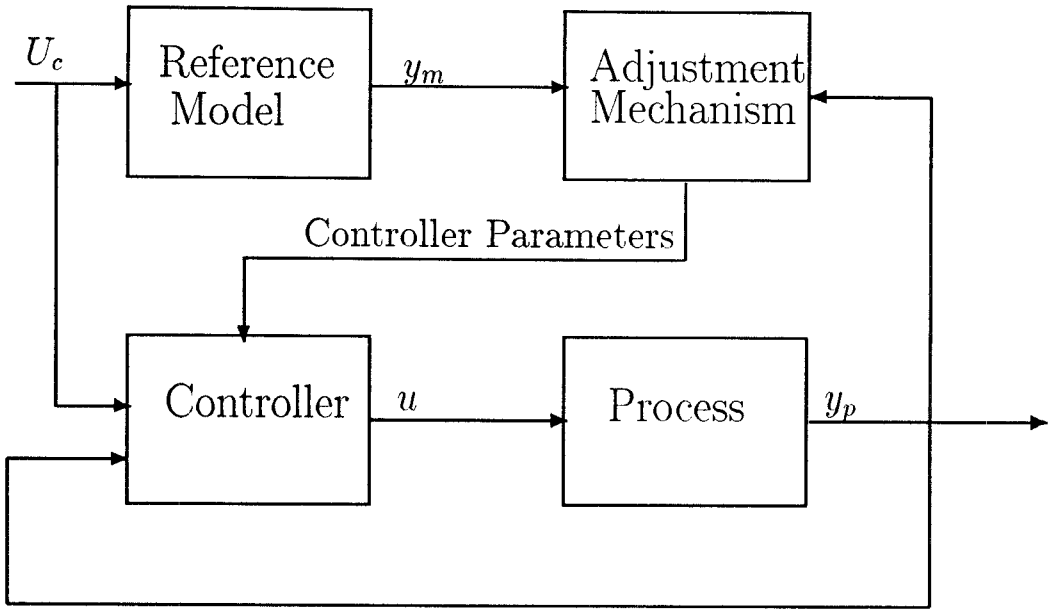

Figure 3.1: Model Reference Adaptive Control (MRAC)

There are three common approaches to update the controller:

- The gradient approach
- Lyapunov stability theory
- Passivity theory

These methods have been widely described in the adaptive control literature: [OWK69,Par81,Par66,Lan79,Moh88,NL80,N+85,Mon74,Mor80]. Our approach is based on Lyapunov stability theory. First, an error system is derived. Then, an adaptation mechanism is developed to guarantee that the error $y_m - y_p$ goes to zero.

The error system is obtained in section 3.4, and the adaptive PID parameters adjustment rules are derived in section 3.5.

3.3 System Description:

Even though PID controllers have a simple structure, they can control most industrial systems sufficiently provided that the demands on the performance of the control are not too high. In general, as it was discussed in section 2.7 and 2.8, PID control is sufficient for processes where the *dominant dynamics* are of *second order*. Since PID controllers have limited complexity, more complicated systems with high frequency dynamics should be controlled by more sophisticated algorithms. Our goal is to derive an adaptive PID tuning technique for a class of nonlinear dynamical systems modeled or approximated by the following nonlinear time-varying differential equation:

$$\ddot{y}_p = a_{p_1}(y_p, \dot{y}_p; t) \cdot \dot{y}_p + a_{p_0}(y_p, \dot{y}_p; t) \cdot y_p + f(\sigma; t) + b_p(y_p, \dot{y}_p; t) \cdot u + v \quad (3.1)$$

$$\sigma \stackrel{\text{def}}{=} g(y_p, \dot{y}_p, \ddot{y}_p, \ldots) \quad , \quad b_p \neq 0$$

$$|v| \leq v_{max} \quad (3.2)$$

where y_p is the process output, u is the control, a_{p_0}, a_{p_1}, and b_p are known nonlinear time-varying coefficients, v is a disturbance, $g(\cdot)$ and $f(\cdot)$ are *unknown* nonlinear functions. It is, however, assumed that there exists a measurable function $\rho(y_p, \dot{y}_p; t)$ such that:

$$|f(\sigma; t)| \leq \rho(y_p, \dot{y}_p; t) \quad \forall \sigma, y_p, \dot{y}_p, t \quad (3.3)$$

It should be noted that equation (3.1) does not restrict the class of actual nonlinear systems to which the controller can be applied to be second order. It only restricts the *nominal* system model to such a form. Moreover, if the high frequency dynamics are bounded through the inequality in (3.3), then no approximation is necessary.

The reference model, specified by the designer, which describes the behavior expected from the controlled process is identified by a second order linear, time-invariant, asymptotically stable differential equation:

$$\ddot{y}_m = a_{m_1}\dot{y}_m + a_{m_0}y_m + b_m u_c \tag{3.4}$$

where u_c is a piecewise continuous bounded input command signal to the reference model.

3.4 Notations and Problem Formulation:

By defining the state vectors as:

$$x_p = \begin{bmatrix} x_{p_1} \\ x_{p_2} \end{bmatrix} \stackrel{\text{def}}{=} \begin{bmatrix} y_p \\ \dot{y}_p \end{bmatrix} \quad , \quad x_m = \begin{bmatrix} x_{m1} \\ x_{m2} \end{bmatrix} \stackrel{\text{def}}{=} \begin{bmatrix} y_m \\ \dot{y}_m \end{bmatrix}$$

and the error vector as:

$$\bar{e} = \begin{bmatrix} e_1 \\ e_2 \end{bmatrix} \stackrel{\text{def}}{=} x_m - x_p = \begin{bmatrix} y_m - y_p \\ \dot{y}_m - \dot{y}_p \end{bmatrix}$$

we can rewrite equations (3.1) and (3.4):

$$\dot{x}_p \;=\; A_p(x_p;t)\cdot x_p \;+\; F(\sigma;t) \;+\; B_p(x_p;t)\cdot u \;+\; D \tag{3.5}$$

$$\dot{x}_m \;=\; A_m \cdot x_m \;+\; B_m \cdot u_c \tag{3.6}$$

where:

$$A_p(x_p;t) \;=\; \begin{bmatrix} 0 & 1 \\ a_{p_0}(x_p;t) & a_{p_1}(x_p;t) \end{bmatrix} \quad , \quad F(\sigma;t) \;=\; \begin{bmatrix} 0 \\ f(\sigma;t) \end{bmatrix}$$

$$B_p(x_p;t) \;=\; \begin{bmatrix} 0 \\ b_p(x_p;t) \end{bmatrix} \quad , \quad D \;=\; \begin{bmatrix} 0 \\ v \end{bmatrix}$$

$$A_m \;=\; \begin{bmatrix} 0 & 1 \\ a_{m_0} & a_{m_1} \end{bmatrix} \quad , \quad B_m \;=\; \begin{bmatrix} 0 \\ b_m \end{bmatrix}$$

The control law u is the output of a PID controller which has the general structure given by equations (2.6) and (2.7). The adaptive control loop we will be using has the form shown in Figure (3.2).

In the control loop of Figure (3.2), there are several signal paths (such as $e_i, y_p, y_m, u_c, \ldots$) available to the designer, and an adaptive PID controller may therefore have different combinations of the terms defined by relations given in (2.7).

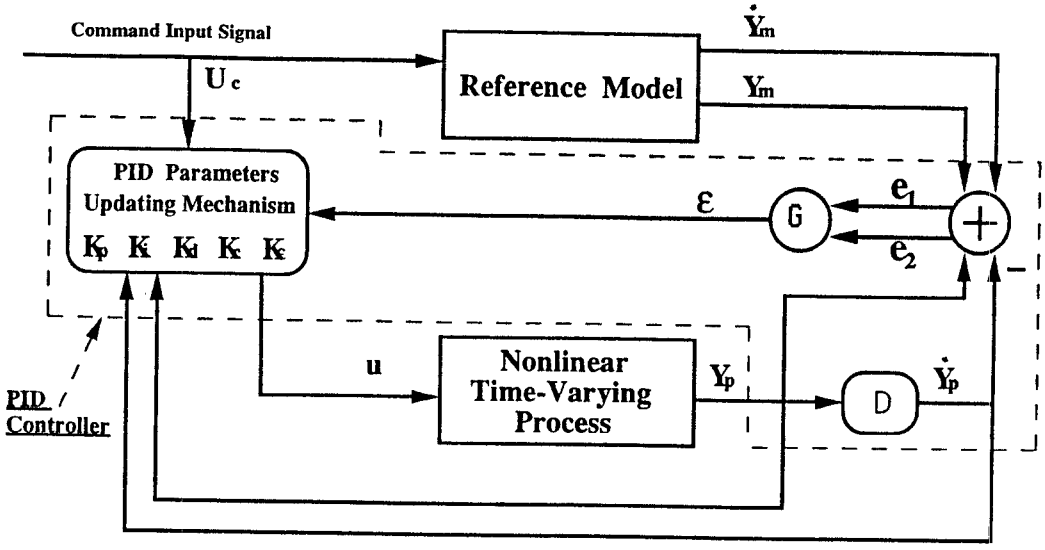

Figure 3.2: Adaptive PID Controller

Hence, an adaptive PID controller has a higher degree of freedom, in the sense described in section 2.3.4, than the conventional error regulators introduced in the last chapter. One possible candidate for the control law of the type given by (2.6) is the following:

$$u(t) = Py_p + Dy_p + Pu_c + P\epsilon + I\epsilon \tag{3.7}$$

where ϵ is a function of e_1 and e_2 called the *"weighted error"*, which we will define in the next section.

The objective of the PID controller in Figure (3.2), then, is to guarantee that

the error $\bar{e}$ tends asymptotically to zero, i.e.

$$\bar{e}(t) = \begin{bmatrix} y_m - y_p \\ \dot{y}_m - \dot{y}_p \end{bmatrix} \longrightarrow \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad \text{as } t \longrightarrow \infty$$

By subtracting equations (3.5) and (3.6), we observe that the error vector $\bar{e}$ satisfies the following vector differential equation:

$$\dot{\bar{e}} = A_m \cdot \bar{e} + (A_m - A_p) \cdot x_p - F(\sigma) - B_p \cdot u - D + B_m \cdot u_c \qquad (3.8)$$

Equation (3.8) is a nonlinear time-varying system with u defined by equation (2.6), and is said to be written in " error space " since it describes the evolution of process errors with respect to the desired trajectories. We would like to analyze the stability properties of this system for various forms of the control law u such as the one given by (3.7). More specifically, we would like to find the *structure* of the control algorithm and the *value* of its corresponding parameters, so that the system in (3.8) becomes asymptotically stable. Hence, the model following problem (see section 3.2) can be restated as to find the respective *updating rules* for PID parameters in (3.7), i.e. $K_{P_{y_p}}$, $K_{d_{y_p}}$, $K_{P_{u_c}}$, K_{P_ϵ}, and K_{i_ϵ}, such that the solution of (3.8) tends asymptotically to zero.

3.5 Adaptive PID Design

Let $G \in \Re^{1\times 2}$ be a gain vector defined as:

$$G \stackrel{\text{def}}{=} B_p^T \cdot P \qquad (3.9)$$

where $P \in \Re^{2\times 2}$ is a symmetric positive definite matrix which is the solution of the Lyapunov equation:

$$P \cdot A_m \; + \; {A_m}^T \cdot P \; = \; -Q \tag{3.10}$$

and $Q \in \Re^{2\times 2}$ is a symmetric positive definite design matrix. As we show in the next section, the rate at which the error $\bar{e}$ approaches to zero depends on Q and is constrained by A_m through equation (3.10). Since A_m is an asymptotic stable matrix, specified in (3.6), the Lyapunov equation (3.10) gives a unique symmetric positive definite solution P for every symmetric positive definite design matrix Q.

Let:

$$\epsilon \stackrel{\text{def}}{=} G \cdot \bar{e} \tag{3.11}$$

where ϵ is a scalar function of e_1 and e_2 called "weighted error," and G is as defined in (3.9).

Define $B_p^{\dagger}$ as the pseudoinverse of B_p:

$$B_p^{\dagger} \; = \; (B_p^T \cdot B_p)^{-1} \cdot B_p^T$$

and observe that:

$$\begin{aligned} B_p^{\dagger} \quad &= \quad \begin{bmatrix} 0 & \frac{1}{b_p} \end{bmatrix} \\ B_p \cdot B_p^{\dagger} \; &= \; \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} \end{aligned} \tag{3.12}$$

Hence:

$$\begin{aligned} B_m &= (B_p \cdot B_p^{\dagger}) \cdot b_m \\ A_m - A_p &= (B_p \cdot B_p^{\dagger}) \cdot (A_m - A_p) \\ D &= (B_p \cdot B_p^{\dagger}) \cdot D \\ F &= (B_p \cdot B_p^{\dagger}) \cdot F \end{aligned} \tag{3.13}$$

These relations are known as Erzberger's conditions for perfect model following [Erz68].

We choose the following Lyapunov function:

$$V(\bar{e}) = \bar{e}^T \cdot P \cdot \bar{e} \tag{3.14}$$

with P the solution of the Lyapunov equation (3.10), and the process control in the form:

$$u_1(t) = K_{Py_p} y_p + K_{dy_p} \dot{y}_p + K_{Pu_c} u_c + K_\epsilon \tag{3.15}$$

where K_ϵ is an *offset* term. For $\dot{V}$ one obtains from (3.8) and (3.14) the following expression:

$$\begin{aligned} \dot{V} &= \bar{e}^T (P \cdot A_m + {A_m}^T \cdot P)\bar{e} + 2\bar{e}^T P (A_m - A_p) x_p - 2\bar{e}^T P \cdot F(\sigma) \\ &\quad - 2\bar{e}^T P \cdot B_p u + 2\bar{e}^T P \cdot B_m u_c - 2\bar{e}^T P \cdot D \end{aligned}$$

from (3.10) and (3.13) this leads to:

$$\begin{aligned} \dot{V} &= -\bar{e}^T Q \bar{e} + 2\bar{e}^T G^T B_p^{\dagger} (A_m - A_p) x_p - 2\bar{e}^T G^T B_p^{\dagger} F(\sigma) \\ &\quad -2\bar{e}^T G^T u + 2\bar{e}^T G^T B_p^{\dagger} B_m u_c - 2\bar{e}^T G^T B_p^{\dagger} D \\ &= -\bar{e}^T Q \bar{e} + 2\bar{e}^T G^T \Big[B_p^{\dagger} (A_m - A_p) x_p \\ &\quad - B_p^{\dagger} F(\sigma) + B_p^{\dagger} B_m u_c - B_p^{\dagger} D - u \Big] \end{aligned} \tag{3.16}$$

Taking into account (3.11) and (3.12):

$$\dot{V}(\bar{e}) = -\bar{e}^T Q \bar{e} + \frac{2\epsilon}{b_p}[a_0 y_p + a_1 \dot{y}_p - f(\sigma) + b_m u_c - v - b_p u] \quad (3.17)$$

where:

$$\begin{aligned} a_0 &= a_{m_0} - a_{p_0} \\ a_1 &= a_{m_1} - a_{p_1} \end{aligned} \quad (3.18)$$

and a_{m_i}, a_{p_i} are defined in (3.1) and (3.3).

Therefore, assuming the process control u in (3.17) is given by (3.15), we have:

$$\begin{aligned} \dot{V}(\bar{e}) &= -\bar{e}^T Q \bar{e} \\ &\quad + \tfrac{2\epsilon}{b_p}\Big[(a_0 - b_p K_{P_{y_p}}) y_p + (a_1 - b_p K_{d_{y_p}}) \dot{y}_p \\ &\quad + (b_m - b_p K_{P_{u_c}}) u_c - (f(\sigma) + v + b_p K_\epsilon)\Big] \end{aligned} \quad (3.19)$$

A suitable choice for the controller parameters is:

$$\begin{aligned} K_{P_{y_p}} &= \frac{a_0(x_p;t)}{b_p(x_p;t)} \\ K_{d_{y_p}} &= \frac{a_1(x_p;t)}{b_p(x_p;t)} \\ K_{P_{u_c}} &= \frac{b_m}{b_p(x_p;t)} \\ K_\epsilon &= \frac{\rho(x_p;t) + v_{max}}{|b_p(x_p;t)|} \cdot sign(\epsilon) \end{aligned} \quad (3.20)$$

where $\rho(\cdot)$ and v_{max} are defined in (3.3) and (3.2), a_0 and a_1 are defined in (3.18), and:

$$sign(\epsilon) = \begin{cases} +1 & \text{if } \epsilon > 0 \\ 0 & \text{if } \epsilon = 0 \\ -1 & \text{if } \epsilon < 0 \end{cases} \quad (3.21)$$

which acts as a relay (refer to section 2.4.2). Choosing the controller parameters according to (3.20) causes the error $e_1 = y_m - y_p$ to approach zero asymptotically. This is apparent by substituting these parameters into (3.19) which gives:

$$\begin{aligned} \dot{V}(\bar{e}) &= -\bar{e}^T Q \bar{e} - \frac{2\epsilon}{b_p}\left[f(\sigma) + v + b_p \cdot \frac{\rho(x_p;t) + v_{max}}{|b_p|} \cdot \epsilon\, sign(\epsilon)\right] \\ &= -\bar{e}^T Q \bar{e} - 2\left[\frac{\epsilon}{b_p} f(\sigma) + \frac{\epsilon}{b_p} v + \frac{\rho(x_p;t) + v_{max}}{|b_p|} \cdot \epsilon\, sign(\epsilon)\right] \end{aligned} \quad (3.22)$$

from inequalities in (3.2) and (3.3) we get:

$$\dot{V}(\bar{e}) \leq -\bar{e}^T Q \bar{e} \leq -\lambda_{min}(Q) \cdot \|\bar{e}\|^2 \quad (3.23)$$

Hence, since $\dot{V}(\bar{e})$ is zero only at $\bar{e} = \bar{0}$ and is negative everywhere else, we conclude that as long as there is error, the error vector $\bar{e}$ tends to zero, i.e:

$$\lim_{t \to \infty} \bar{e} = \bar{0} \qquad \forall \bar{e}(0) \in \Re^2 \quad (3.24)$$

and consequently: $y_p \longrightarrow y_m$.

In order for (3.20) to guarantee asymptotic stability of the error system given by (3.8), we need to assure that the null solution, i.e. $\bar{e} = 0$, is the equilibrium point of (3.8). This is not clear from the conditions given by (3.1)-(3.4) since the structure of disturbances v and the nonlinear term $f(\cdot)$ is not given, and consequently $f(\bar{e})$ is not necessarily zero at $\bar{e} = 0$ for *different operating points* x_m. As a result, despite the fact that the error $\bar{e}$ approaches to the origin (as long as there is nonzero error), and hence $x_p \longrightarrow x_m$, disturbances v and the non-zero nonlinearity $f(x_m)$ reexcite the error system (3.8) when $\bar{e}$ reaches zero (since $\ddot{e}_1 \neq 0$). Therefore, the origin is not the equilibrium state and consequently, the

error vector will remain bounded where the magnitude of the bounds depends on the form of v, v_{max}, and the value of $f(\cdot)$ at x_m.

By introducing an integrator into the control law (3.15), a zero steady state solution can practically be enforced in the error system (see section 2.2). To interpret this, assume $v = 0$, the integrator adds just enough control u to compensate for non-zero nonlinearities at $x_p = x_m$, or to say at $\bar{e} = 0$. The null solution of (3.8) will then be asymptotically stable. Small perturbations (about the equilibrium state 0) due to the bounded disturbances v will not affect the stability of (3.8) since, as mentioned before, the error approaches to zero.

With the addition of the integrator, the control law in (3.15) is modified to:

$$u_2(t) = K_{Py_p} y_p + K_{dy_p} \dot{y}_p + K_{Pu_c} u_c + K_\epsilon + I \qquad (3.25)$$

where I stands for the integral action term. We derive this term by modifying the Lyapunov function used in (3.14) and search for conditions that conserve the stability of the adaptive system. Consider the candidate Lyapunov function:

$$W(\bar{e}) = \bar{e}^T \cdot P \cdot \bar{e} + \frac{1}{2\gamma} \cdot I^2 \qquad (3.26)$$

where P is the symmetric positive definite matrix given by (3.10). Note that from (3.14) we have:

$$W = V + \frac{1}{2\gamma} \cdot I^2$$

Let us now evaluate $\dot{W}$:

$$\dot{W} = \dot{V} + \frac{1}{\gamma} \cdot I \cdot \frac{dI}{dt}$$

Using the control law in (3.25) with coefficients given by (3.20) results in a slight modification to (3.22) (we omit the details):

$$\begin{aligned}\dot{W}(\bar{e}) &= -\bar{e}^T Q \bar{e} - 2\left[\frac{\epsilon}{b_p} f(\sigma) + \frac{\epsilon}{b_p}\upsilon + \frac{\rho(x_p;t) + \upsilon_{max}}{|b_p|} \cdot sign(\epsilon)\right] \\ &\quad -2\epsilon \cdot I + \frac{1}{\gamma} \cdot I \cdot \frac{dI}{dt} \end{aligned} \tag{3.27}$$

Hence:

$$\begin{aligned}\dot{W} &\leq -\bar{e}^T Q \bar{e} - 2\epsilon \cdot I + \frac{1}{\gamma} \cdot I \cdot \frac{dI}{dt} \\ &= -\bar{e}^T Q \bar{e} + \frac{I}{\gamma} \cdot \left[\frac{dI}{dt} - 2\gamma\epsilon\right] \end{aligned} \tag{3.28}$$

Consequently, if we choose:

$$\frac{dI}{dt} = 2\gamma \cdot \epsilon \tag{3.29}$$

then $\dot{W}$ will be a negative definite function:

$$\dot{W}(\bar{e}) \leq -\bar{e}^T Q \bar{e} \leq -\lambda_{min}(Q) \cdot \|\bar{e}\|^2 \tag{3.30}$$

Note that $\dot{W} = 0$ only if $\bar{e} = 0$. (3.28) and (3.29) suggest that a suitable choice for the integral term I in (3.25) is:

$$I = I_\epsilon = K_{i_\epsilon} \int_0^t \epsilon dt \tag{3.31}$$

where $K_{i_\epsilon} = 2\gamma$. As a result, W in (3.26) will be decreasing as long as the error $\bar{e}$ is not zero, and it is constant when $\bar{e} = \bar{0}$. Clearly, the error is forced to converge to zero from any non-zero initial state $\bar{e}(0)$. The asymptotic stability of the error system follows.

It should be noted that if the disturbance v *slowly* varies, such as simple ramp or step function disturbances (offsets), over some time intervals, the integrator will still enforce the zero equilibrium in (3.8) at corresponding time intervals. If disturbances are much faster than the integral action time constant and the process dynamics, an error envelope will exist in which the error oscillates around the zero state equilibrium. The amplitude of this envelope will depend on the magnitude of disturbances v and on how fast the process reacts to its input (delay in the process).

In (3.29), γ is a constant design parameter to be chosen by the user, and its value depends in general on the process gain, time-constant, and the operating point x_m. If γ is chosen very large, the integrator accumulates large values and while the stability will still be conserved since (3.30) does not depend on the numerical value of γ, the transient response will be poor. In this case, as it will be explored in section 3.7.1, *integral anti-windup* needs to be employed. In the next section, after we approximate the time-constant of the adaptive system and estimate the rate of convergence at which the error approaches to zero, suggestions on the numerical range of γ will be given.

3.6 Rate of Convergence

In this section we discuss the transient behavior of the error system. More specifically, we search for an estimate of the decay rate of the error $\bar{e}$ to its zero equilibrium state. Define:

$$\eta \stackrel{\text{def}}{=} min\left[\frac{\bar{e}^T Q \bar{e}}{\bar{e}^T P \bar{e}}\right] \tag{3.32}$$

An equivalent definition of η is:

$$\eta \stackrel{\text{def}}{=} \min_{\bar{e}}\left\{\bar{e}^T Q \bar{e}\ ;\ \bar{e}^T P \bar{e} = 1\right\} \tag{3.33}$$

We observe:

$$\eta \leq \frac{\bar{e}^T Q \bar{e}}{\bar{e}^T P \bar{e}} \leq \frac{-\dot{V}}{V}$$

where $V = \bar{e}^T P \bar{e}$ is as defined in (3.14). Then:

$$\dot{V} \leq -\eta \cdot V$$

and

$$V \leq V(\bar{e}(t_0), t_0) \cdot e^{-\eta(t-t_0)} \tag{3.34}$$

where $\bar{e}(t_0) = \bar{e}_0$ is the initial error at time t_0. Hence:

$$\bar{e}^T P \bar{e} \leq \left(\bar{e}_0{}^T P \bar{e}_0\right) \cdot e^{-\eta(t-t_0)} \tag{3.35}$$

Interpreting $\bar{e}^T P \bar{e}$ as the distance from the origin in error space, (3.35) approximates how fast the equilibrium is approached. Since P is a positive definite matrix, we have:

$$\lambda_{min}(P) \cdot \|\bar{e}\|^2 \leq \bar{e}^T P \bar{e} \leq \lambda_{max}(P) \cdot \|\bar{e}\|^2 \tag{3.36}$$

where $\lambda_{min}(P)$ and $\lambda_{max}(P)$ are minimum and maximum eigenvalues of P respectively. Therefore:

$$\|\bar{e}\|^2 \leq \frac{\bar{e}^T P \bar{e}}{\lambda_{min}(P)} \leq \frac{\bar{e}_0{}^T P \bar{e}_0}{\lambda_{min}(P)} \cdot e^{-\eta(t-t_0)} \tag{3.37}$$

and:

$$\|\bar{e}\| \leq \sqrt{\frac{\lambda_{max}(P)}{\lambda_{min}(P)}} \cdot \|\bar{e}(t_0)\| \cdot e^{\frac{-\eta}{2}(t-t_0)} \tag{3.38}$$

Equation (3.38) implies that $\|\bar{e}\|$ decays *at least* as fast as $e^{\frac{-\eta}{2}t}$. This suggests that the transient rate of convergence for y_p to approach y_m (or: $\bar{e} \rightarrow \bar{0}$) is at least:

$$r = \eta/2 \tag{3.39}$$

Hence, $1/r$ is an estimate of the *time-constant* for the adaptive system.

To compute η in (3.33) one can use the Lagrange multiplier technique as described in [Ort70]. Let μ be the Lagrange multiplier. From (3.33), we need to minimize:

$$\min_{\bar{e}} \left\{ \bar{e}^T Q \bar{e} - \mu \cdot \bar{e}^T P \bar{e} \right\}$$

Taking partial derivative with respective to $\bar{e}$ and equating it to zero gives:

$$(Q - \mu P) \cdot \bar{e}_{min} = \bar{0} \tag{3.40}$$

which implies that μ is an *eigenvalue* of the matrix (QP^{-1}), and:

$$\bar{e}_{min}^T Q \bar{e}_{min} = \mu \cdot \bar{e}_{min}^T P \bar{e}_{min} = \mu$$

since $\bar{e}_{min}^T P \bar{e}_{min} = 1$ from (3.33). The term $\bar{e}_{min}^T Q \bar{e}_{min}$ is minimum when μ is minimum and hence:

$$\eta = \lambda_{min}(QP^{-1}) \tag{3.41}$$

i.e. η is the *minimum eigenvalue* of the matrix (QP^{-1}). Note that η is positive since $\bar{e}^T Q \bar{e}$ in (3.33) is always positive (Q is *p.d.*). Hence:

$$r = \frac{\lambda_{min}(QP^{-1})}{2} \tag{3.42}$$

Now that we have an estimate for the rate at which error converges to zero, equation (3.38) and (3.42) can be used in order to estimate a numerical range for γ in (3.29) which result in a smooth transient response for the adaptive system. From (3.11) and (3.9) we have:

$$\|\epsilon\| = \|G \cdot \bar{e}\| \leq |b_p| \cdot \|P \cdot \bar{e}\| \leq |b_p| \lambda_{max}(P) \cdot \|\bar{e}\|$$

and (3.38) gives:

$$\begin{aligned} \|\epsilon\| &\leq |b_p| \lambda_{max}(P) \cdot \sqrt{\frac{\lambda_{max}(P)}{\lambda_{min}(P)}} \cdot \|\bar{e}(t_0)\| \cdot e^{-r(t-t_0)} \\ &\stackrel{\text{def}}{=} M \cdot \|\bar{e}(t_0)\| \cdot e^{-r(t-t_0)} \end{aligned} \tag{3.43}$$

and:

$$\|I\| = \| \int_{t_0}^{t} 2\gamma\epsilon \, dt \| \leq 2\gamma \int \|\epsilon\| \, dt$$

Now assuming b_p is constant:

$$\begin{aligned} \|I\| &\leq 2\gamma M \cdot \|\bar{e}(t_0)\| \cdot \int_{t_0}^{t} e^{-r(t-t_0)} \, dt \\ &= \frac{2\gamma M}{r} \cdot \|\bar{e}(t_0)\| \cdot \left[1 - e^{-r(t-t_0)}\right] \end{aligned} \tag{3.44}$$

If b_p is not constant, then its average over the corresponding time period should be approximated within M in (3.44). Inequality (3.44) suggests that the integral term always remains *bounded* at most by $\frac{2\gamma M}{r} \cdot \|\bar{e}(t_0)\|$. Also, that I approaches its value at steady state, denoted by I^o, at most with the rate r. From (3.26) and (3.30):

$$W(\bar{e}) \;=\; \bar{e}^T P \bar{e} \;+\; \frac{1}{2\gamma} \cdot I^2 \;\leq\; W(\bar{e}(t_0)) \qquad \forall t \geq t_0 \tag{3.45}$$

Hence:

$$\frac{1}{2\gamma} \cdot I^2 \;\leq\; \bar{e}^T(t_0)\; P\; \bar{e}(t_0) \;\leq\; \lambda_{max}(P) \cdot \|\bar{e}(t_0)\|^2$$

and:

$$|I| \;\leq\; \sqrt{2\gamma \lambda_{max}(P)} \cdot \|\bar{e}(t_0)\| \tag{3.46}$$

This together with (3.44) implies:

$$|I| \;\leq\; min\; \left\{ \sqrt{2\gamma \lambda_{max}(P)}\;,\; \frac{2\gamma M}{r} \right\} \cdot \|\bar{e}(t_0)\| \;\stackrel{\text{def}}{=}\; I_{max} \tag{3.47}$$

As mentioned earlier, the primary objective of the integrator is to enforce a zero steady state in the error system. This implies that the bound on $|I|$ in (3.47) should be big enough so that the integral term can reach I^o (the value of I at the steady state $\bar{e} = \bar{0}$ or $x_p = x_m$). From (3.8), I^o may be computed as:

$$I^o \;=\; \frac{-f(x_m)}{b_p(x_m)} \tag{3.48}$$

and hence one candidate for γ is such that:

$$I_{max} \;\simeq\; I^o \tag{3.49}$$

However, since $f(\cdot)$ is unknown we use (3.48) instead to find a bound I^o_{max} on I^o such that $|I^o| \leq I^o_{max}$ and then choose γ based on I^o_{max}. Another alternative is to perform a test by simply observing the process response, using some nominal value for γ, and then measuring the value of I at the given operating condition x_m. The value for γ found by using I^o_{max} instead of I^o can also be used as an initial setting for γ in this test. In most cases where the bound function $\rho(\cdot)$ in (3.2) is chosen reasonably tight, the following value for γ obtained based on I^o_{max} works reasonably well. From (3.48):

$$I^o_{max} = \frac{\rho(x_m)}{|b_p(x_m)|} \tag{3.50}$$

and in order for I^o_{max} to be reachable by I in (3.47) and to prevent integral windup a suitable choice for γ is such that: $I_{max} \simeq I^o_{max}$ or:

$$\gamma \simeq min\left\{ \frac{\rho^2(x_m)}{2\lambda_{max}(P)|b_p(x_m)|^2\|\bar{e}(t_0)\|} \, , \, \frac{r \cdot \rho(x_m)}{2M|b_p(x_m)| \cdot \|\bar{e}(t_0)\|} \right\} \tag{3.51}$$

Since Q is a design matrix, to be chosen by the designer, one can adjust η in (3.41) by varying Q. By assumption, A_m in (3.6) is an asymptotically stable matrix. Hence, for any positive definite matrix Q, the Lyapunov equation (3.10) gives a *unique* positive definite matrix P, and for a fixed Q, η is constrained by this Lyapunov equation. For a given reference system A_m, in order to achieve a better rate of convergence r, we need to choose Q such that $\lambda_{min}(QP^{-1})$ is as large as it could get.

To find a suitable Q, let us choose the diagonal form:

$$Q_q = \begin{bmatrix} q & 0 \\ 0 & 1 \end{bmatrix} \tag{3.52}$$

where q is a parameter to be determined. Note that Q_q in (3.52) is normalized since by multiplying Q_q with a scalar number eigenvalues of (QP^{-1}) do not change. From (3.10) and (3.52) we have:

$$\begin{bmatrix} p_1 & p_2 \\ p_2 & p_3 \end{bmatrix} \cdot \begin{bmatrix} 0 & 1 \\ a_{m_0} & a_{m_1} \end{bmatrix} + \begin{bmatrix} 0 & a_{m_0} \\ 1 & a_{m_1} \end{bmatrix} \cdot \begin{bmatrix} p_1 & p_2 \\ p_2 & p_3 \end{bmatrix} = - \begin{bmatrix} q & 0 \\ 0 & 1 \end{bmatrix}$$

and can be solved for p_i:

$$\begin{cases} p_1 = \frac{a_{m_0}^2 q - a_{m_0} + a_{m_1}^2}{2a_{m_0}a_{m_1}} \\ p_2 = \frac{-1}{2a_{m_0}} \\ p_3 = \frac{1 - q\,a_{m_0}}{2a_{m_0}a_{m_1}} \end{cases} \tag{3.53}$$

Hence, for any A_m, P as a function of q is given by (3.53). Eigenvalues of (QP^{-1}) can then be computed as a function of q, where the smaller of the two is denoted by $\lambda_{min}(q)$. To maximize $\lambda_{min}(q)$, one can solve the corresponding optimization problem:

$$\max_q \ \lambda_{min}(q)$$

and find a suitable value for q that gives the largest $\lambda_{min}(QP^{-1})$.

Another way to find the best q for a given A_m is to simply plot $\lambda_{min}(q)$ as a function of q, and take the numerical value of q corresponding to maximum $\lambda_{min}(QP^{-1})$. An example is treated in the next chapter.

3.7 Modifications of the Algorithm

In this section, the algorithm presented in section 3.5 is partially modified. In section 3.7.1, The effect of anti-windup (discussed in sec 2.3.2) on the stability of the adaptive controller will be explored. The relay-type offset term K_ϵ in (3.20) will be replaced by a *continuous* simple proportional term followed by a hard limiter in section 3.7.2.

It should be noted that all modifications of P, I, and D terms discussed in section 2.3 should also be carried out in implementing the adaptive PID controller.

3.7.1 Integral Action with Anti-Windup:

As discussed in section 2.3.2, to prevent the build-up of error when actuator is saturated, and hence, get a better performance result when the controller comes out of the saturated condition, integral anti-windup should be employed. For this purpose, it is common to limit the integral term with some fixed maximum and minimum bounds. The magnitudes of these bounds depend on the dynamics and limitations of the control actuators and the process.

Limiting integral action may cause stability problems since the update rule in (3.29) and the relationship in (3.30) will not hold on the boundary limits of the integral term (i.e. when $\frac{dI}{dt} = 0$ is forced) unless $\epsilon = 0$. Hence, we need to

compensate for this in other terms of the control law.

Let us define the saturation function as:

$$sat\,(x,c) \;=\; \begin{cases} c & if\, x > c \\ x & if\, |x| \le c \\ -c & if\, x < -c \end{cases} \qquad (3.54)$$

$$c \;>\; 0$$

then the integral with anti-windup is simply:

$$\begin{aligned} I &= sat\,(I_\epsilon, c) \\ I_\epsilon &= K_{i_\epsilon} \textstyle\int_0^t \epsilon\, d\tau \end{aligned} \qquad (3.55)$$

where $c > 0$ is the limit of the integral term, and is a design parameter. We modify (3.26) to produce the candidate Lyapunov function:

$$W(\bar{e}) \;=\; \bar{e}^T \cdot P \cdot \bar{e} \;+\; \frac{1}{2\gamma} \cdot (I - c)^2 \qquad (3.56)$$

with K_ϵ in (3.20) modified by:

$$K_\epsilon \;=\; \left[\frac{\rho(x_p;t) + v_{max}}{|b_p(x_p;t)|} \;+\; c \right] \cdot sign(\epsilon) \qquad (3.57)$$

To evaluate $\dot{W}$ when the control law u is given by (3.25), we have:

$$\begin{aligned} \dot{W}(\bar{e}) \;=\; & -\bar{e}^T Q \bar{e} \\ & + \tfrac{2\epsilon}{b_p} \Big[(a_0 - b_p K_{P_{y_p}}) y_p \;+\; (a_1 - b_p K_{d_{y_p}}) \dot{y}_p \\ & \quad + (b_m - b_p K_{P_{u_c}}) u_c - \; (f(\sigma) + v + b_p K_\epsilon) \Big] \\ & \qquad -2\epsilon I \;+\; \tfrac{1}{\gamma}(I - c) \cdot \tfrac{dI}{dt} \end{aligned} \qquad (3.58)$$

and employing (3.55), (3.57), and (3.20) gives:

1. Before integral saturation ($|I| < c$):

$$\begin{aligned}
\dot{W}(\bar{e}) &\leq -\bar{e}^T Q \bar{e} - 2c\epsilon sign(\epsilon) \\
&\quad + \tfrac{I}{\gamma} \cdot \underbrace{\left(\frac{dI}{dt} - 2\gamma\epsilon\right)}_{0} - c \cdot \overbrace{\left(1/\gamma \frac{dI}{dt}\right)}^{2\epsilon} \\
&= -\bar{e}^T Q \bar{e} - 2c\epsilon(sign(\epsilon) + 1) \\
&\leq -\bar{e}^T Q \bar{e}
\end{aligned}$$

Hence:

$$\begin{cases} \dot{W} < 0 & if\ \bar{e} \neq \bar{0} \\ \dot{W} = 0 & if\ \bar{e} = \bar{0} \end{cases} \tag{3.59}$$

2. During integral saturation ($|I| = c$):

$$\begin{aligned}
\dot{W}(\bar{e}) &\leq -\bar{e}^T Q \bar{e} - 2c\epsilon sign(\epsilon) \\
&\quad + \tfrac{I}{\gamma} \cdot \left(\underbrace{\frac{dI}{dt}}_{0} - 2\gamma\epsilon\right) - c \cdot \overbrace{\left(1/\gamma \frac{dI}{dt}\right)}^{0} \\
&= -\bar{e}^T Q \bar{e} - 2c\epsilon(sign(\epsilon) \pm 1) \\
&\leq -\bar{e}^T Q \bar{e}
\end{aligned}$$

Hence:

$$\begin{cases} \dot{W} < 0 & if\ \bar{e} \neq \bar{0} \\ \dot{W} = 0 & if\ \bar{e} = \bar{0} \end{cases} \tag{3.60}$$

(3.59) and (3.60) show that, with modification of K_ϵ in (3.57), W is decreasing before and during integral saturation as long as there is some error $\bar{e}$, and it is constant when $\bar{e} = \bar{0}$. Asymptotic stability of the error system, therefore, will not be effected by integral saturation.

3.7.2 Continuous Version:

The K_ϵ term in (3.20) and (3.57) acts as a *relay* with a gain $\xi(x_p;t)$ where:

$$\xi(x_p;t) \stackrel{\text{def}}{=} \frac{\rho(x_p;t)+v_{max}}{|b_p(x_p;t)|} + c \tag{3.61}$$

This *discontinuous* term sometimes introduces undesirable chattering in the control signal which often makes the controller difficult to realize in practice. A *continuous* version of this term can therefore be implemented using the following approximation:

$$sign(\epsilon) \approx \begin{cases} +1 & if\ \epsilon > \xi_0 \\ \epsilon/\xi_0 & if\ |\epsilon| \le \xi_0 \\ -1 & if\ \epsilon < -\xi_0 \end{cases} \tag{3.62}$$

where ξ_0 is a design parameter, typically small and its value depends on the controller actuators' characteristics. Notice that by an appropriate choice of ξ_0 the resulting continuous control law, achieved by implementing (3.62) in the K_ϵ term of (3.20) or (3.57), is arbitrarily close to the original discontinuous term, and hence, it gives a performance close to that of the discontinuous version. The modified control law is then the following PID-type control:

$$\begin{aligned} u(t) &= P_{y_p} + D_{y_p} + P_{u_c} + P_\epsilon + I \\ &= K_{P_{y_p}} y_p + K_{d_{y_p}} \dot{y}_p + K_{P_{u_c}} u_c + K_{P_\epsilon} sat\,(\epsilon, \xi_0) + sat\,(I_\epsilon, c) \end{aligned} \tag{3.63}$$

where:

$$\begin{cases} K_{Py_p} &= \frac{a_{m_0}-a_{p_0}}{b_p} \\ K_{dy_p} &= \frac{a_{m_1}-a_{p_1}}{b_p} \\ K_{Pu_c} &= \frac{b_m}{b_p} \\ K_{P\epsilon} &= \frac{\xi(\cdot)}{\xi_0}. \\ K_{i\epsilon} &= 2\,\gamma \end{cases} \tag{3.64}$$

where *sat* (ϵ, ξ_0) is a *limiter function* of height ξ_0 defined in (3.54), $\xi(\cdot)$ is defined in (3.61), ξ_0 is a design parameter, and a_{m_i}, a_{p_i}, b_m, and b_p are given in (3.1) and (3.4).

An alternative to the fixed approximation in (3.62) is to vary ξ_0 as the error $\bar{e}$ changes. The following choice of $\xi_0(\bar{e})$ guarantees the asymptotic stability of the error system (3.8) achieved by the PID controller given by (3.63) for any scalar $\delta > 0$, where δ is a design parameter and μ_{min} is the minimum eigenvalue of Q:

$$\xi_0 = \frac{\mu_{min}(Q)}{2\delta\xi(\cdot)} \|\bar{e}\|^2 \tag{3.65}$$

which results in:

$$\begin{aligned} \dot{V} &\leq -\tilde{\lambda} \cdot \|\bar{e}\|^2 \\ \tilde{\lambda} &= \tfrac{\delta-1}{\delta} \end{aligned} \tag{3.66}$$

Hence, the rate of convergence for $\|\bar{e}\| \rightarrow \bar{0}$ will be $\tilde{\lambda}/2$.

Another alternative in approximating the $sign(\cdot)$ function with a continuous

function is the following [ACG84]:

$$sign(\epsilon) \approx \frac{\epsilon}{|\epsilon| + \tau} \tag{3.67}$$

where τ is a small positive constant. This approximation acts like a "soft limiter" of height one.

3.8 Generalizations to Multiloop Systems:

In this section we apply our design scheme described in sections 3.5 and 3.7 which originally was developed for SISO systems to an *interacting multiloop* system where each subsystem is assumed to be of the form given by descriptions (3.1)-(3.3).

We consider a dynamical system composed of n subprocesses governed by:

$$\dot{X}_p = A_p(X_p;t) \cdot X_p + B_p(X_p;t) \cdot U + F(\sigma;t) + D \tag{3.68}$$

Where:

$$X_p = \begin{bmatrix} y_{p_1} \\ y_{p_2} \\ \vdots \\ y_{p_n} \\ \hline \dot{y}_{p_1} \\ \dot{y}_{p_2} \\ \vdots \\ \dot{y}_{p_n} \end{bmatrix} \stackrel{\text{def}}{=} \begin{bmatrix} x_{p_1} \\ x_{p_2} \\ \vdots \\ \\ \\ x_{p_{2n}} \end{bmatrix} = \begin{bmatrix} Y_p \\ \hline \dot{Y}_p \end{bmatrix} \quad , \quad U = \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_n \end{bmatrix}$$

$$A_p = \left[\begin{array}{ccc|ccc} & 0 & & & I_n & \\ \hline a_{1_{P_1}} & \cdots & a_{n_{P_1}} & a_{n+1_{P_1}} & \cdots & a_{2n_{P_1}} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ a_{1_{P_n}} & \cdots & a_{n_{P_n}} & a_{n+1_{P_n}} & \cdots & a_{2n_{P_n}} \end{array}\right] \stackrel{\text{def}}{=} \left[\begin{array}{c|c} 0_{n\times n} & I_{n\times n} \\ \hline A_{p_0} & A_{p_1} \end{array}\right]$$

$$B_p = \left[\begin{array}{cccc} 0 & 0 & \cdots & 0 \\ \vdots & & & \vdots \\ 0 & 0 & \cdots & 0 \\ \hline b_{P_1} & 0 & \cdots & 0 \\ 0 & b_{P_2} & & 0 \\ \vdots & & \ddots & \vdots \\ 0 & 0 & \cdots & b_{P_n} \end{array}\right] \stackrel{\text{def}}{=} \left[\begin{array}{c} 0_{n\times n} \\ \hline B_{p_1} \end{array}\right]$$

$$F(\sigma;t) = \left[\begin{array}{c} 0 \\ \vdots \\ 0 \\ \hline f_1 \\ \vdots \\ f_n \end{array}\right] \stackrel{\text{def}}{=} \left[\begin{array}{c} 0_{n\times 1} \\ \hline F_1 \end{array}\right] \quad , \quad D = \left[\begin{array}{c} 0 \\ \vdots \\ 0 \\ \hline v_1 \\ \vdots \\ v_n \end{array}\right] \stackrel{\text{def}}{=} \left[\begin{array}{c} 0_{n\times 1} \\ \hline D_1 \end{array}\right]$$

and y_{p_i} is the output of the ith subprocess. Note that all matrix entries $a_{j_{P_i}}$ and b_{P_i} can be nonlinear functions of X_p. δ is a function of y_{p_i}, $\dot{y}_{p_i}$ and higher derivatives of outputs. While the functional forms of a_{i_P} and b_{P_i} are assumed known, the nonlinear functions f_i and disturbances v_i are uncertain and subject to the following relationships for some known measurable functions $\rho_i(X_p;t)$ and

bounds $v_{i_{max}}$:

$$\begin{aligned} |f_i(\sigma;t)| &\leq \rho_i(X_p;t) \\ |v_i| &\leq v_{i_{max}} \end{aligned} \tag{3.69}$$

The first step of the design is to choose a reference model, a linear second order time-invariant differential equation, for each subprocess:

$$\ddot{y}_{m_i} = a_{1_{m_i}}\dot{y}_{m_i} + a_{0_{m_i}}y_{m_i} + b_{m_i}u_{c_i} \quad i = 1, 2, \cdots, n \tag{3.70}$$

As noted in section 3.3, the desired performance characteristics expected from the controlled system is given by (3.70) which may also be rewritten in the following form:

$$\dot{X}_m = A_m \cdot X_m + B_m \cdot U_c \tag{3.71}$$

where $X_m \in \Re^{2n}, U_c \in \Re^n, A_m \in \Re^{2n\times 2n}, B_m \in \Re^{2n\times 2n}$, and:

$$A_m = \left[\begin{array}{c|c} 0_{n\times n} & I_n \\ \hline A_{m0} & A_{m1} \end{array}\right] \quad , \quad B_m = \left[\begin{array}{c} 0 \\ \hline B_{m1} \end{array}\right]$$

$$X_m = \left[\begin{array}{c} y_{m_1} \\ \vdots \\ y_{m_n} \\ \hline \dot{y}_{m_1} \\ \vdots \\ \dot{y}_{m_n} \end{array}\right] = \left[\begin{array}{c} Y_m \\ \hline \dot{Y}_m \end{array}\right] \quad , \quad U_c = \left[\begin{array}{c} u_{c_1} \\ \vdots \\ u_{c_n} \end{array}\right]$$

with $A_{m0} = diag\ (a_{0_{m_i}}), A_{m1} = diag\ (a_{1_{m_i}}), B_{m1} = diag\ (b_{m_i})$.

Define the error vector as:

$$\bar{e} \;=\; X_m \;-\; X_p \tag{3.72}$$

and the vector differential equation governing the behavior of $\bar{e}$ is:

$$\dot{\bar{e}} \;=\; A_m \cdot \bar{e} \;+\; (A_m - A_p) \cdot X_p \;-\; F(\sigma;t) \;-\; B_p \cdot U \;-\; D \;+\; B_m \cdot U_c \tag{3.73}$$

Hence, the design objective is to find the controls u_i in U such that the error $\bar{e}$ tends asymptotically to zero, i.e.

$$\bar{e}(t) \;=\; \begin{bmatrix} Y_m - Y_p \\ \dot{Y}_m - \dot{Y}_p \end{bmatrix} \longrightarrow \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad \text{as } t \longrightarrow \infty \tag{3.74}$$

Define the gain matrix $G \in \Re^{n \times 2n}$:

$$G \;\stackrel{\text{def}}{=}\; B_p^T \cdot P \tag{3.75}$$

where $P \in \Re^{2n \times 2n}$ is a symmetric positive definite matrix and is the solution of the Lyapunov equation:

$$P \cdot A_m \;+\; {A_m}^T \cdot P \;=\; -Q \tag{3.76}$$

and $Q \in \Re^{2n \times 2n}$ is a symmetric positive definite design matrix which, as discussed in section 3.5 and 3.6, specifies the rate of convergence in the error system.

Observe that the matching conditions (Erzberger's conditions [Erz68]) for

perfect model following are satisfied:

$$\begin{aligned} B_m &= (B_p \cdot B_p^{\dagger}) \cdot b_m \\ A_m - A_p &= (B_p \cdot B_p^{\dagger}) \cdot (A_m - A_p) \\ D &= (B_p \cdot B_p^{\dagger}) \cdot D \\ F &= (B_p \cdot B_p^{\dagger}) \cdot F \end{aligned} \tag{3.77}$$

with[1]:

$$B_p^{\dagger} = \left[0 \left| diag\left(\frac{1}{b_{P_i}}\right) \right. \right] \tag{3.78}$$

Let the weighted error $\bar{\epsilon} \in \Re^{n \times 1}$ be:

$$\epsilon \stackrel{\text{def}}{=} G \cdot \bar{e} = \begin{bmatrix} \epsilon_1 \\ \vdots \\ \epsilon_n \end{bmatrix} \tag{3.79}$$

with G obtained by (3.75), and choose the Lyapunov function candidate:

$$V(\bar{e}) = \bar{e}^T \cdot P \cdot \bar{e} \tag{3.80}$$

where P is the solution of (3.76).

For $\dot{V}$, we obtain the same expression as in (3.16):

$$\begin{aligned} \dot{V} = -\bar{e}^T Q \bar{e} + 2 \overbrace{\bar{e}^T G^T}^{\bar{\epsilon}^T} \Big[& B_p^{\dagger}(A_m - A_p)x_p \\ & - B_p^{\dagger} F(\sigma) + B_p^{\dagger} B_m u_c - B_p^{\dagger} D - U \Big] \end{aligned} \tag{3.81}$$

[1] $B_p^{\dagger} = (B_p^T \cdot B_p)^{-1} \cdot B_p^T$ is the pseudoinverse of B_p as defined in section 3.5.

From (3.69), (3.78), and (3.79) we have:

$$\begin{aligned}\dot{V}(\bar{e}) = & -\bar{e}^T Q \bar{e} \\ & + 2 \sum_{i=1}^{n} \frac{\bar{\epsilon}_i}{b_{P_i}} \Big[a_i^0 y_{p_i} + a_i^1 \dot{y}_p + \sum_{\substack{j=1 \\ j \neq i}}^{n} \Big[a_{j_{P_i}} y_{p_j} + a_{n+j_{P_i}} \dot{y}_{p_j} \Big] \\ & - f_i(\sigma) + b_{m_i} u_{c_i} - v_i - b_{P_i} u_i \Big] \end{aligned} \tag{3.82}$$

where:

$$\begin{aligned} a_i^0 &= a_{0_{m_i}} - a_{i_{P_i}} \\ a_i^0 &= a_{1_{m_i}} - a_{n+i_{P_i}} \end{aligned} \tag{3.83}$$

Note that each ith term of the summation in (3.82) has a structure identical to that of equation (3.17) in section 3.5 except the interacting terms with coefficients $a_{k_{P_i}}$ for $k = 1, \cdots, 2n$ which is the result of multiloop interactions defined within (3.68). This similarity between MIMO case and SISO case simply generalizes our treatment developed in previous sections and suggests the same form of PID parameter adaptation laws given in (3.63) and (3.64) with slight modification:

$$\begin{aligned} u_i(t) &= P y_{p_i} + D \dot{y}_{p_i} + P u_{c_i} + P \epsilon_i + I_i + \sum_{\substack{j=1 \\ j \neq i}}^{n} \Big[P^i y_{p_j} + D^i \dot{y}_{p_j} \Big] \\ &= K^i_{P y_p} y_{p_i} + K^i_{d y_p} \dot{y}_{p_i} + K^i_{P u_c} u_{c_i} + K^i_{P \epsilon} sat\,(\epsilon_i, \xi_0{}^i) + sat\,(I_\epsilon{}^i, c_i) \\ &\quad + \sum_{\substack{j=1 \\ j \neq i}}^{n} \Big[K^i_{P_j} y_{p_j} + K^i_{d_j} \dot{y}_{p_j} \Big] \end{aligned} \tag{3.84}$$

where:

$$\begin{cases} K^i_{Py_p} = \frac{a_{0_{m_i}} - a_{0_{P_i}}}{b_{P_i}} \\ K^i_{dy_p} = \frac{a_{1_{m_i}} - a_{1_{P_i}}}{b_{P_i}} \\ K^i_{Pu_c} = \frac{b_{m_i}}{b_p} \\ K^i_{P\epsilon} = \frac{\xi_i(\cdot)}{\xi_0} \\ K^i_{i\epsilon} = 2\,\gamma_i \\ K^i_{P_j} = \frac{-a_{j_{P_i}}}{b_{P_j}} \\ K^i_{d_j} = \frac{-a_{n+j_{P_i}}}{b_{P_j}} \end{cases} \tag{3.85}$$

with:

$$\xi_i(X_p;t) \stackrel{\text{def}}{=} \frac{\rho_i(X_p;t) + v_{i_{max}}}{|b_{P_i}(X_p;t)|} + c_i \tag{3.86}$$

as defined in section 3.7. Notice that in simple multiloop processes, $a_{k_{P_i}} \equiv 0$ for most $k \neq i$ which substantially simplifies the algorithm above.

Chapter 4

Simulations

In this chapter we apply the design scheme developed in the last chapter to several nonlinear processes which illustrate the performance of the proposed adaptive PID controller. A simulation routine for this purpose was developed using EASY5W (Engineering Analysis System) software package. A schematic view of the simulated adaptive system generated by EASY5W is shown in Figure (4.1).

4.1 Example 1:

Consider a process that satisfies the following differential equation:

$$\ddot{y}_p = a_{p_1} \cdot \dot{y}_p + a_{p_0} \cdot y_p + \overbrace{\alpha_1 \cdot \sin(\beta_1 y_p) + \alpha_2 \cdot \cos(\beta_2 y_p)}^{f(\cdot)} + b_p \cdot u + v \quad (4.1)$$

where y_p is the process output, u is the control input, and v is a disturbance in the form of a random process.

Assuming $a_{p_1} = 1, a_{p_0} = 2, b_p = 5, \alpha_1 \leq 1$, and $\alpha_2 \leq 1.5$, we have:

$$|f(y_p)| \leq \rho = 2.5 \tag{4.2}$$

Suppose the parameters $\alpha_1, \alpha_2, \beta_1$ and β_2 are unknown to the controller. For simulation, we chose: $\beta_1 = 100$, $\beta_2 = 10, \alpha_1 = 1$, and $\alpha_2 = 1.5$. The bound v_{max} on the disturbance v is:

$$|v| \leq v_{max} = 1.5 \tag{4.3}$$

The *desired* linear model considered is:

$$\ddot{y}_p = -19\dot{y}_m - 90y_m + 90u_c \tag{4.4}$$

with two poles at $s = -9$ and $s = -10$. Figures (4.2) and (4.3) show the transient response of the simulated adaptive system with initial error $e_1(0) = 16$, and the PID control law as in (3.63):

$$\begin{aligned} u(t) &= P_{y_p} + D_{y_p} + P_{u_c} + P_\epsilon + I \\ &= K_{P_{y_p}} y_p + K_{d_{y_p}} \dot{y}_p + K_{P_{u_c}} u_c + K_{P_\epsilon} sat\,(\epsilon, \xi_0) + sat\,(I_\epsilon, c) \end{aligned} \tag{4.5}$$

where the PID parameters are determined from (3.64):

$$\begin{cases} K_{P_{y_p}} = -18.4 \\ K_{d_{y_p}} = -4 \\ K_{P_{u_c}} = 18 \\ K_{P_\epsilon} = 0.9 \\ K_{i_\epsilon} = 2 \end{cases} \tag{4.6}$$

and for ξ_0 in (3.62) we chose: $\xi_0 = 0.01$. Figures (4.4) and (4.5) show the same results for a longer simulation run time.

We note that $f(\cdot)$ in (4.1) is treated as an uncertain nonlinearity, and the controller does not assume knowledge of this term. The bound in (4.2) is therefore, instead of $f(\cdot)$, employed to tune the controller using the tuning rules given by (3.64). The command input signal u_c and the control u for this simulation are shown in Figure (4.6). In Figure (4.7), integral term I_ϵ, proportional error term K_ϵ and disturbances v are shown.

The design matrix Q_q was chosen as ($q = 0.011$):

$$Q_q = \begin{bmatrix} 1 & 0 \\ 0 & 0.011 \end{bmatrix} \tag{4.7}$$

where q was optimized through the procedure described in section 3.6. Figure (4.8) shows the relationship between q and $\lambda_{min}(QP^{-1})$, which was obtained using Mathematica:

Hence, P_q was found by solving the Lyapunov equation (3.10):

$$P_q = \begin{bmatrix} 0.1579 & 0.005556 \\ 0.005556 & 0.0005819 \end{bmatrix} \tag{4.8}$$

Consequently, ϵ given by (3.11) and (3.9), was obtained using $P = 1000\,P_q$ with:

$$\begin{cases} \lambda_{min}(QP^{-1}) = 5.56 \\ \lambda_{max}(QP^{-1}) = 32.44 \end{cases}$$

which determined the rate of convergence for the adaptive system as:

$$r \;=\; \frac{\lambda_{min}(QP^{-1})}{2} \;=\; 2.78$$

which means the error decays at least as fast as $e^{-2.78\,t}$.

Figure (4.9) shows the error trajectories in the adaptive loop, namely: ϵ, e_1, and e_2, and as shown, they approach zero asymptotically.

4.2 Example 2:

Our second illustration was made using a nonlinear plant governed by the following differential equation:

$$\ddot{y}_p \;-\; \dot{y}_p \;-\; 2y_p^2 \;+\; \overbrace{\alpha_1 \cdot \sin(\beta_1 t) \;+\; \alpha_2 \cdot \cos(\beta_2 y_p)}^{f(\cdot)} \;=\; 5 \cdot u \;+\; v \qquad (4.9)$$

Choosing: $\alpha_1 = 1, \alpha_2 = 1.5, \beta_1 = 3$, and $\beta_2 = 6$ gave:

$$|f(y_p)| \;\leq\; \rho \;=\; 2.5 \qquad (4.10)$$

Throughout this simulation we used: $\xi_0 = 0.01$, $v_{max} = 1$, and $e_1(t = 0) = 8$. Figures (4.10) through (4.29) show the results for four different cases. We used the same *desired* model as before (equation (4.4)), and therefore the matrices P and Q are the same as the ones in the last example (4.7) and (4.8).

Controller parameters were again computed from (3.64):

$$\begin{cases} K_{P_{y_p}} = & -18 - 2y_p/5 \\ K_{d_{y_p}} = & -4 \\ K_{P_{u_c}} = & 18 \\ K_{P_\epsilon} = \frac{\rho + v_{max}}{b_p} = & \frac{3.5}{5} \\ K_{i_\epsilon} = & 2 \end{cases} \tag{4.11}$$

In case 1, Figures (4.10) through (4.13), the command input signal:

$$u_c = 10\sin(2t) + 5\cos(3t) \tag{4.12}$$

was applied. In case 2, Figures (4.17) through (4.14), the same input was applied, but a 5% measurement noise was added to the plant output y_p, and its derivative was estimated with a second order filter. The filter used the Tustin's approximation method given by equation (2.17) (see section 2.3.3 for details) with $\tau = 50$.

In cases 3 and 4, Figures (4.18) through (4.23) and Figures (4.24) through (4.29) respectively, we repeated the previous cases except for the input signal u_c which was replaced by a *square wave* signal.

Figures (4.14) and (4.24) show a slight degradation of the transient response of the closed-loop system due to the output measurement noise in the adaptive system. Figures (4.15) and (4.25) compare the *desired model* output derivative, *plant* output derivative, and the *approximated* derivative. These figures confirm our earlier statements concerning the effect of the derivative term (D) in

improving the transient response and the importance of the output derivative approximation employed in the PID controller.

4.3 Example 3:

In the third example, a *saturation function*, such as the one given by equation (3.54), was added to the plant treated in the last example. The amplitude of the saturation function used was $c = 5$ which resulted in:

$$|f(y_p)| \leq \rho = 2.5 + 5 = 7.5 \tag{4.13}$$

Hence, the PID parameters were the same as in (4.11) except for the K_ϵ term which needed to be changed to :

$$K_\epsilon = \frac{7.5}{5} = 1.5 \tag{4.14}$$

The results for two cases are plotted in Figures (4.30) through (4.37). In the first case, Figures (4.30) through (4.33), the input was the same as the one in (4.12), and no measurement noise was applied. In the second case, however, a 5% measurement noise was added and the output derivative was estimated as we did in case 2 and 4 in the previous example. Notice the deterioration in the performance when the measurement noise was introduced in the system. As in the earlier two examples, all the error trajectories in the adaptive system approached zero.

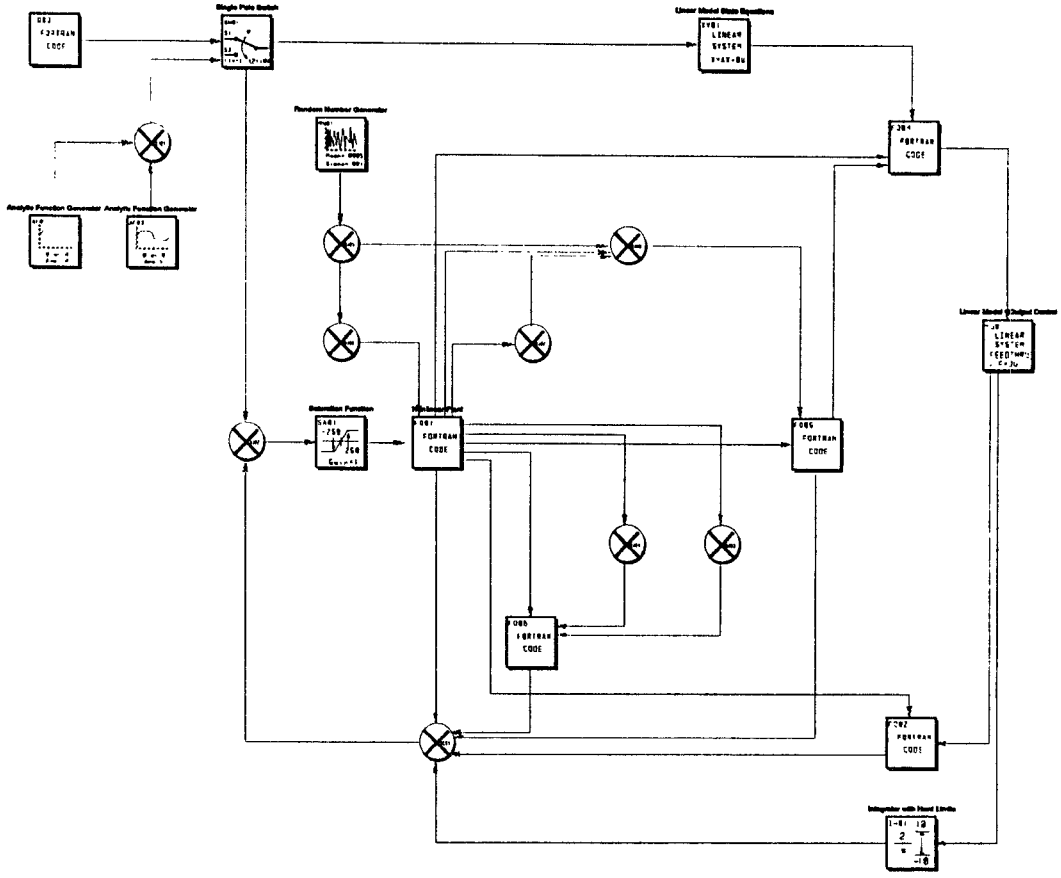

Figure 4.1: Schematic View of the Adaptive PID Control Simulator

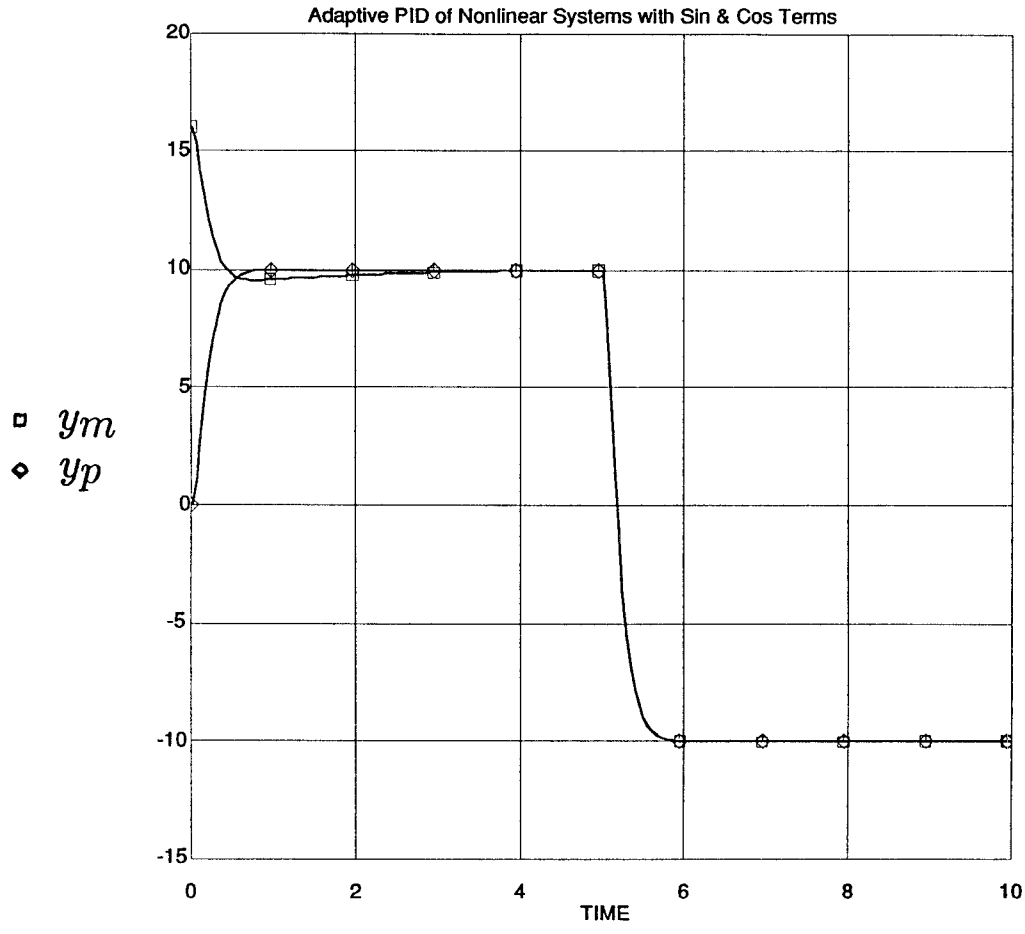

Figure 4.2: Desired Model and Plant Output Trajectories: y_m and y_p in Example 1

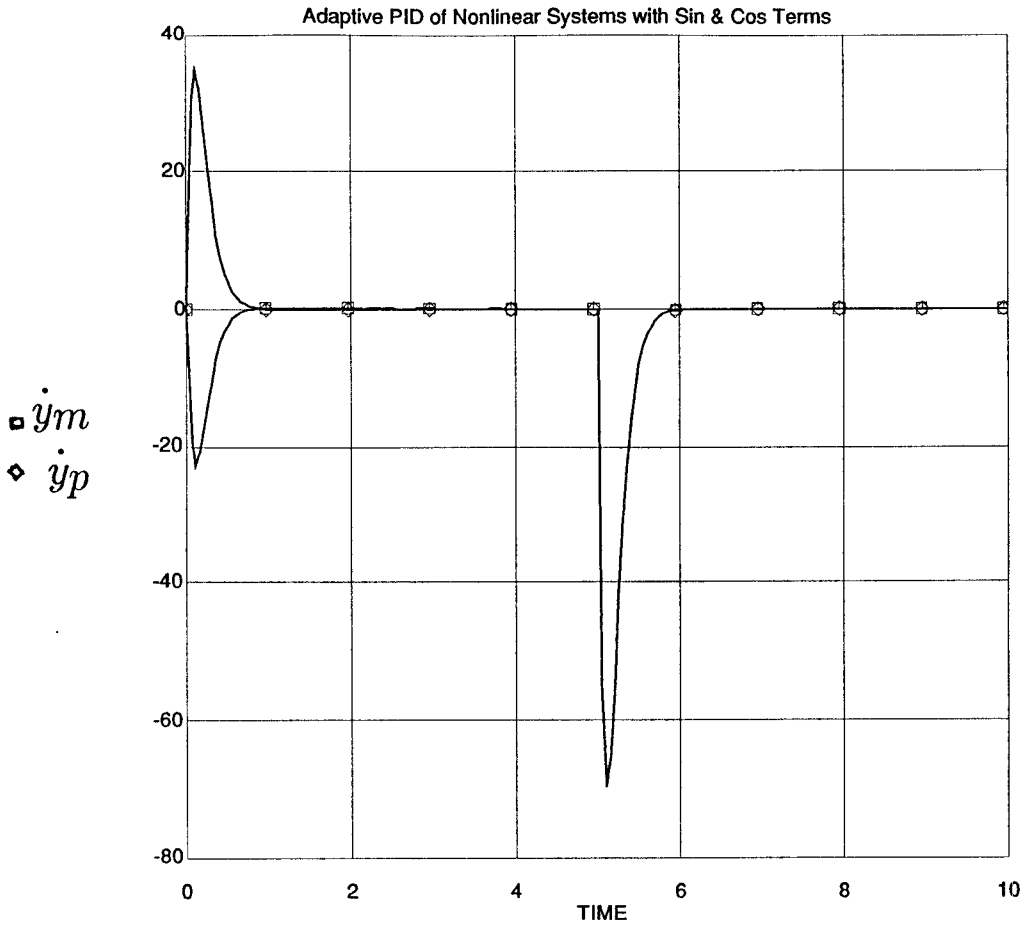

Figure 4.3: Desired Model and Plant Output Derivative Trajectories: $\dot{y}_m$ and $\dot{y}_p$ in Example 1

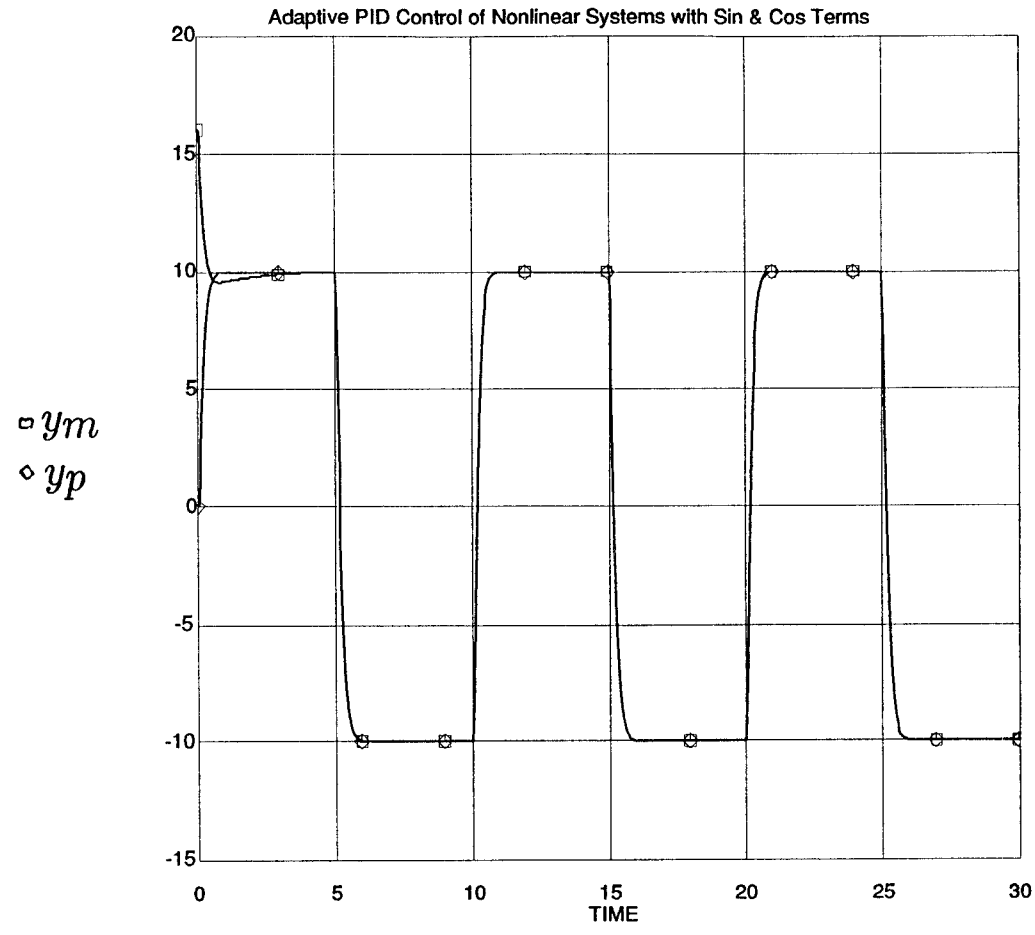

Figure 4.4: Desired Model and Plant Output Trajectories: y_m and y_p in Example 1

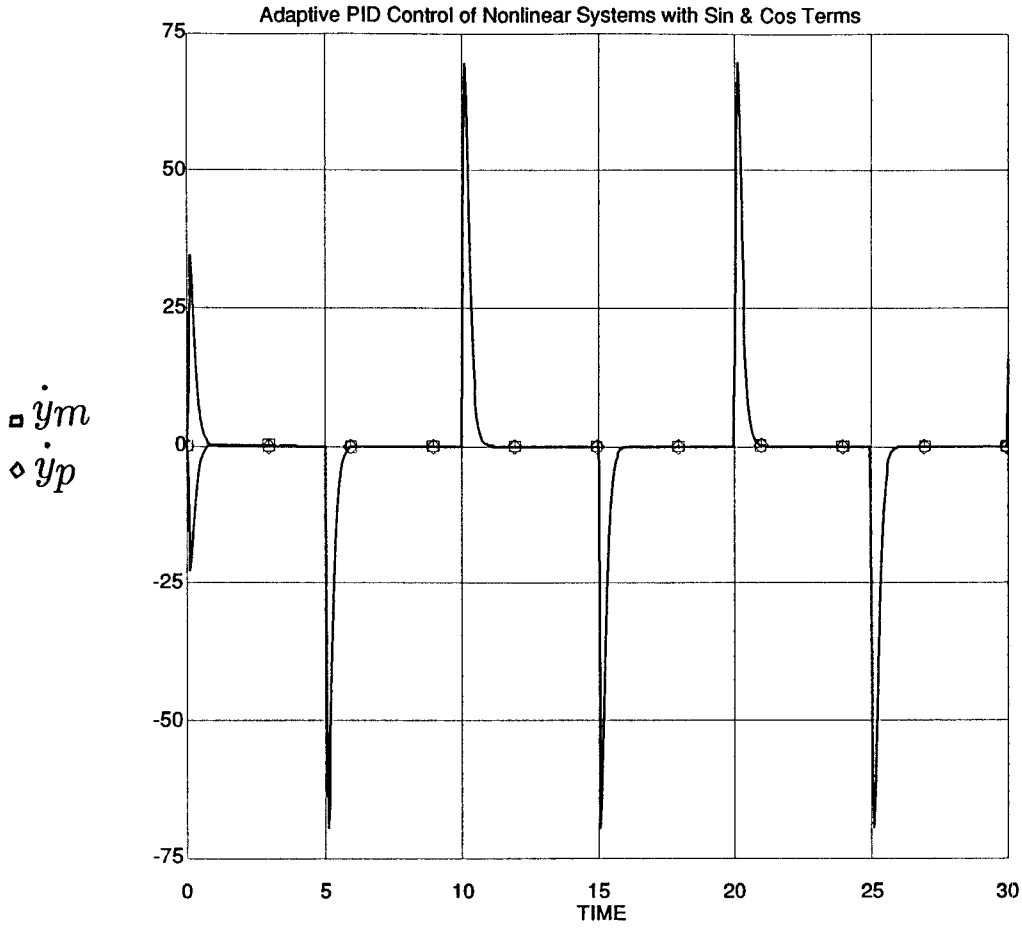

Figure 4.5: Desired Model and Plant Output Derivative Trajectories: $\dot{y}_m$ and $\dot{y}_p$ in Example 1

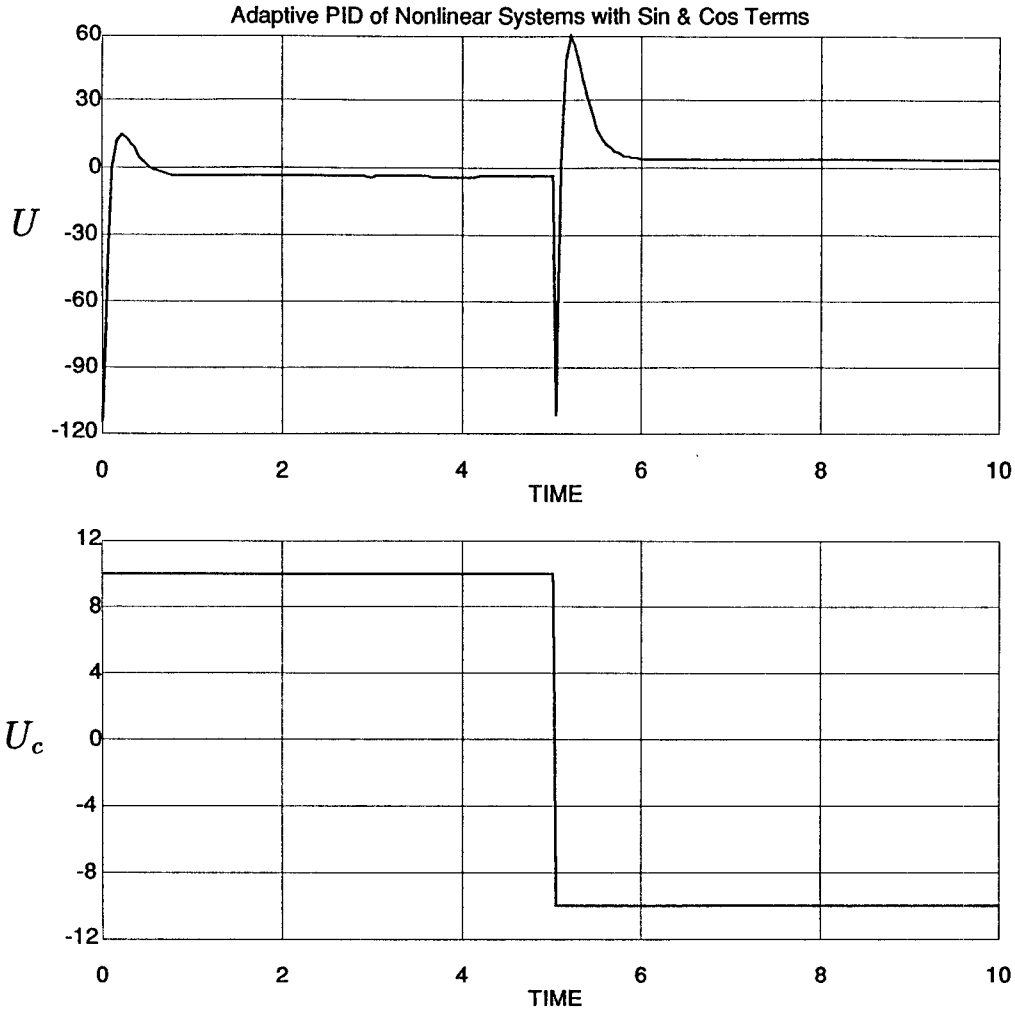

Figure 4.6: Control u and Command Input Signal u_c in Example 1

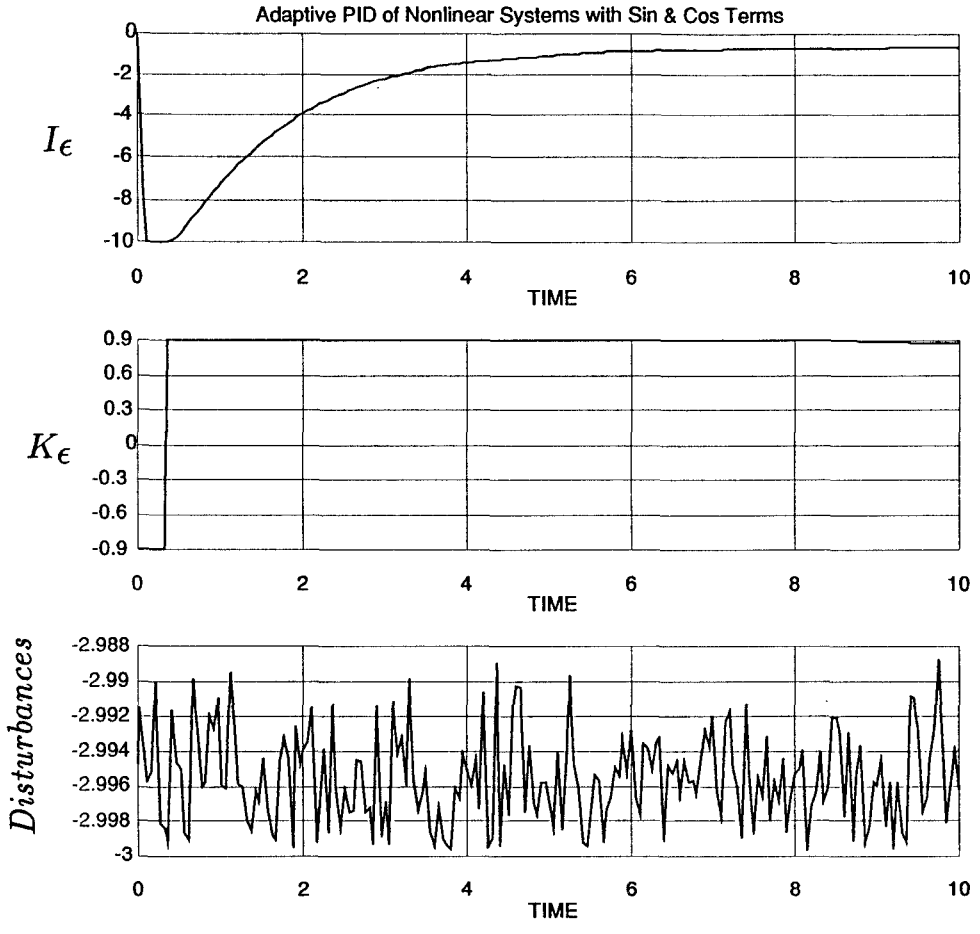

Figure 4.7: Integral term I_ϵ, K_ϵ, and Disturbances v in Example 1

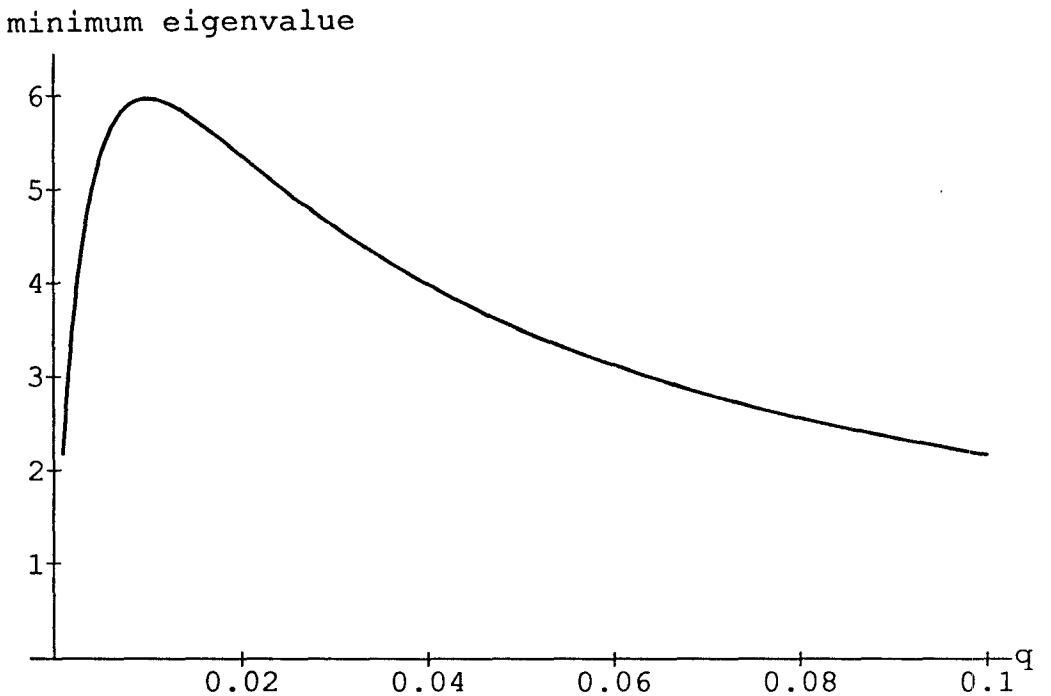

Figure 4.8: $\lambda_{min}(QP^{-1})$ vs. q

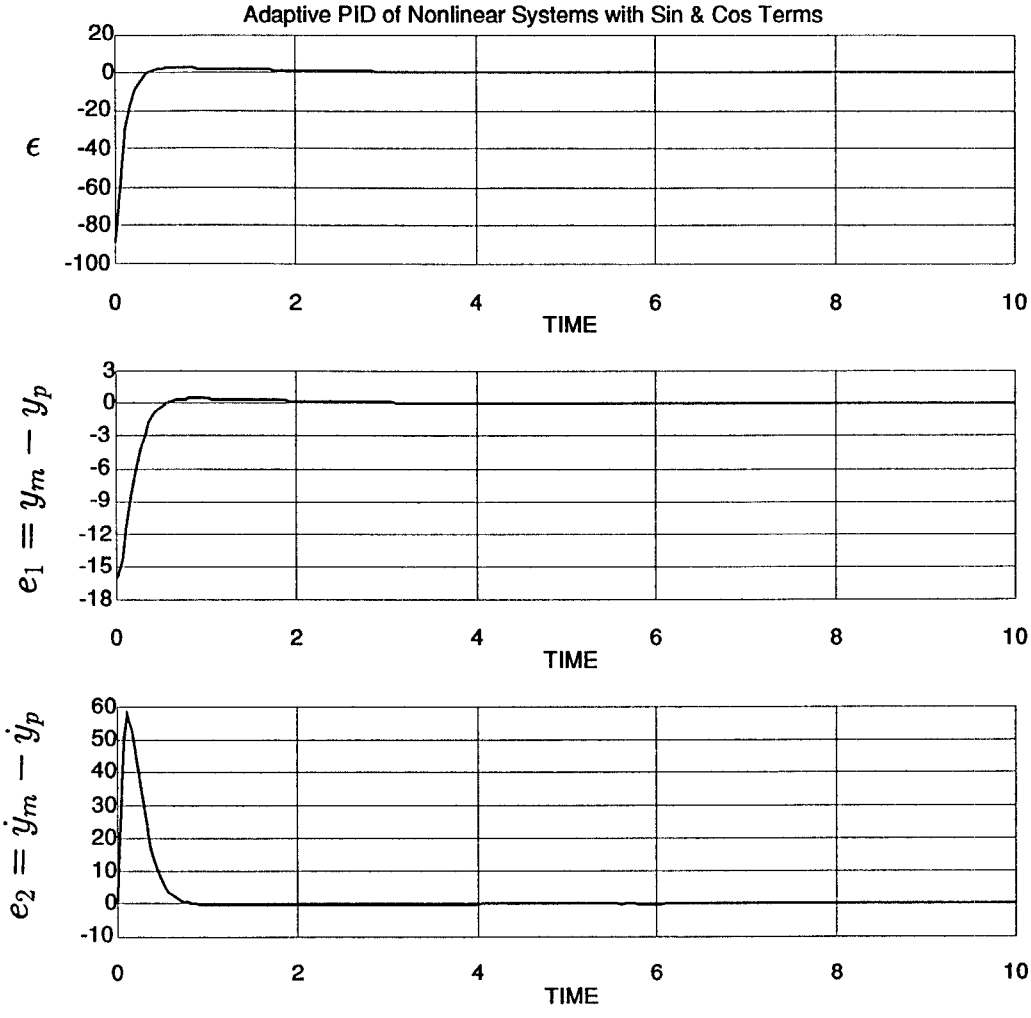

Figure 4.9: Error Trajectories in Example 1

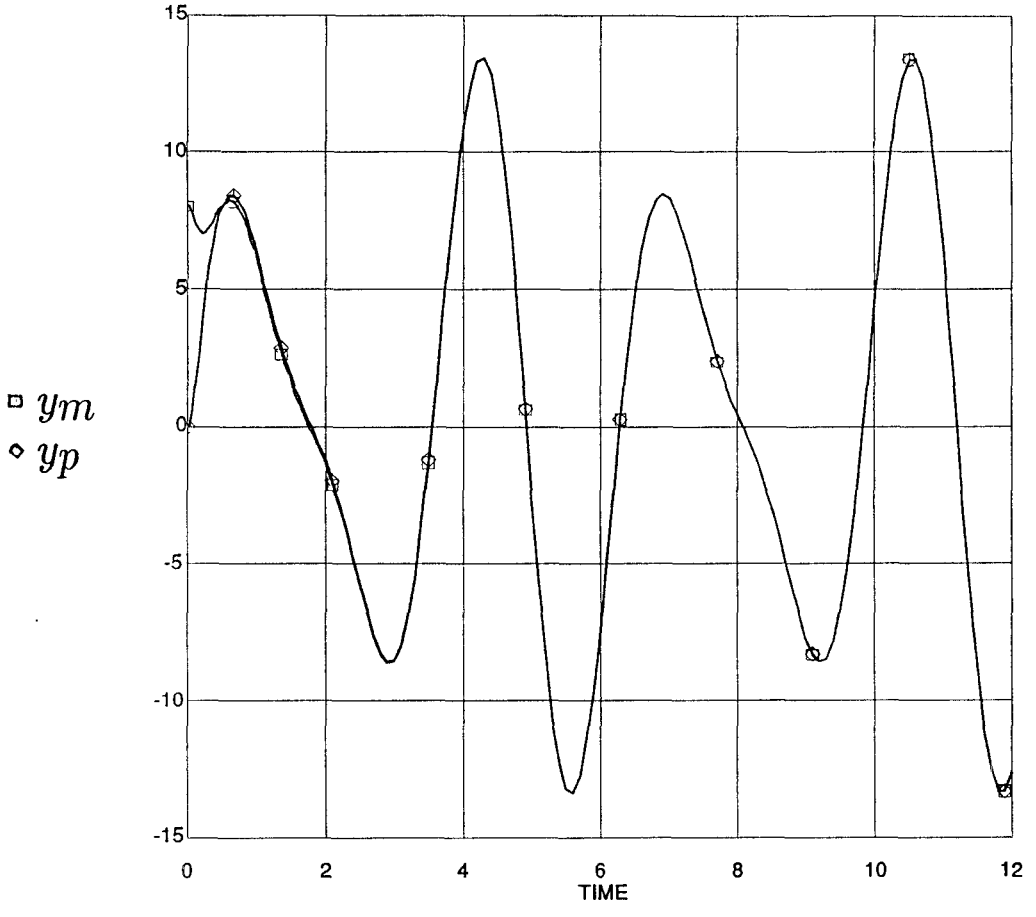

Figure 4.10: Desired Model and Plant Output Trajectories: y_m and y_p in Example 2, Case 1

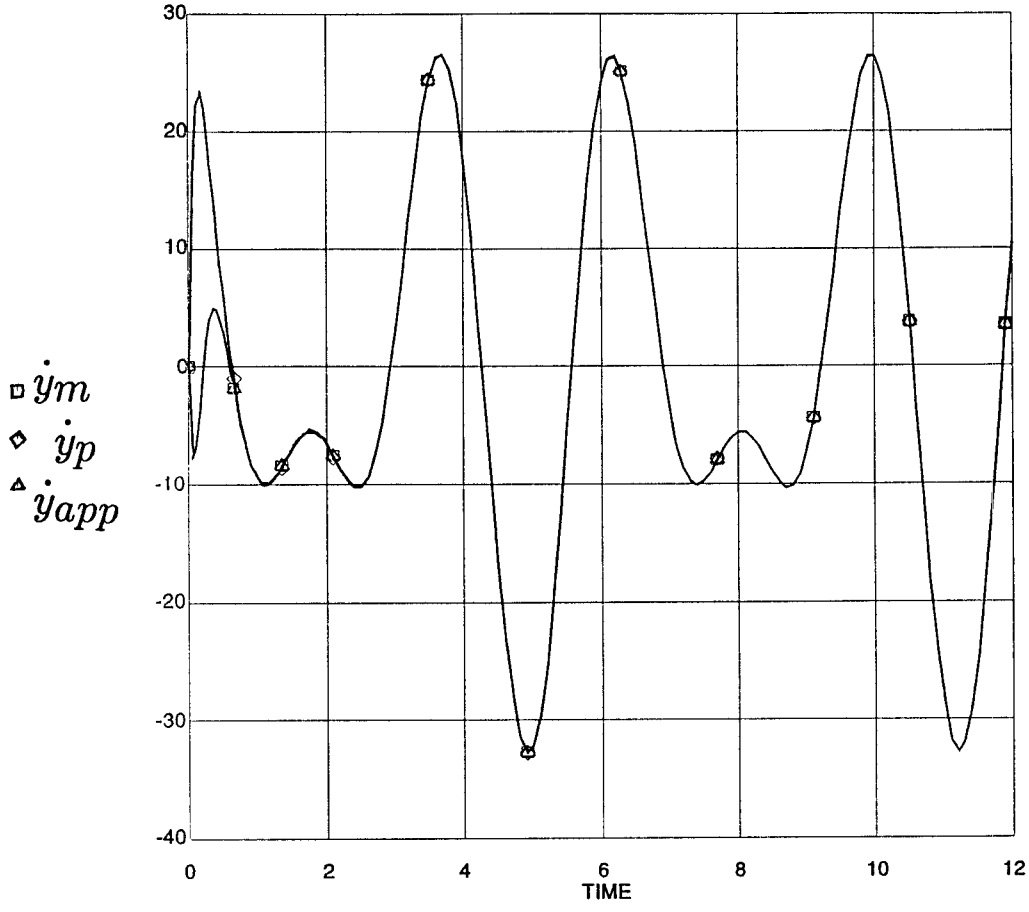

Figure 4.11: Desired Model and Plant Output Derivative Trajectories: $\dot{y}_m$ and $\dot{y}_p$ in Example 2, Case 1

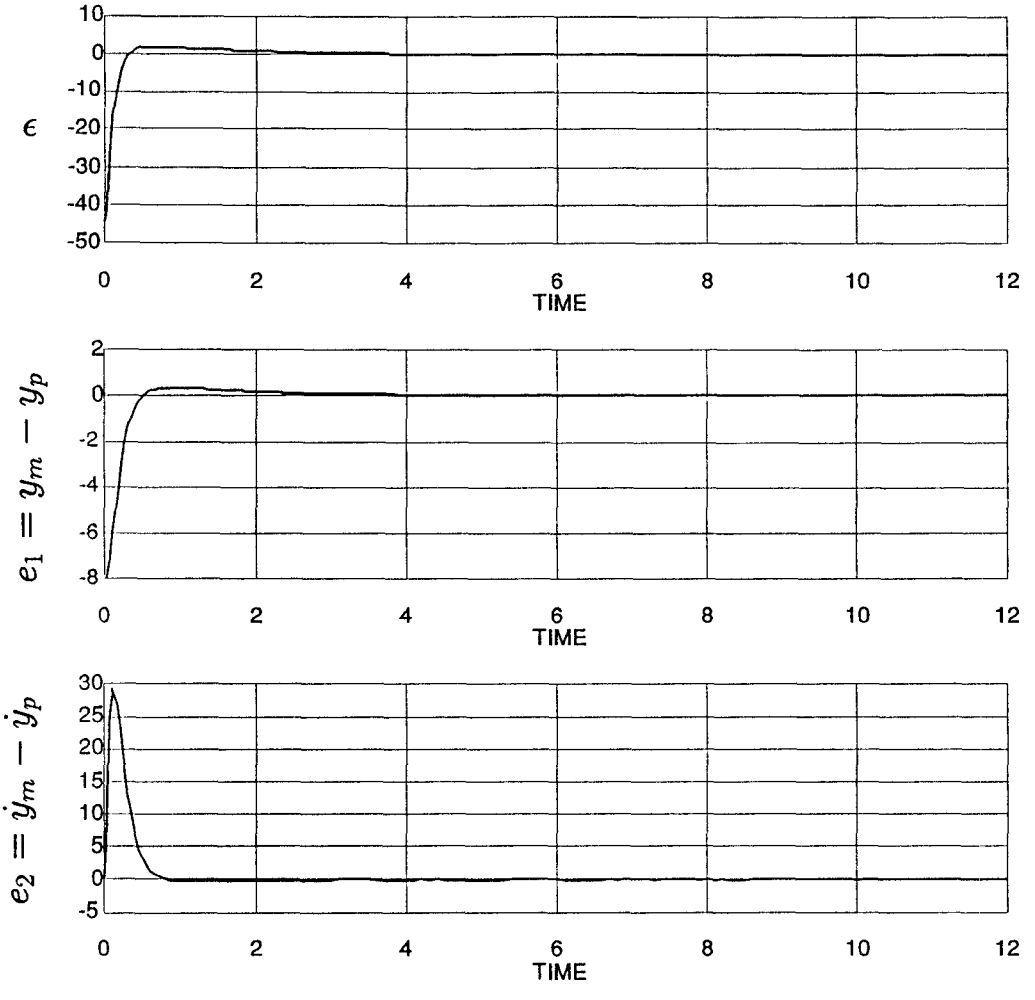

Figure 4.12: Error Trajectories in Example 2, Case 1

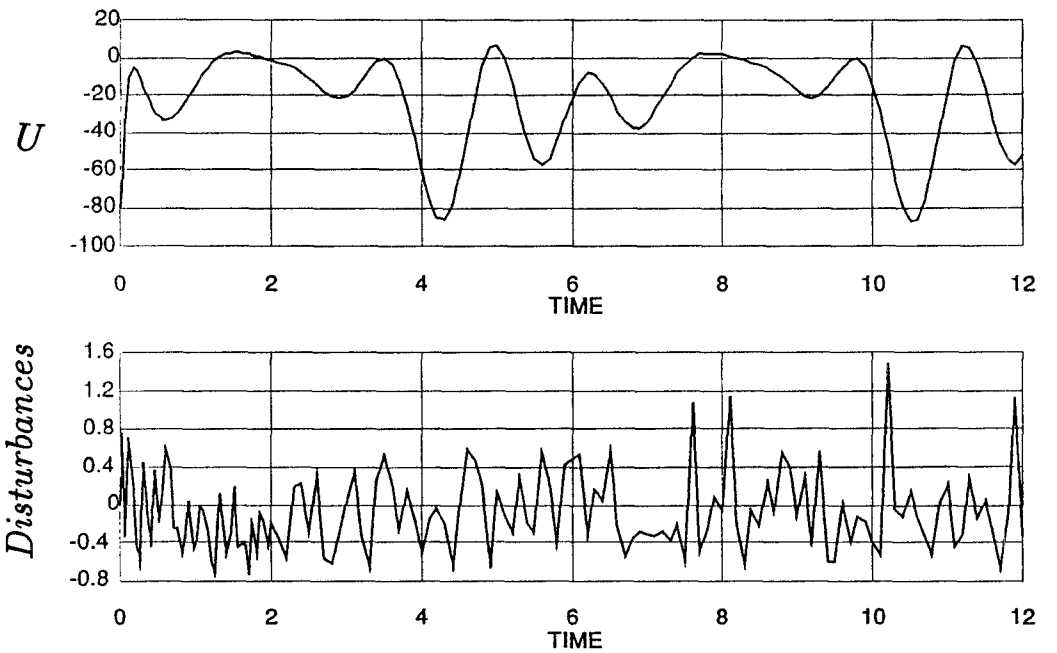

Figure 4.13: Control u and Disturbances v in Example 2, Case 1

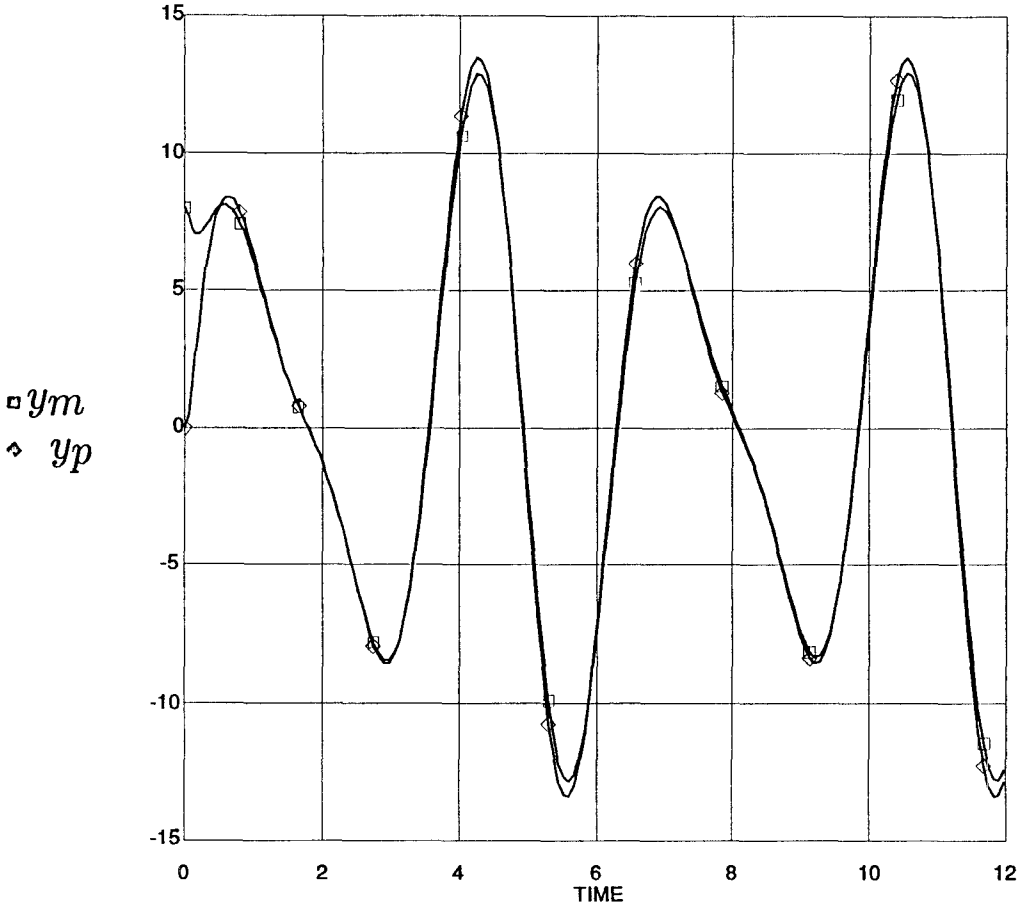

Figure 4.14: Desired Model and Plant Output Trajectories: y_m and y_p in Example 2, Case 2

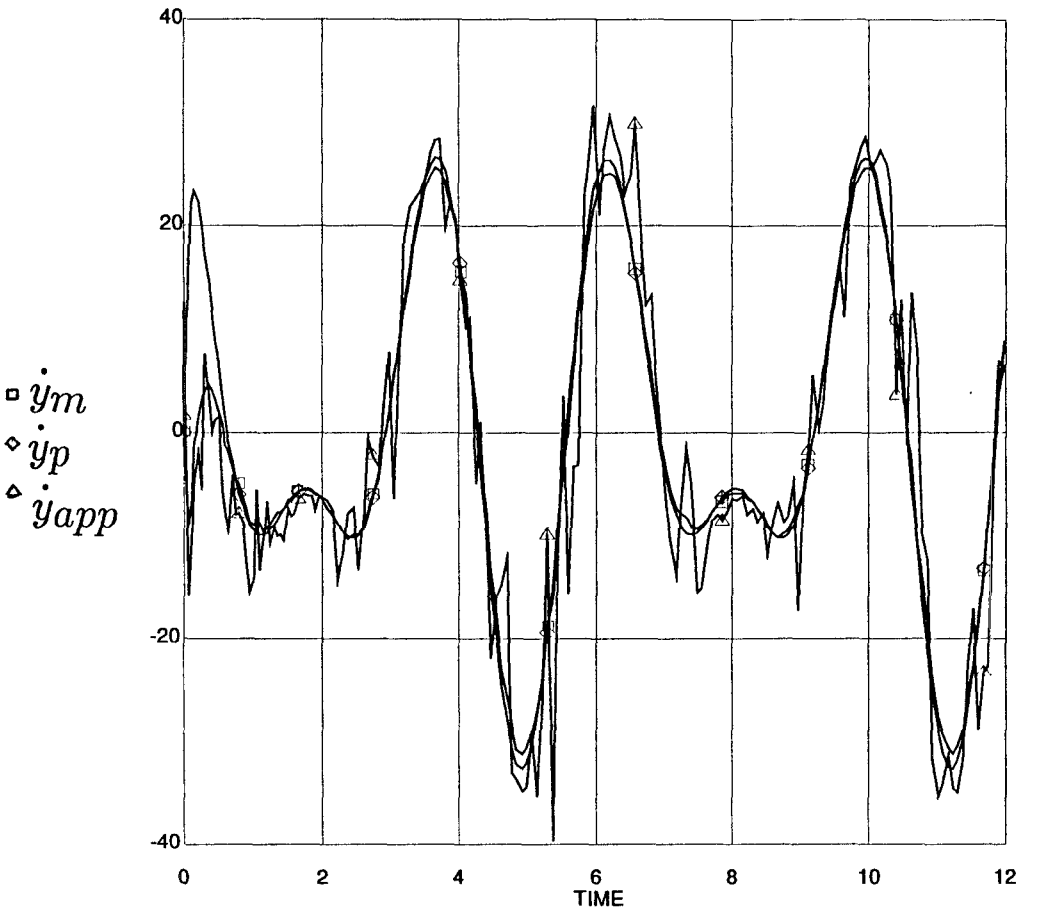

Figure 4.15: Desired Model and Plant Output Derivative Trajectories: $\dot{y}_m$ and $\dot{y}_p$ in Example 2, Case 2

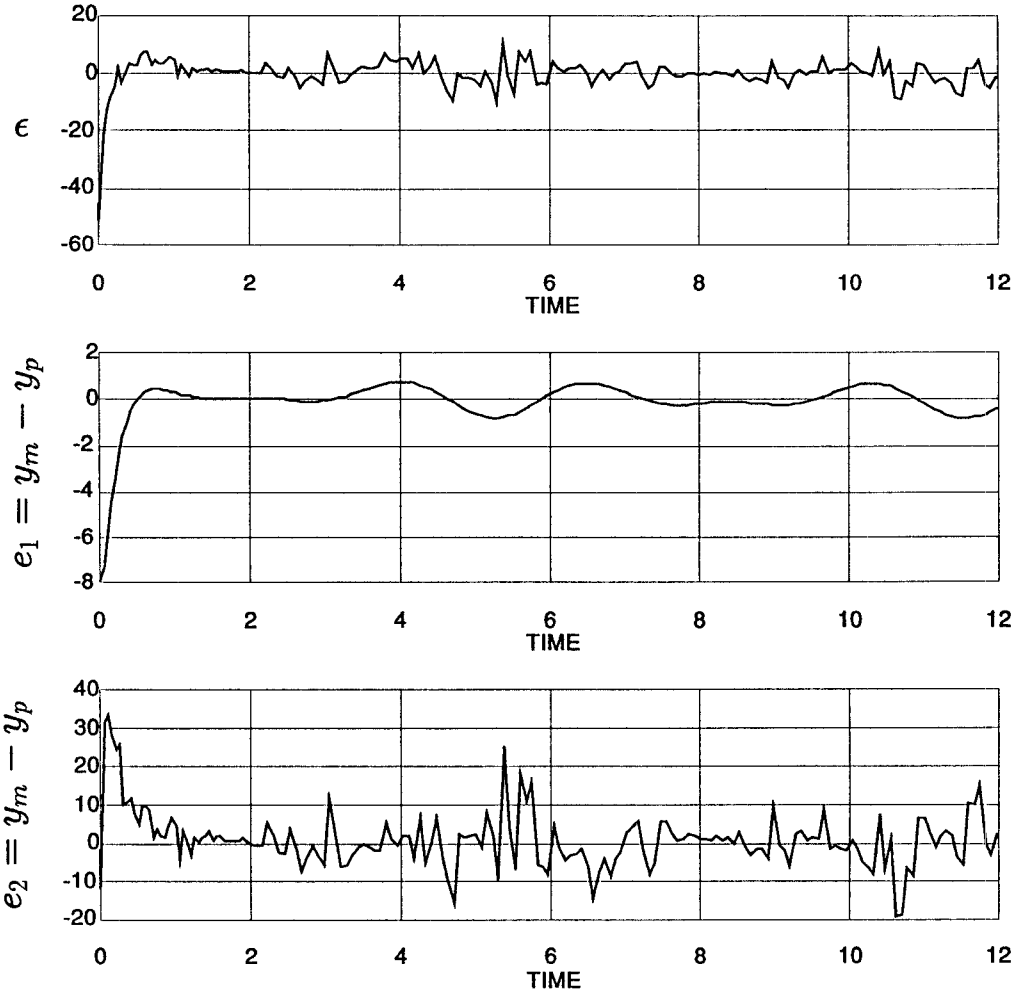

Figure 4.16: Error Trajectories in Example 2, Case 2

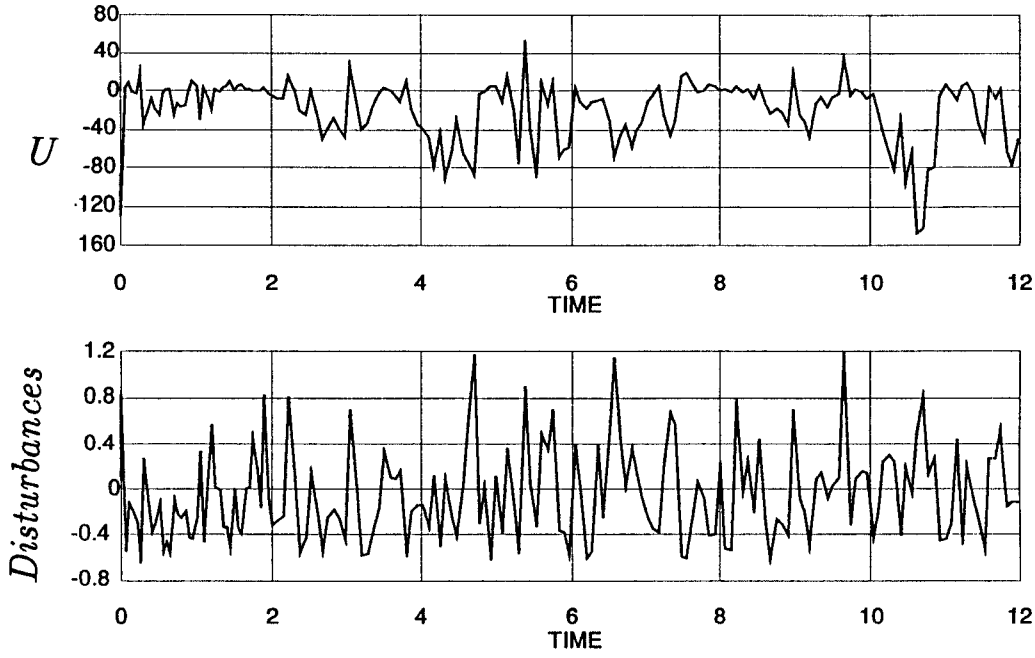

Figure 4.17: Control u and Disturbances v in Example 2, Case 2

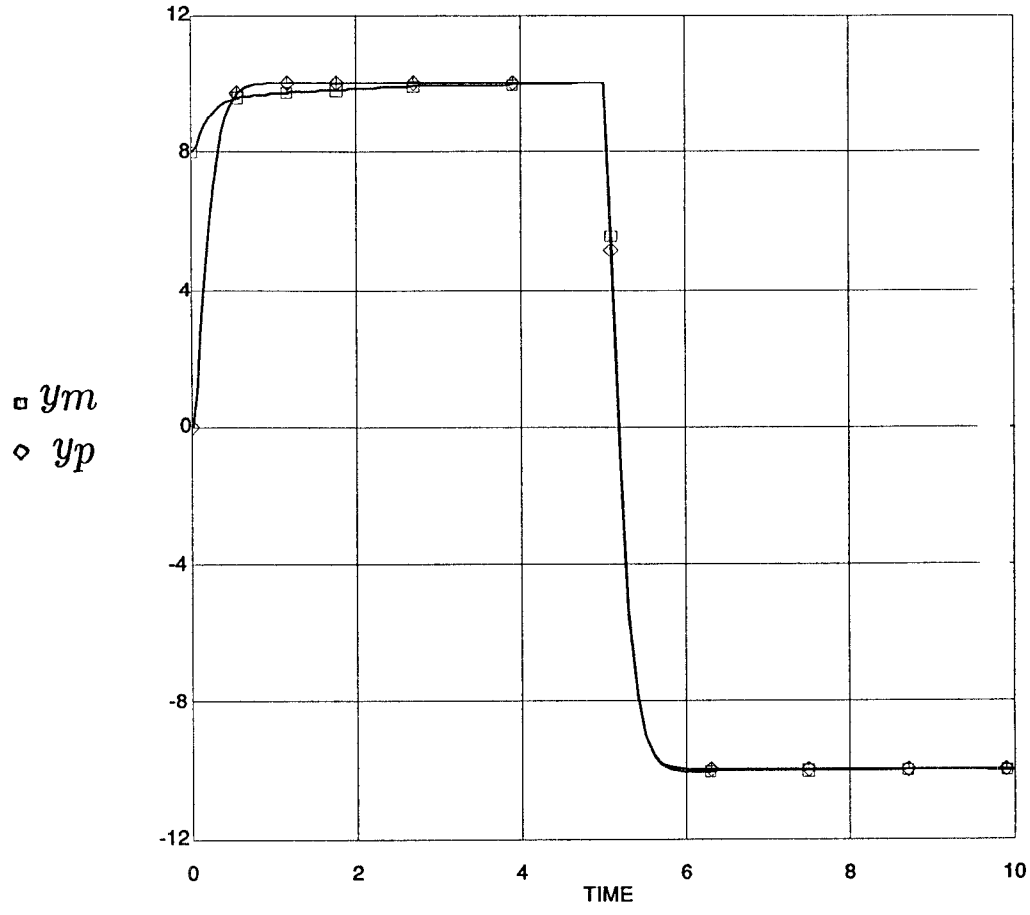

Figure 4.18: Desired Model and Plant Output Trajectories: y_m and y_p in Example 2, Case 3

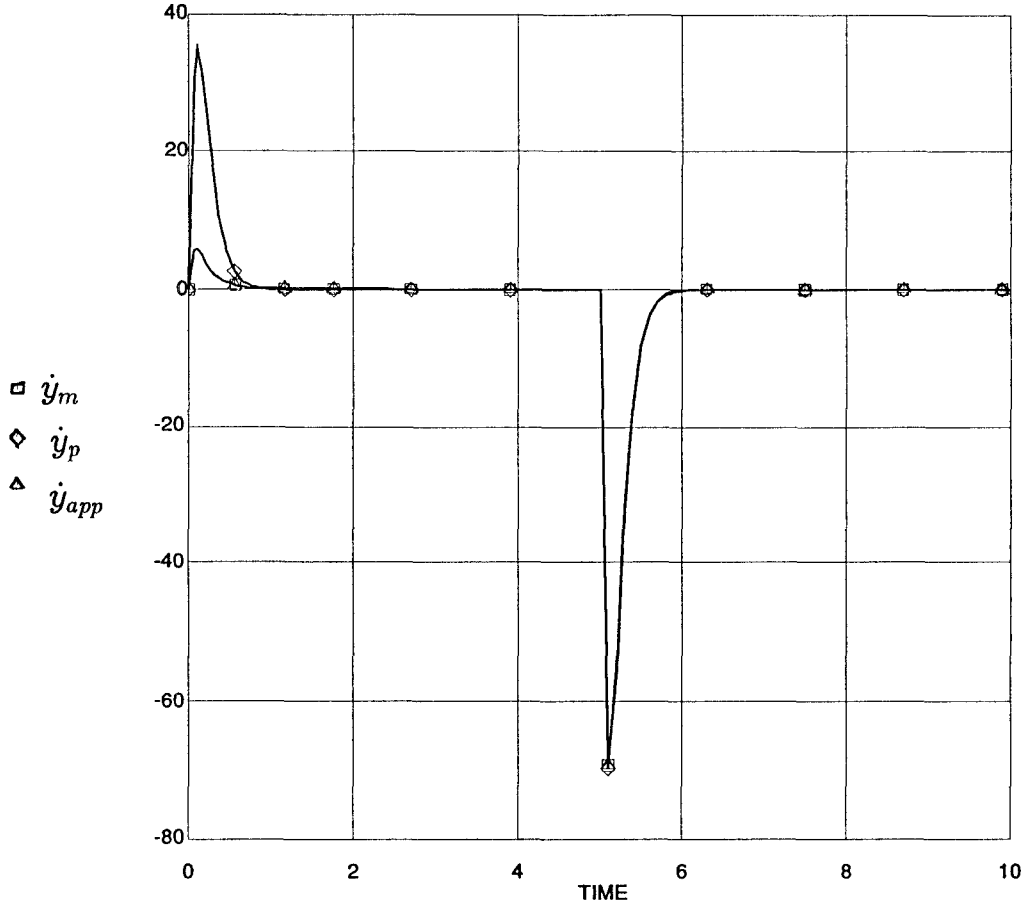

Figure 4.19: Desired Model and Plant Output Derivative Trajectories: $\dot{y}_m$ and $\dot{y}_p$ in Example 2, Case 3

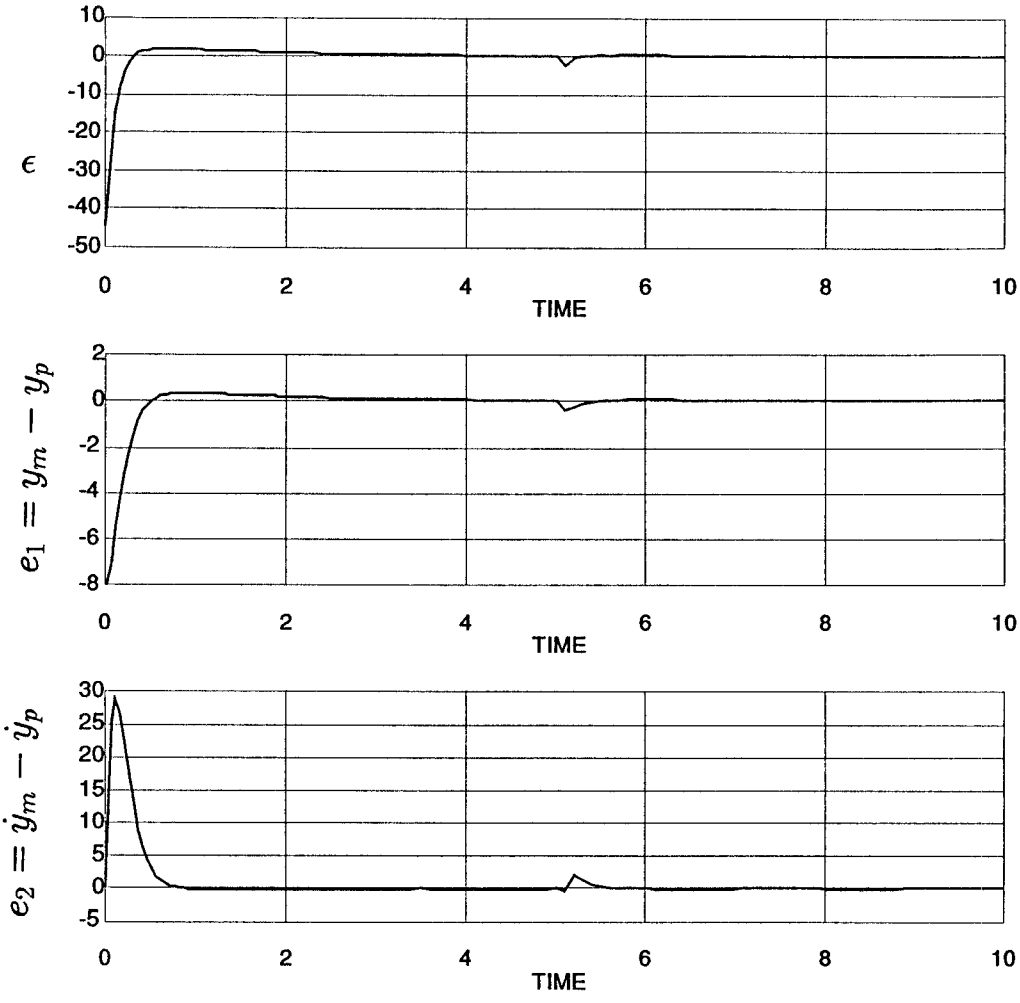

Figure 4.20: Error Trajectories in Example 2, Case 3

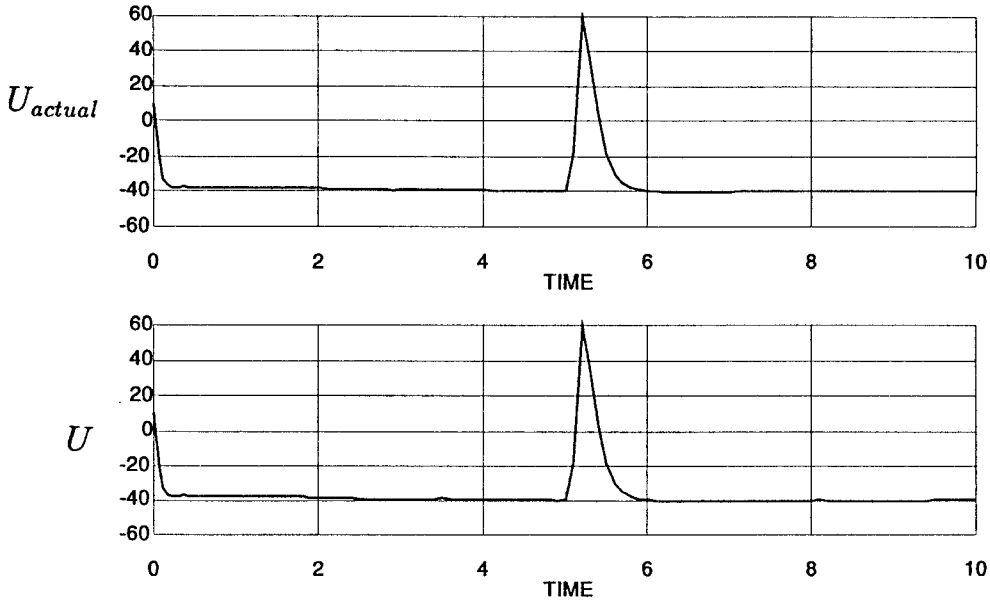

Figure 4.21: Computed Control U and Actual Control U_{actual} in Example 2, Case 3

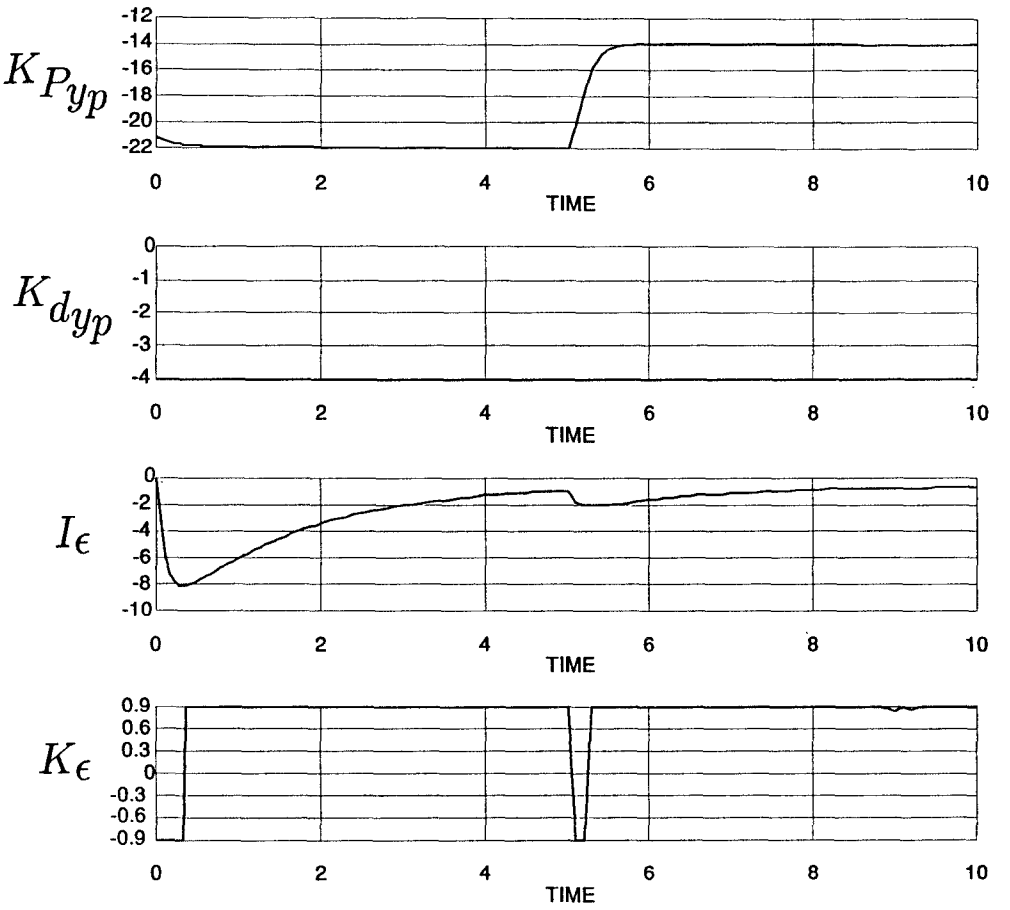

Figure 4.22: Controller Parameters K_{Py_p}, K_{dy_p}, I_ϵ, K_ϵ, in Example 2, Case 3

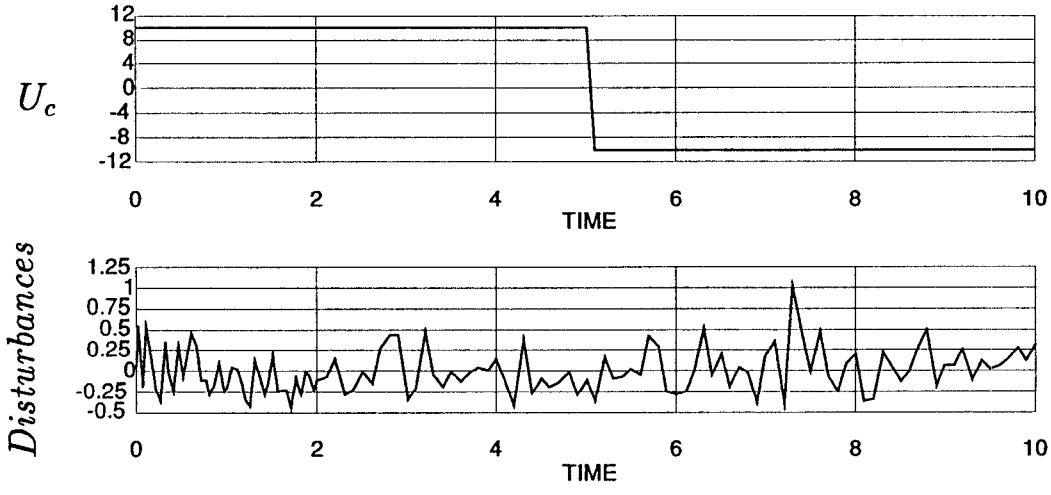

Figure 4.23: Command Input Signal u_c and Disturbances v in Example 2, Case 3

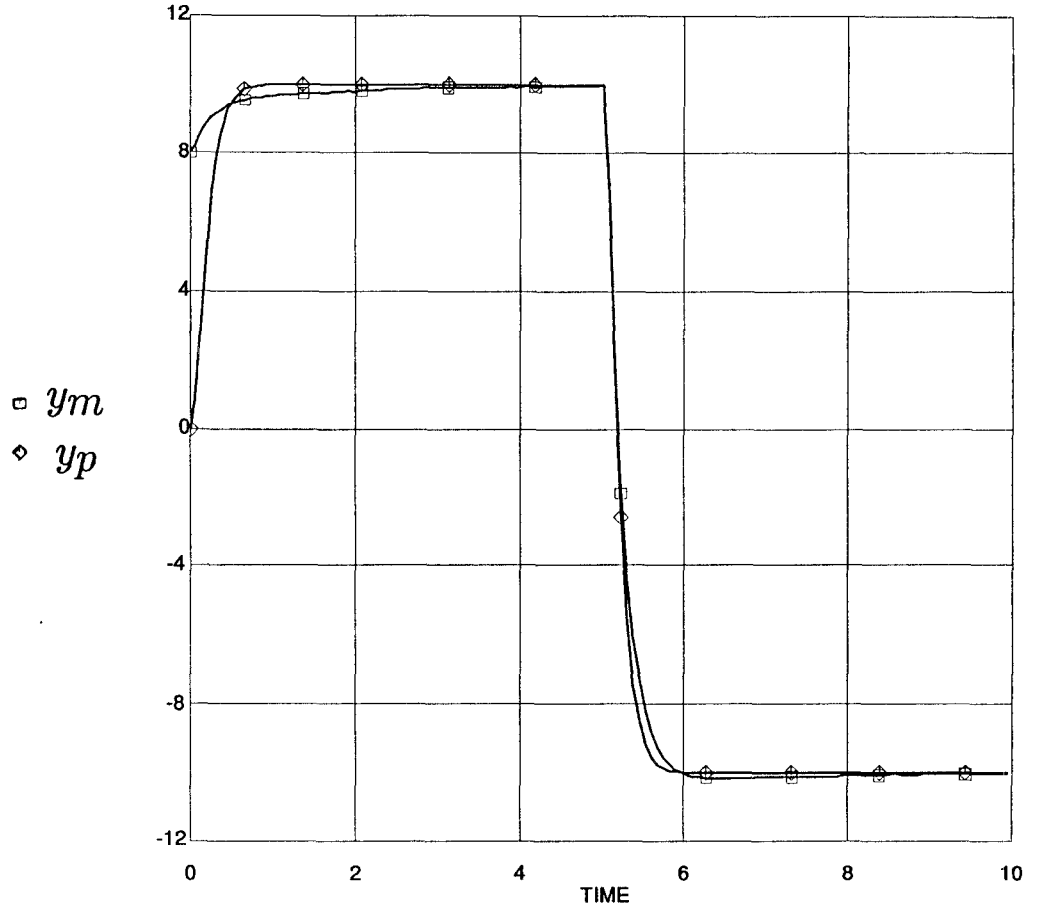

Figure 4.24: Desired Model and Plant Output Trajectories: y_m and y_p in Example 2, Case 4

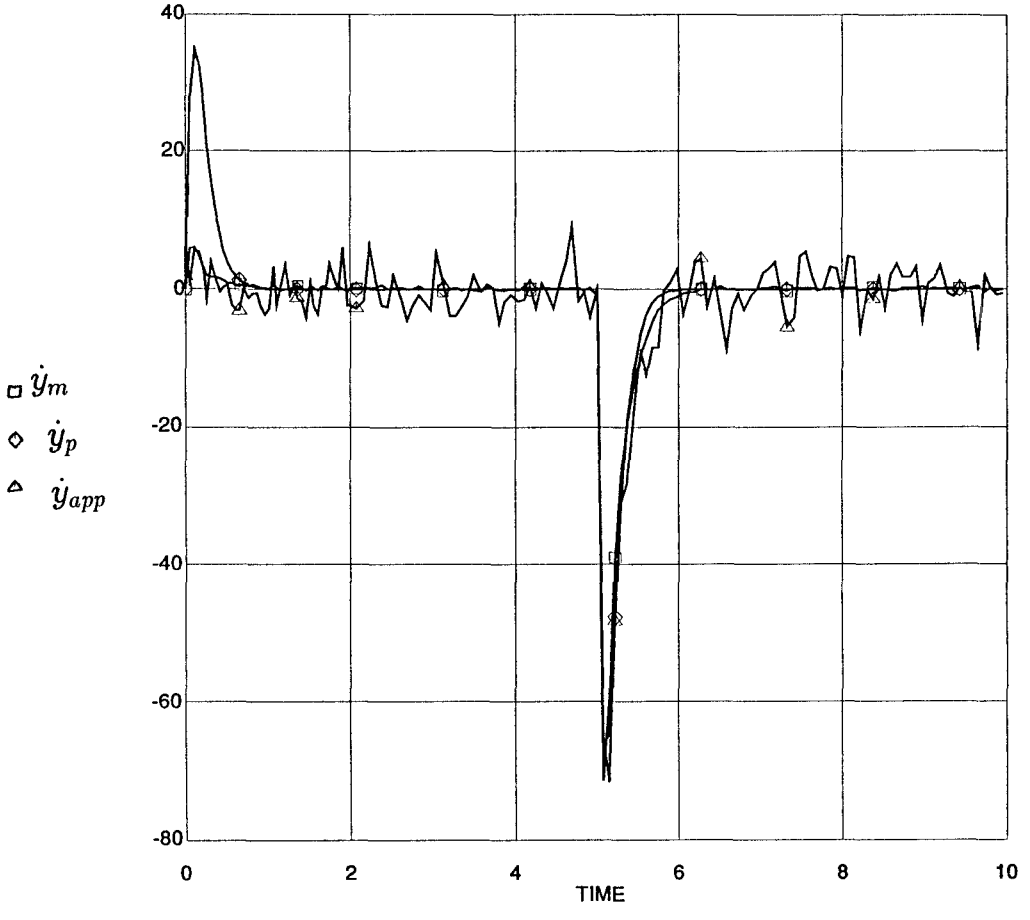

Figure 4.25: Desired Model and Plant Output Derivative Trajectories: $\dot{y}_m$ and $\dot{y}_p$ in Example 2, Case 4

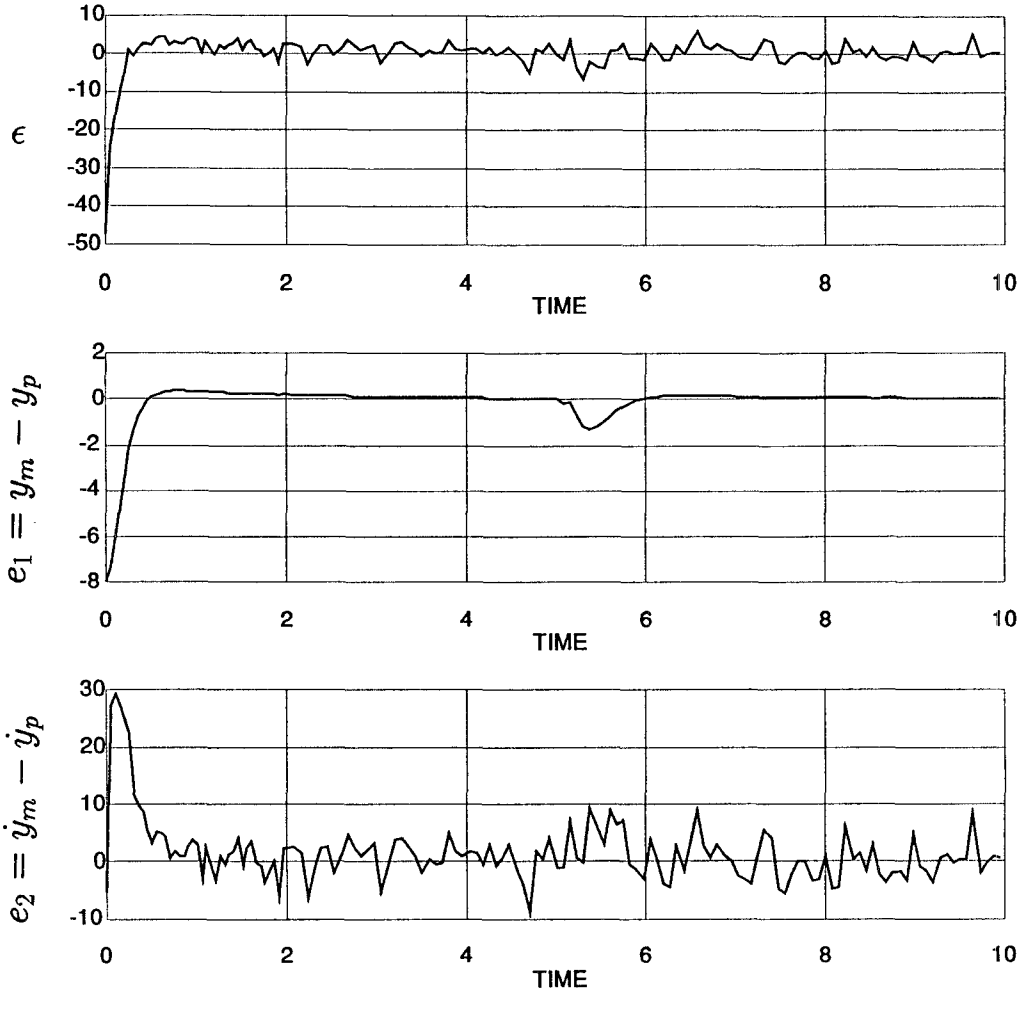

Figure 4.26: Error Trajectories in Example 2, Case 4

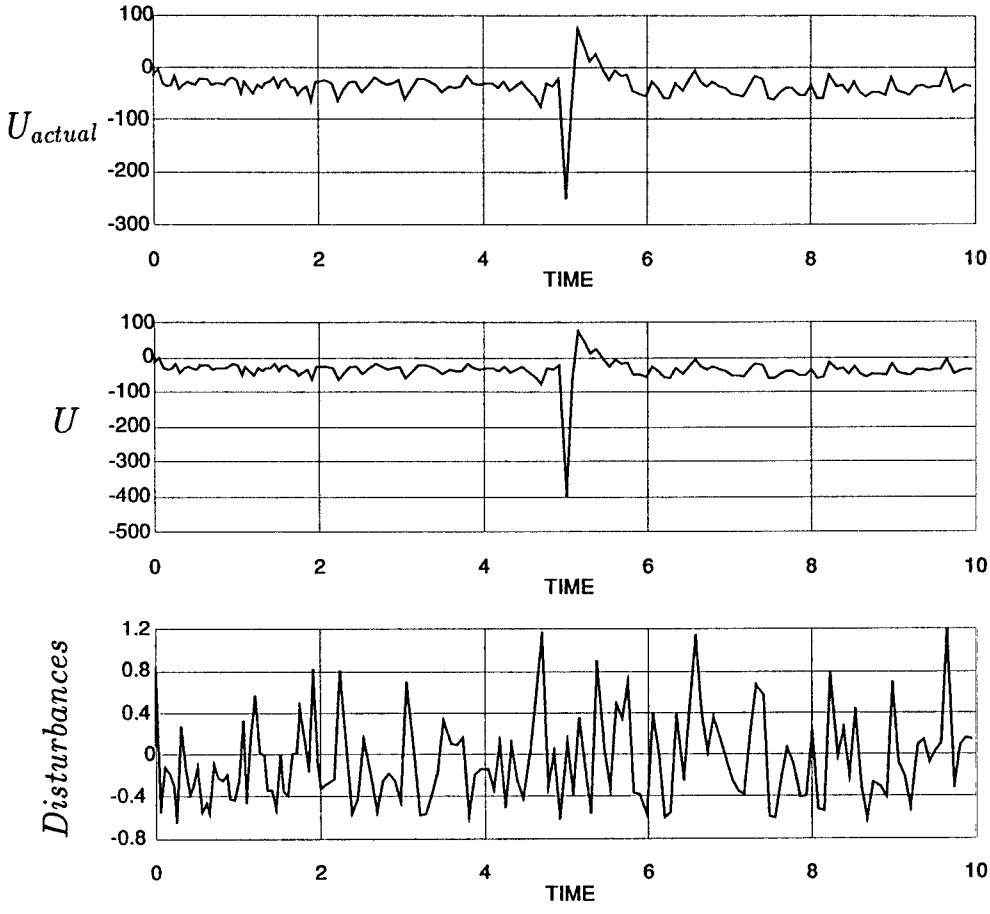

Figure 4.27: Computed Control U, Actual Control U_{actual}, and Disturbances v in Example 2, Case 4

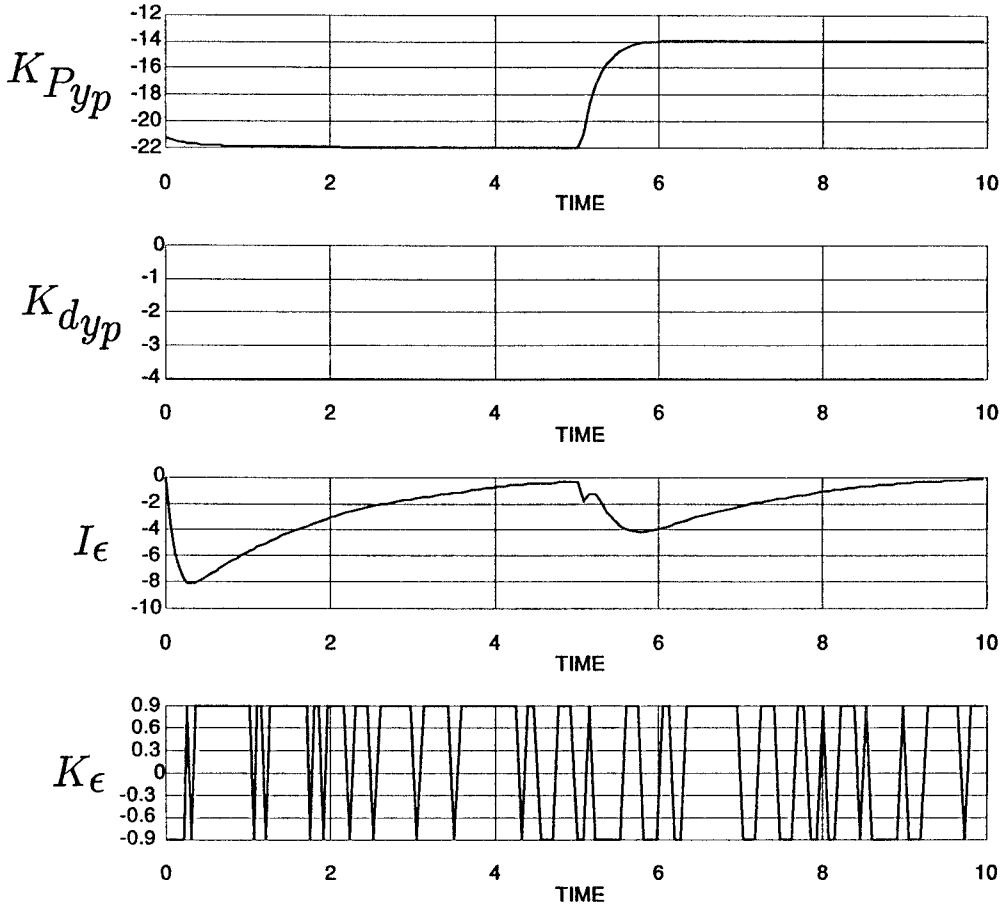

Figure 4.28: Controller Parameters $K_{P_{y_p}}$, $K_{d_{y_p}}$, I_ϵ, K_ϵ, in Example 2, Case 4

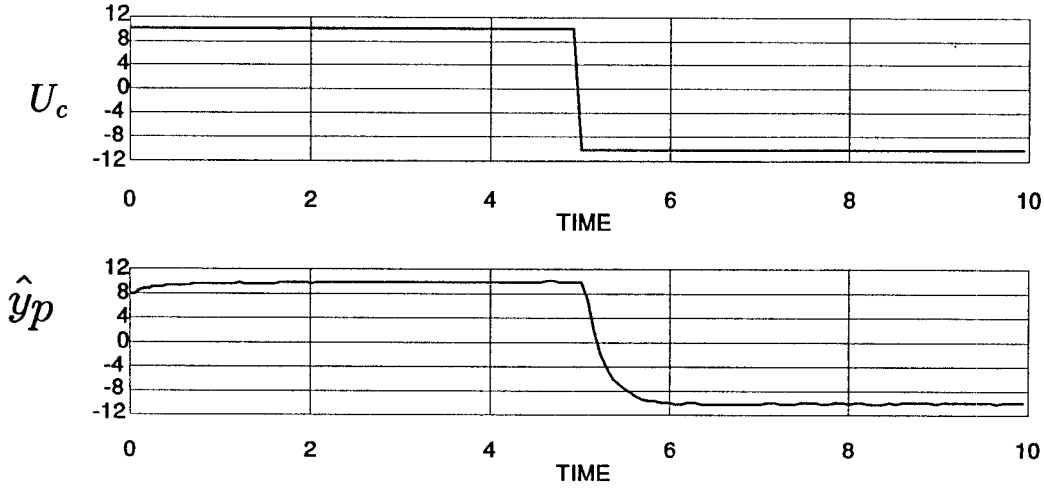

Figure 4.29: Command Input Signal u_c and Plant Output with Noise $\hat{y}_p$ in Example 2, Case 4

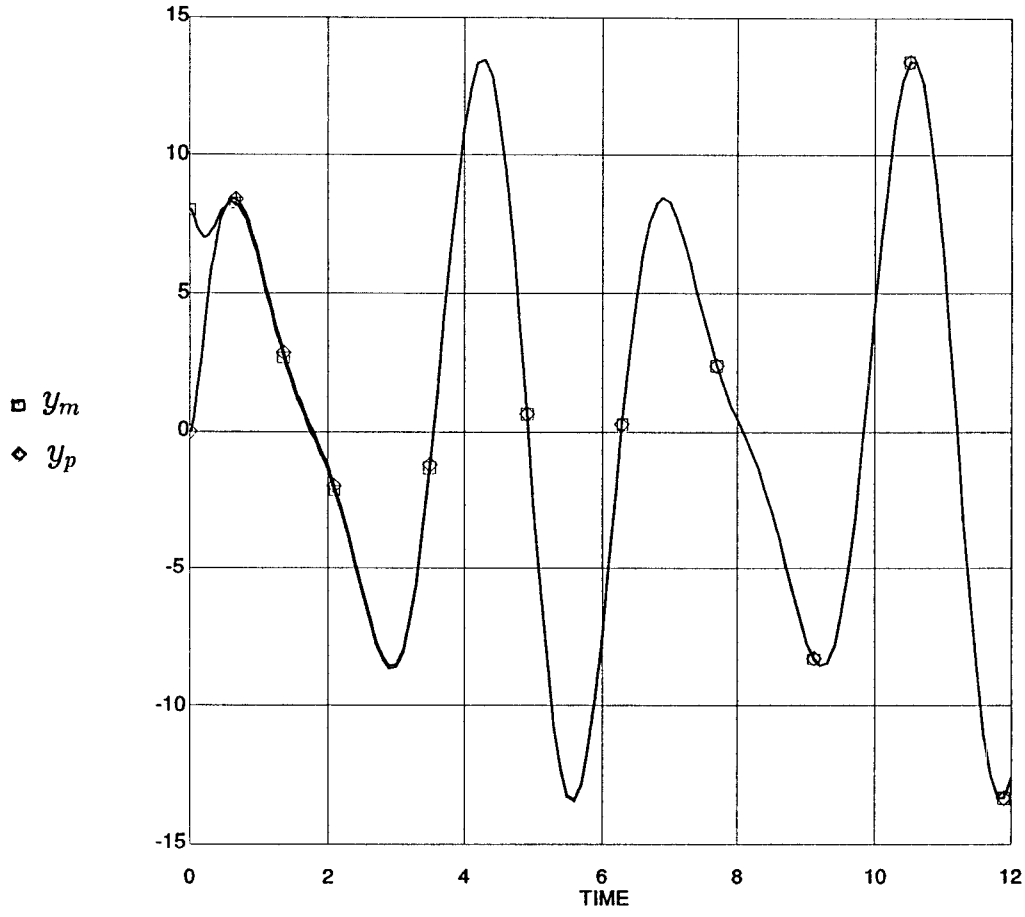

Figure 4.30: Desired Model and Plant Output Trajectories: y_m and y_p in Example 3, Case 1

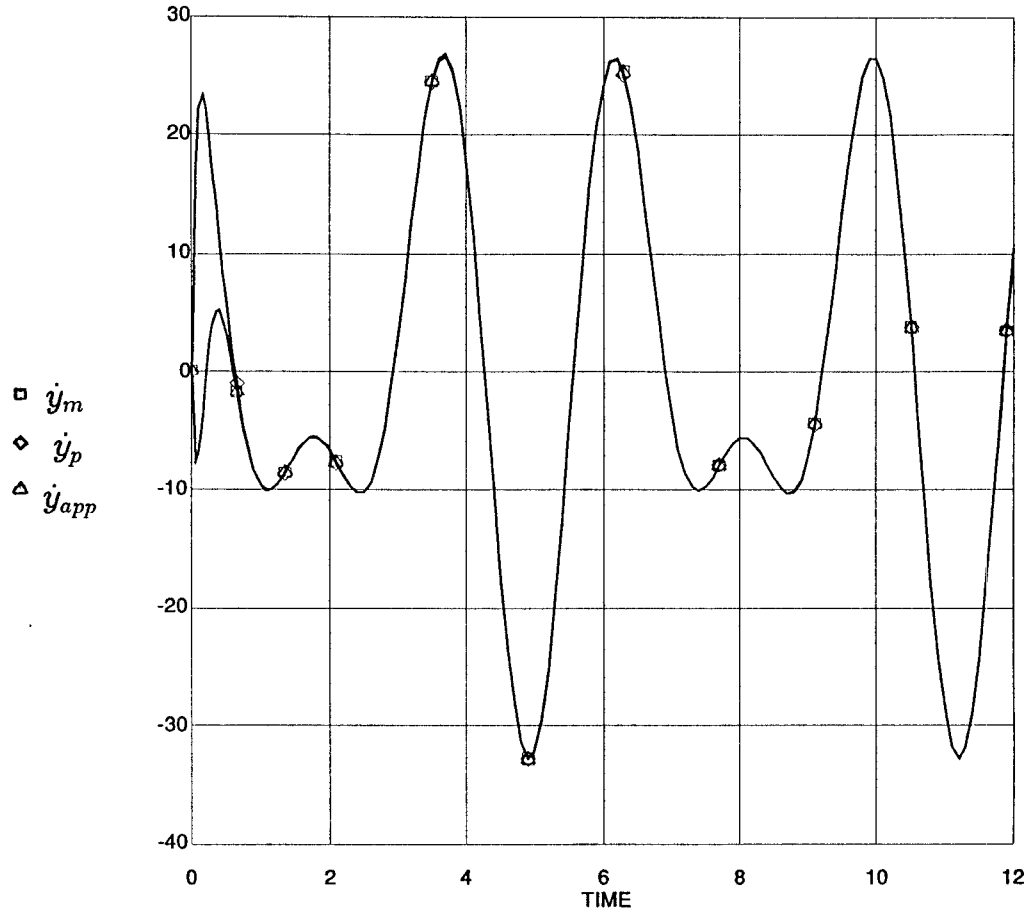

Figure 4.31: Desired Model and Plant Output Derivative Trajectories: $\dot{y}_m$ and $\dot{y}_p$ in Example 3, Case 1

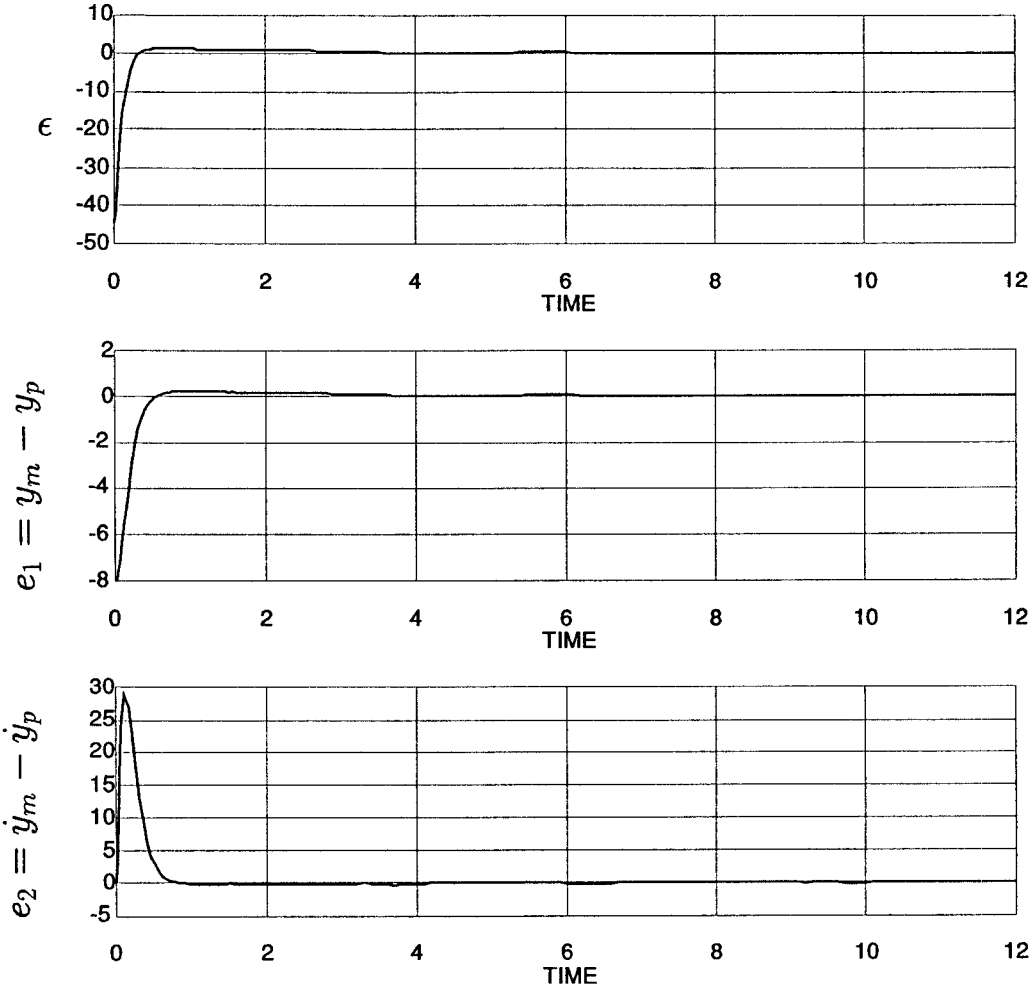

Figure 4.32: Error Trajectories in Example 3, Case 1

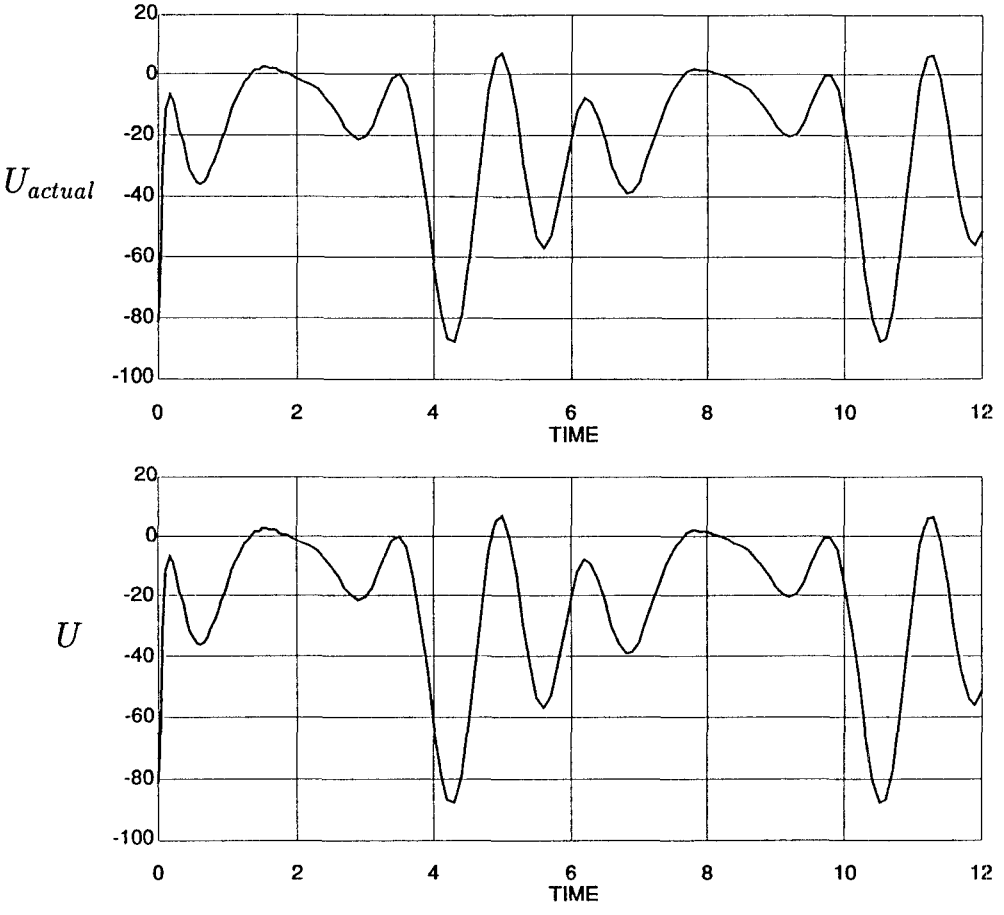

Figure 4.33: Control U in Example 3, Case 1

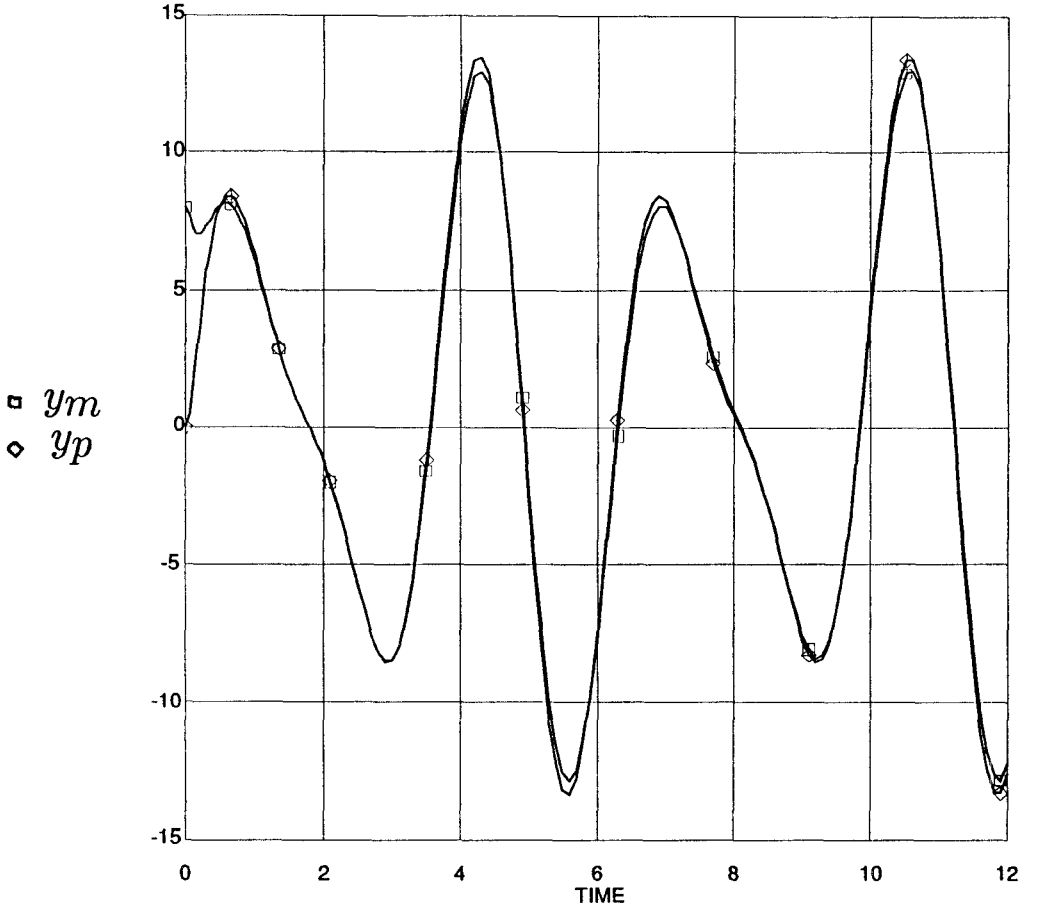

Figure 4.34: Desired Model and Plant Output Trajectories: y_m and y_p in Example 3, Case 2

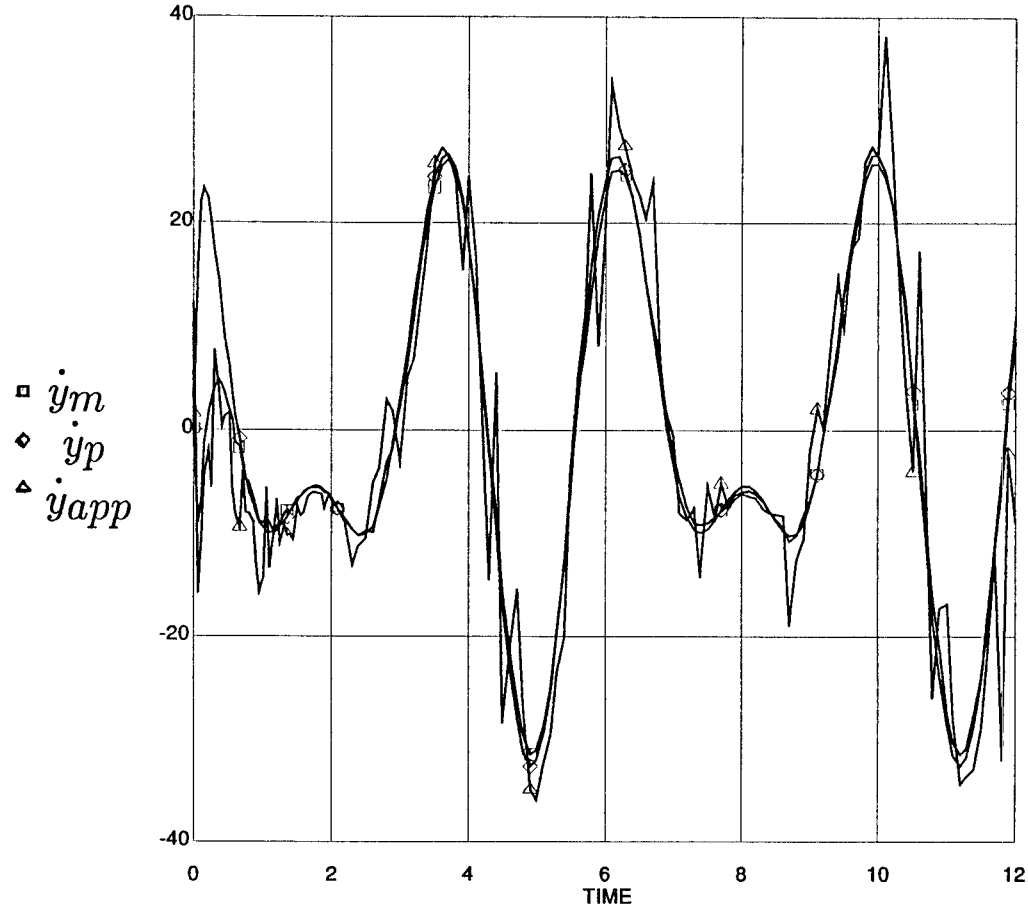

Figure 4.35: Desired Model and Plant Output Derivative Trajectories: $\dot{y}_m$ and $\dot{y}_p$ in Example 3, Case 2

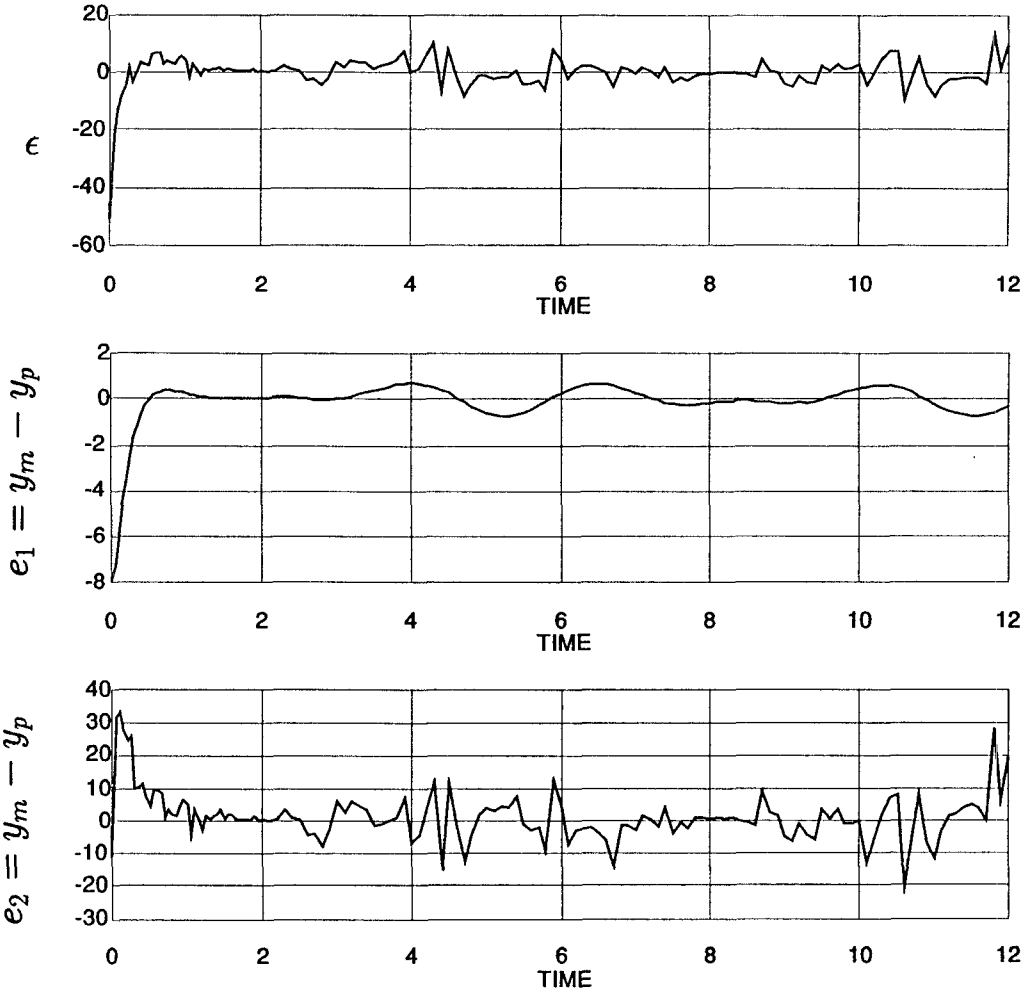

Figure 4.36: Error Trajectories in Example 3, Case 2

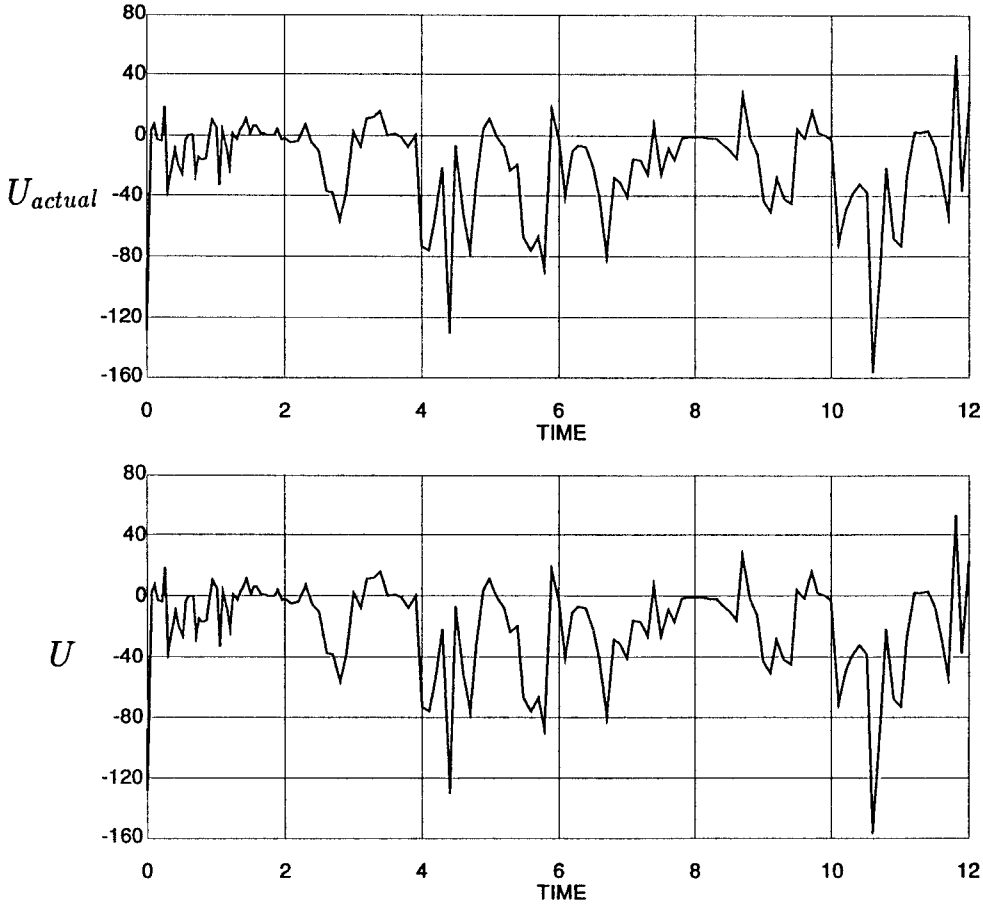

Figure 4.37: Control U in Example 3, Case 2

Chapter 5

Conclusions and Future Research

5.1 Conclusions:

In this thesis we presented a systematic way to design an adaptive PID controller for nonlinear systems where the dominant dynamics are of second order. The structure and tuning of the proposed adaptive controller is very straight forward and is given in terms of simple formulas.

While conventional PID controllers and classical adaptive PID controllers can be used only if the controlled plant can be considered linear and time-invariant over a certain operating range during the tuning and adaptation processes, the

design scheme and tuning rules presented in this thesis are developed and justified for *nonlinear time-varying and/or uncertain plants.* The uncertain nonlinearities, however, are assumed to be bounded by a known constant or a function of the plant output and its derivative. Accordingly, the resulting adaptive PID controller can handle systems with completely or partially known nonlinearities without losing the simple structure of a typical PID algorithm. The *disturbances* in the system are dealt with in a systematic way within the design scheme and they are assumed to be bounded by a known parameter.

In Chapter Two, we introduced the basics of the PID control and presented several tuning techniques for PID controllers. The well-known open-loop and closed-loop Ziegler-Nichols classical tuning methods were reviewed. The relay feedback experiment of Astrom and Hagglund that gives the essential information about the process dynamics necessary for the tuning of a PID controller was described. We showed how simple conventional process identification techniques can be combined with PID tuning rules to give *autotuning.* With the help of microprocessor technology, tuning procedure is therefore performed automatically on demand from an operator. We also reviewed the *self-tuning regulator* (STR) problem and described how STR can be employed in the design of PID controllers. We also showed that STR is obtained by combining an on-line *estimation* routine with the controller *underlying design problem.* A pole-placement approach was used in order to illustrate the STR technique.

In Chapter Three, we turned our attention to nonlinear time-varying and uncertain plants. We described the *model reference adaptive control* (MRAC) problem and obtained an error model for the adaptive system that describes the evolution of the process errors relative to some desired trajectories. We showed how the design objectives and performance criteria can be stated within a desired *reference* model that describes the behavior expected from the controlled process. We noted that the purpose of the adaptive PID controller is to assure that the plant output follows that of the reference model; and for this we employed the Lyapunov stability theory to seek the conditions upon which the asymptotic stability of the error system is guaranteed.

For the SISO case we used the asymptotic stability conditions of the error system to construct a PID-type controller scheme and derived its corresponding tuning rules. We then proceeded to obtain an estimate of the decay rate for the error in the adaptive loop and found that this rate is constrained to the reference model characteristics. We showed how the designer can increase this rate and find a highest possible rate once the reference model is chosen. We slightly modified the algorithm in order to get a smooth continuous PID control law and summarized the results in equations (3.63) and (3.64).

The MIMO case was simply a generalization of our investigation in the SISO case. We developed a design scheme for a system composed of n interacting subprocesses, each of which was controlled by a simple PID controller. We

showed the closed-loop stability of the adaptive system subject to these controls.

In Chapter Four, we illustrated the performance of the proposed adaptive PID controller. We looked at the behavior of several nonlinear systems subject to uncertainty and random disturbances. Although these systems were open-loop unstable, we saw that the desired trajectories were followed as it was expected through our stability analysis for the closed-loop behavior of the overall adaptive system. The simulations showed that the approximations we made in Chapter 3 in order to get a smooth and continuous control worked very well. Moreover, we saw the effect of the output measurement noise on the derivative estimation which led to slight degradation of the performance of the adaptive system.

5.2 Future Research:

There are several areas of investigation which need to be explored. Although extensive simulations have confirmed the usefulness of the suggested algorithm in this thesis, laboratory tests and implementation of the design scheme must be carried out in order to see the usefulness of the proposed algorithm in industrial control practice.

For systems with high order dominant dynamics, PID control is generally not adequate; and accordingly, upgrading the existing PID design schemes that handle dominant high frequencies needs to be further explored. This will lead

undoubtedly to higher order and more complex controllers. As mentioned in Chapter Two, the simple structure of PID controllers limit their performance, and systems with large delays or with complex dynamics are hard to control with these controllers. For these systems, incorporating deadtime compensators or adaptive predictive algorithms in a nonlinear setting is therefore needed.

One can further automate the control process by introducing an *identification* scheme in the adaptive loop to estimate a *range* for some of the plant parameters. This could effectively help designers to employ tighter bounds in the design scheme and result in a less costly and smoother control.

Bibliography

[ACG84] G. Ambrosino, G. Celentano, and F. Garofalo. Variable structure model reference adaptive control systems. *Int. J. Control*, 39(6):1339, 1984.

[AH84] K. J. Astrom and T. Hagglund. Automatic tuning of simple regulators with specifications on phase and amplitude margins. *Automatica*, 20(5):645, 1984.

[AH88] K. J. Astrom and T. Haglund. *Automatic Tuning of PID Controllers.* Instrument Society of America, 1988.

[AM79] B. Anderson and J. Moore. *Optimal Filtering.* Prentice-Hall, Inc., 1979.

[AM85] S. Arimoto and F. Miyazaki. Stability and robustness of pid feedback control for robot manipulators of sensory capability. *Third International Symposium of Robotics Research, Gouvieux, France*, July 1985.

[And81] N. Andreiev. A new dimention: A self-tuning controller that continually optimizes pid controllers. *Control Eng.*, page 84, August 1981.

[Ast80] K. J. Astrom. Self-tuning regulators-design principles and applications. In *Applications of Adaptive Control.* Academic Press, 1980.

[Ast83] K. J. Astrom. Theory and applications of adaptive control-a survey. *Automatica*, 19:471, 1983.

[Ast87] K. J. Astrom. Adaptive feedback control. *Proceedings of the IEEE*, 75(2), February 1987.

[AW89] K. J. Astrom and B. Wittenmark. *Adaptive Control.* Addison-Wesley, 1989.

[B+84] A. Balestrino et al. Nonlinear adaptive model-following control. *Automatica*, 20:559, 1984.

[Bes89] Y. Bestaoui. Decentralised pd and pid robotic regulators. *IEE Proceedings*, 136, pt.D(4), July 1989.

[Bou89] R. Boult. Predictive control takes over where pid leaves off. *Control Eng.*, page 67, December 1989.

[Bri77] E. H. Bristol. Pattern recognition: An alternative to parameter identification in adaptive control. *Automatica*, 13:197, 1977.

[CG75] D. W. Clarke and P. J. Gawthrop. A self-tuning controller. *Proc. Inst. Elec. Eng.*, 122:929, 1975.

[CG79] D. W. Clarke and P. J. Gawthrop. Self-tuning control. *Proc. Inst. Elec. Eng.*, 126:633, 1979.

[CG81] D. W. Clarke and P. J. Gawthrop. Implementation and applications of microprocessor-based self-tuners. *Automatica*, 17:233, 1981.

[Cha87] V. V. Chalam. *Adaptive Control Systems.* Marcel Dekker, New York, 1987.

[Cla81] D. W. Clarke. Introduction to self-tuning controllers. In *Self-tuning and Adaptive Control.* Peter Peregrinus, Ltd., London, UK, 1981.

[CPP84] M. La Cava, G. Paletta, and C. Picardi. Stability analysis of pwm feedback control systems with pid regulators. *Int. J. Control*, 39(5):987, 1984.

[Dea90] H. Deardon. Self-tuning aids plant optimisation. *Control and Instrumentation*, page 85, April 1990.

[Erz68] H. Erzberger. Analysis and design of model following systems by state space techniques. *Proc. joint ACC, Ann Arbor*, page 572, 1968.

[Fie62] W. B. Field. Adaptive three-mode controller. *ISA J.*, 9(2):30, 1962.

[Gaw77] P. J. Gawthrop. Some interpretations of the self-tuning controller. *Proc. IEE*, 124:889, 1977.

[Gaw82] P. J. Gawthrop. Using the self-tuning controller to tune pid regulators. *Susex Univ. Int. Report. CE/T/2*, 1982.

[Gaw86] P. J. Gawthrop. Self-tuning pid controllers: Algorithms and implementation. *IEEE Trans. Automatic Control*, 31(3):201, 1986.

[Gaw87] P. J. Gawthrop. *Continuous-Time Self-Tuning Control-Volume 1: Design.* Reaserch Studies Press, 1987.

[Gaw88] P. J. Gawthrop. Implementation of continuous-time controllers. In *Implementation of Self-Tuning Controllers.* 1988.

[Gel74] Arthur Gelb. *Applied Optimal Estimation.* MIT Press, 1974.

[Ger87] J. P. Gerry. A comparison of pid control algorithms. *Control Eng.*, page 102, March 1987.

[GO68] P. W. Gallier and R. E. Otto. Self-tuning computer adapts ddc algorithms. *Instru. Tech.*, page 65, February 1968.

[GS84] G. C. Goodwin and K. S. Sin. *Adaptive Filtering, Prediction, and Control.* Prentice-Hall, 1984.

[Har90] I. Hardie. Auto-tuned control. *Control and Instrumentation*, page 33, March 1990.

[Haw83] W. M. Hawk. A self-tuning, self-contained pid controller. *Proc. ACC, San Francisco*, page 838, 1983.

[HB81] Harris and Billings. *Self-Tuning and Adaptive Control.* Peter Peregrinus, Ltd., London, UK, 1981.

[HW50] P. Hazebroek and B. L. Van Der Waerden. Theoretical considerations on the optimum adjustment of regulators. *ASME Trans.*, page 309, April 1950.

[Ise81] R. Isermann. *Digital Control Systems.* Springer Verlag, Berlin, 1981.

[Kai80] Thomas Kailath. *Linear Systems.* Lecture Notes in Control and Information Sciences. Prentice Hall, Inc., 1980.

[Kal58] R. E. Kalman. Design of a self-optimizing control system. *ASME Trans.*, 80:468, 1958.

[KM84] T. W. Kraus and T. J. Myron. Self-tuning pid controller uses patern recognition approach. *Control Eng.*, page 106, June 1984.

[Kom89] E. J. Kompass. Pid control software: Features and comparisons. *Control Eng.*, page 155, September 1989.

[KS85] H. N. Koivo and J. Sorvari. On-line tuning of a multivariable pid controller for robot manipulators. *Proc. 24th IEEE Conf. on Decision and Control, Fort Lauderdale, FL*, page 1502, 1985.

[Lan79] I. D. Landau. *Adaptive Control-The Model Reference Approach.* Marcel Dekker, New York, 1979.

[LC73] D. P. Lindorff and R. L. Caroll. Survey of adaptive control using lyapunov design. *Int. J. Control*, (18):897, 1973.

[Leo87] C. T. Leondes. *Control and Dynamic Systems, Advances in Theory and Applications*, volume 25. Academic Press, 1987.

[Lju87] Lennart Ljung. *System Identification: Theory for the User.* Prentice Hall, Inc., 1987.

[LS83] Lennart Ljung and T. Soderstrom. *Theory and Practice of Recursive Identification.* MIT Press, 1983.

[Moh88] C. Mohtadi. Numerical algorithms in self-tuning control. In *Implementation of Self-Tuning Controllers.* 1988.

[Mon74] R. V. Monopoli. Model reference adaptive control with an augmented error signal. *IEEE Trans. Automatic Control*, AC-19:474, 1974.

[Mor80] A. S. Morse. Global stability of parameter-adaptive control systems. *IEEE Trans. Automatic Control*, AC-25:433, 1980.

[Mor87] H. M. Morris. How adaptive are adaptive process controllers? *Control Eng.*, page 96, March 1987.

[N+84] Y. Nishikawa et al. A method for auto-tuning of pid control parameters. *Automatica*, 20(3):321, 1984.

[N+85] K. S. Narendra et al. A general approach to the stability analysis of adaptive systems. *Int. J. Control*, 41:193, 1985.

[Nar86] K. S. Narendra. *Adaptive and Learning Systems, Theory and Applications.* Plenum Press, New York, 1986.

[NL80] K. S. Narendra and Y. Lin. Design of stable modern reference adaptive controllers. In *Applications of Adaptive Control.* Academic Press, 1980.

[Nom88] P. E. Nomikos. Multivariable self-tuning controllers for industrial applications. 1988. D. Phil. Thesis, Univ. of Sussex, England.

[Nor86] J. P. Norton. *An Introduction to Identification.* Academic Press, 1986.

[OK84] R. Ortega and R. Kelly. Pid self-tuners: Some theoretical and practical aspects. *IEEE Trans. on Industrial Electronics*, IE-31(4):332, 1984.

[Ort70] K. Ortega. *Modern Control Engineering.* Prentice Hall, Inc., 1970.

[OWK69] P. V. Osburn, H. P. Whitaker, and A. Kezer. New developments in the design of adaptive control systems. *Inst. Aeronautical Sciences*, February 1969. Paper No. 61-39.

[P+83] C. G. Proudfoot et al. Self-tuning pi control of a ph neutralization process. *IEE Proc.*, 130, pt. D(5):267, 1983.

[Par66] P. C. Parks. Lyapunov redesign of model reference adaptive control systems. *IEEE Trans. Automatic Control*, AC-11:362, 1966.

[Par81] P. C. Parks. Stability and convergence of adaptive controllers- continuous systems. In *Self-tuning and Adaptive Control.* Peter Peregrinus, Ltd., London, UK, 1981.

[RR82] B. Roffel and J. E. Rijnsdorp. *Process Dynamics, Control and Protection.* Ann Arbor Science Publishers, 1982.

[SB89] S. Sastry and M. Bodson. *Adaptive Control, Stability, Convergence and Robustness.* Prentice-Hall, Inc., 1989.

[Smi72] C. L. Smith. *Digital Computer Process Control.* Intext Educational Pablishers, 1972.

[Tin89a] Brian Tinham. Pid control and loop tuning. *Control and Instrumentation*, page 53, August 1989.

[Tin89b] Brian Tinham. Pid control and loop tuning. *Control and Instrumentation*, page 79, September 1989.

[Tza85] S. G. Tzafestas. Digital pid and self-tuning control. In *Applied Digital Control*, chapter 1. 1985.

[Unb80] H. Unbehauen. *Methods and Applications in Adaptive Control.* Springer Verlag, Berlin, 1980.

[War88a] K. Warwick. *Implementation of Self-Tuning Controllers.* Peter Peregrinus, Ltd., London, UK, 1988.

[War88b] K. Warwick. Simplified algorithms for self-tuning control. In *Implementation of Self-Tuning Controllers.* 1988.

[Wit79] B. Wittenmark. Self-tuning pid controllers based on poleplacement. *Lund Inst. of Technology, Report No. TFRT-7179*, 1979.

[Wit88] C. A. Canudas De Wit. *Adaptive Control for Partially Known Systems, Theory and Applications.* Elsevier Science, New York, 1988.

[WR88] K. Warwick and D. Rees. *Industrial Digital Control Systems.* Peter Peregrinus, Ltd., 1988.

[ZN42] J. G. Ziegler and N. B. Nichols. Optimum settings for automatic controllers. *ASME Trans.*, 64:759, November 1942.

[ZN43] J. G. Ziegler and N. B. Nichols. Process lags in automatic control circuits. *ASME Trans.*, page 433, July 1943.