ECE Technical Reports Electrical and Computer Engineering

4-1-1993

A Fuzzy Precompensator Design for PD Control of Systems with Deadzones

Jong-Hwan Kim
Purdue University School of Electrical Engineering

Jong-Hwan Park
Korea Advanced Institute of Science and Technology Department of Electrical Engineering

Seon-Woo Lee
Korea Advanced Institute of Science and Technology Department of Electrical Engineering

Edwin K. P. Chong
Purdue University School of Electrical Engineering

A Fuzzy Precompensator Design for PD Control of Systems with Deadzones

Jong-Hwan Kim
Jong-Hwan Park
Seon-Woo Lee
Edwin K. P. Chong

TR-EE 93-16
April 1993

School of Electrical Engineering
Purdue University
West Lafayette, Indiana 47907-1285

A Fuzzy Precompensator Design for PD Control of Systems with Deadzones

Jong-Hwan Kim* Jong-Hwan Park* Seon-Woo Lee*
Edwin K. P. Chong†

Abstract

Simple conventional control methods, such as PD and PID controllers, are widely used in industrial applications. Such controllers exhibit poor performance when applied to systems containing nonlinearities arising from unknown deadzones. In this report, we propose a novel fuzzy logic-based precompensation approach for controlling systems with deadzones. The control structure consists of a fuzzy logic-based precompensator followed by a conventional PD controller. Our proposed control scheme shows superior transient and steady-state performance compared to conventional PD and PID controllers. In addition, the scheme is robust to variations in deadzone nonlinearities, as well as the steady-state gain of the plant. We illustrate the effectiveness of our scheme using computer simulation examples.

'Dept. of Electrical Engineering, Korea Advanced Institute of Science and Technology (KAIST), 373-1 Kusung-dong, Yusung-gu, Taejon-shi 305-701, Republic of Korea. The first author is currently on sabbatical at Purdue University.

†School of Electrical Engineering, Purdue University, 1285 Electrical Engineering Bldg., West Lafayette, IN 47907-1285.

Introduction

We propose a fuzzy logic-based scheme for controlling systems with deadzones. Our control structure consists of a fuzzy precompensator and a standard PD controller. The idea underlying the control scheme is based on analyzing the source of large steady-state errors which arise when a conventional PD controller is applied to a system with a deadzone. Our proposed scheme has good transient as well as steady-state performance, and is robust to variations in deadzone nonlinearities.

Nonsmooth nonlinearities are common in many physical components in control systems, such as gears and hydraulic servovalves. Such nonlinearities include saturation, relays, hysteresis, and deadzones, and are often unknown and time varying. For example, a common source of nonlinearities arise from friction, which vary with temperature and wear, and may differ significantly between components which are mass produced. Therefore the study of methods for dealing with nonsmooth nonlinearities has been of interest to control practitioners for some time. In this report, we consider only deadzone nonlinearities. Deadzones are of interest in their own right, and provide good models for many nonsmooth nonlinearities found in practice.

Standard controllers used in practice, such as PD and PID controllers, suffer from poor performance when applied directly to systems with deadzone nonlinearities. For example, a steady-state error occurs when applying a conventional PD controller to a system with deadzones—the size of the steady-state error increases with the deadzone width (see Section 11.2). The steady-state error arises because a PD controller uses only the output error and the change in output error as inputs to the controller. To eliminate the steady-state error, we may attempt to use a PID controller, that also incorporates the "integral" of the output error as an input to the controller. However, as we shall see in Section 11.4, the transient performance in this case is poor.

More advenced control schemes for controlling systems with nonsmooth nonlinearities include sliding mode control [1], and dithering [2]. Motivated by limitations in these methods, such as chattering in sliding mode control, Recker et al. [3] proposed an adaptive nonlinear control scheme for controlling systems with deadzones. In [3], full state measurements were assume to be available. More recently, Tao and Kokotovic [4] considered the more realistic situation where only a single output measurement is available. In practice, however, the transient performance of the adaptive control schemes above is limited.

Fuzzy logic-based controllers have received considerable interest in recent years (see for example [5], [6], [7], [8], [9]). Fuzzy-based methods are useful when precise mathematical formulations are infeasible. Moreover, fuzzy logic controllers often yield superior results to conventional control approaches [7]. In [10], Kim et al. studied a fuzzy logic based controller applied to systems with deadzones. Their scheme exhibits superior transient and steady-state response compared to the schemes described above.

In this report we propose a fuzzy logic-based scheme for controlling systems with deadzones. Our present scheme is simpler and more practical than the one considered in [10]. The control structure we propose in this report consists of simply adding a fuzzy logic based precompensator to a standard PD controller. The idea underlying our approach is based on analyzing the source of the steady-state error resulting from using a PD controller alone. We demonstrate that our controller has excellent transient as well as steady-state performance, and is robust to variations in deadzone nonlinearities as well as the steady-state gain of the plant.

The remainder of this report is organized as follows. In Section II we describe a system with a deadzone, and study the characteristics of a conventional PD controller applied to the system. We show that the PD controller results in poor performance, and give an analysis of the source of steady-state errors. We also study the behavior of a PID controller applied to the same system. In Section III we propose our fuzzy logic precompensation scheme. We describe the idea underlying our approach, and give a precise description of the controller. We also provide simulation plots to illustrate the behavior of our scheme. Finally we conclude in Section IV.

II Characteristics of Conventional PD Controller

In this section we describe a general PD (Proportional-Derivative) controller, and study the behavior of the controller applied to a system with a deadzone.

II.1 Basic Control Structure

We consider the (discrete-time) system shown in Figure 1, which is a conventional PD control system. The transfer function $P(z)$ represents the plant, D represents an actuator with deadzone, $C[e(k), \Delta e(k)] = K_P e(k) + K_D \Delta e(k)$ is a linear function of the error and change of error representing a standard PD control law, K_1 is the feedforward gain, $v(k)$ is the output of the PD controller, $u(k)$ is the output of the actuator, $y_m(k)$ is the reference input (command signal to be followed), $y_p(k)$ is the output of the plant, $e(k)$ is a tracking error between $y_m(k)$ and $y_p(k)$, and $\Delta e(k)$ is the change in tracking error $e(k) - e(k-1)$. The characteristics of the actuator with deadzone D is described by the function

$$D[v] = \begin{cases} m(v-d), & \text{if } v > d \\ 0, & \text{if } -d \leq v \leq d \\ m(v+d), & \text{if } v < -d \end{cases}$$

where $d, m > 0$. Figure 2 illustrates the characteristics of the actuator with deadzone. The parameter 2d specifies the width of the deadzone, while m represents the slope of the response outside the deadzone.

11.2 Analysis of Steady-State System Behavior

We now study the steady-state behavior of the system controlled by the conventional PD controller. The purpose of the analysis is to illustrate a problem that arises in the presence of a deadzone. Specifically, we will show that in the presense of a deadzone, a steady-state error occurs in a Type 0 system controlled by a "well-tuned^n PD controller (while there is no steady-state error if there is no deadzone).

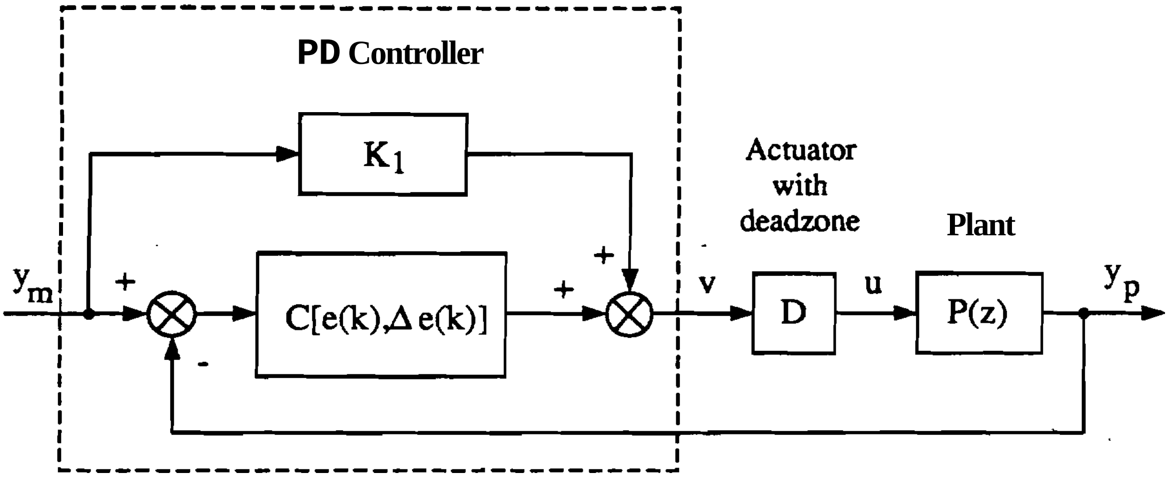


Figure 1: Conventional PD control system with deadzone

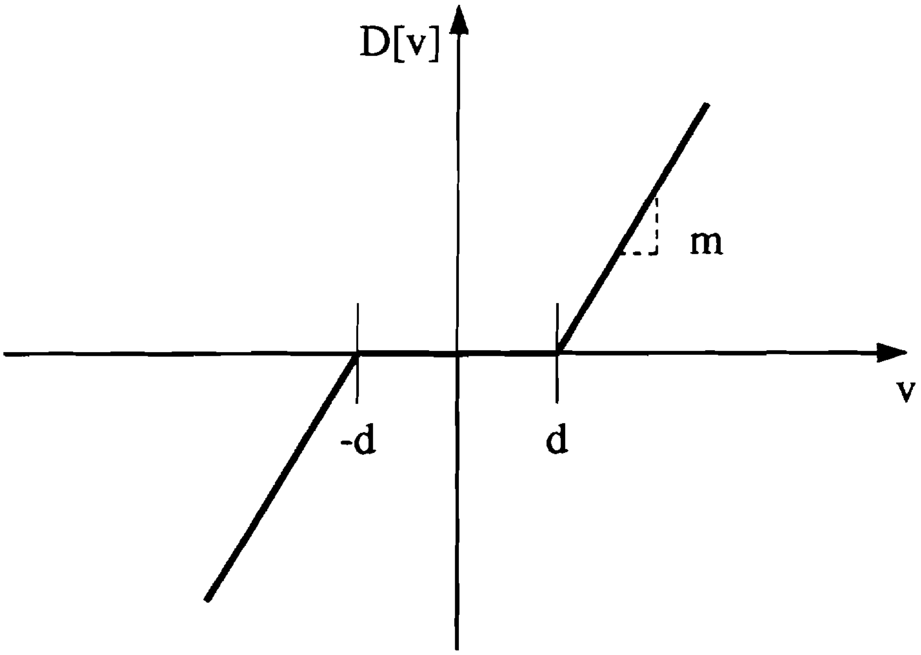


Figure 2: Characteristics of Actuator with deadzone

The dynamics of overall system are described by the following equations:

$$\begin{aligned}
e(k) &= y_m(k) - y_p(k) \\
\Delta e(k) &= e(k) - e(k-1) \\
C[e(k), \Delta e(k)] &= K_P e(k) + K_D \Delta e(k) \\
v(k) &= K_1 y_m(k) + C[e(k), \Delta e(k)] \\
u(k) &= D[v(k)] \\
y_p(k) &= P(z)[u(k)]
\end{aligned}$$

Note that the equation $y_p(k) = P(z)[u(k)]$ involves a slight abuse of notation; however, its meaning should be obvious. Since $C[0,0] = 0$, then if we fix the reference input $y_m(k) = y_m$, the steady-state actuator input is $K_1 y_m$.

Consider the case when there is no deadzone, i.e., d = 0, and m = 1. In this case the plant output can be written as

$$y_p(k) = P(z)[K_1 y_m(k) + C[e(k), \Delta e(k)]]$$

Since $e(k) = y_m(k) - y_p(k)$, then the plant output can also be written as

$$y_p(k) = y_m(k) - e(k)$$

We now fix $y_m(k) = y_m$, and study the behavior of the system in steady-state. We assume that the plant is of Type 0 (i.e., $P(z)$ does not have a pole at $z = 1$). To derive the equation for the steady-state behavior, we set $\Delta e(k) = 0$ to get

$$y_{p,ss} = K_s[K_1 y_m + K_P e_{ss}] = y_m - e_{ss} \tag{1}$$

where K, is the steady-state gain of $P(z)$ (assumed stable), given by K, $=$ $\lim_{z \to 1} P(z)$, $y_{p,ss}$ is the steady-state output, and e„ is the steady-state error. Note that K, $< \infty$ for a Type 0 plant. The steady-state error e„ is then the solution to equation (1), that is,

$$K_s[K_1 y_m + K_P e_{ss}] = y_m - e_{ss} \tag{2}$$

We assume that the controller is "well-tuned", so that $K_1 = K_s^{-1}$. Equation (2) then becomes

$$K_s K_P e_{ss} = -e_{ss} \tag{3}$$

It is clear that the solution to the above equation is simply $e_{ss} = 0$, i.e., the steady-state error is zero, as expected.

We now consider the case when a deadzone is present, i.e., d > 0, and m > 0 are arbitrary. In this case, the steady-state output of the plant can be written as

$$y_{p,ss} = K_s D[K_1 y_m + K_P e_{ss}] = y_m - e_{ss}$$

Therefore, the steady-state error is the solution to the equation

$$K_s D[K_1 y_m + K_P e(k)] - y_m = -e_{ss} \tag{4}$$

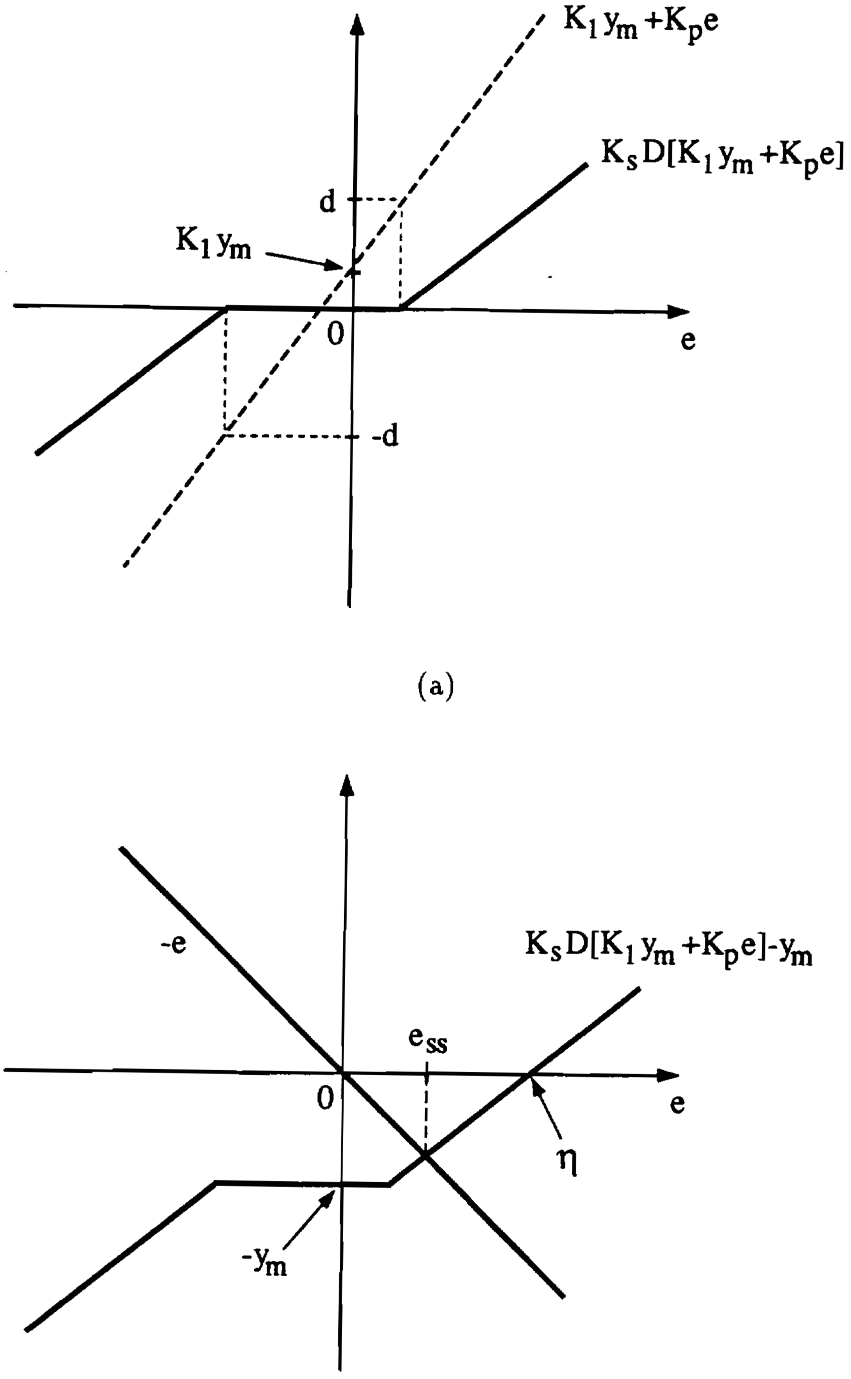
The first term in the left hand side of (4) is illustrated in Figure 3(a). Once again we use a graphical approach to solve (4); see Figure 3(b). As we can see, the solution e_{ss} is not zero, but some nonzero number (with the same sign as y_m; in Figure 3(b) we have assumed a positive y_m). It is clear that the nonzero steady-state error is a direct result of the presence of the deadzone in the actuator. In the next section we illustrate this behavior via an example.

11.3 An Example

Consider a (continuous time) plant with transfer function

$$\frac{10}{s^2 + s + 1}$$

Using the standard sample-and-hold approach, with a sampling time of 0.025 seconds, we apply the PD controller to the plant, as described before. Note that the system is of Type 0. In this example, we set $y_m = 1$, $K_1 = 0.1$, $K_P = 0.7$, and $K_D = 39.2$. Figure 4 shows output responses of the plant for three values of d: 0.0, 0.5, 1.0. In all cases we used m = 1. It is clear from Figure 4 that there is a relatively large steady-state error and overshoot when a deadzone is present. The steady-state error and overshoot increases with the the deadzone width.



(a)

(b)

Figure 3: **Graphs of: (a)** $K,D[K_1 y_m + K_p e]$; $(b) K_s D[K_1 y_m + K_p e] - y_m$ and $-e$

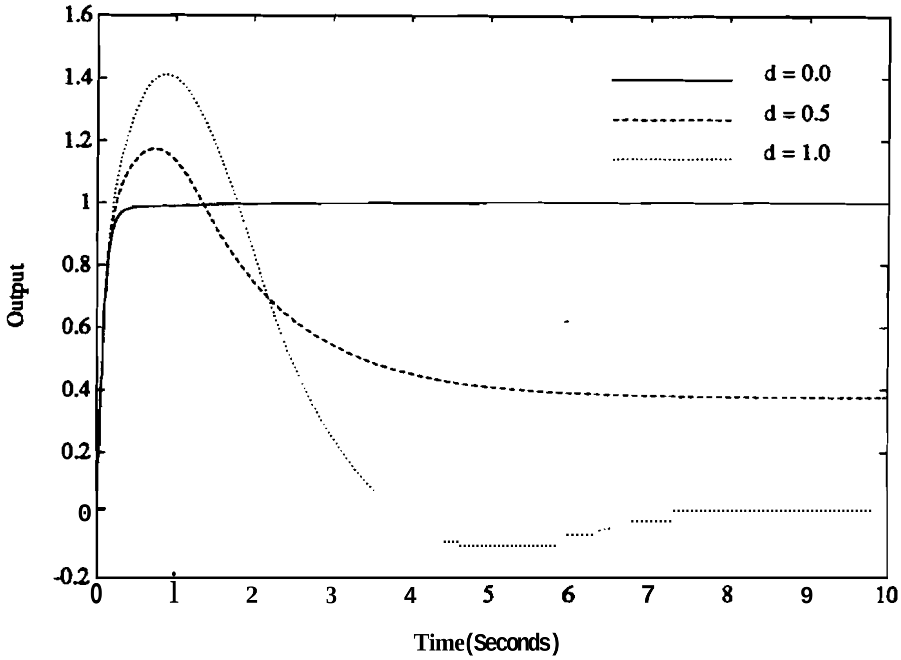


Figure 4: Output responses of plant with conventional **PD** controller

11.4 PID Controller

We may argue that a steady-state error exists in the previous system because the controller uses only the output error and change of output error. It is well known that if we also include the "integral" of the error as an input to the controller, then steady-state errors can be eliminated. In this section we study the behavior of a PID (Proportional-Integral-Derivative) controller applied to the system with a deadzone. The controller includes not only the error and change of error, but also "integral^n of error, as input.

Consider the control structure shown in Figure 5, which consists of a **PID** controller applied to the system with deadzone. The control law used is given by:

$$v(k) = v(k-1) + K_P \Delta e(k) + K_I e(k) + K_D(\Delta e(k) - \Delta e(k-1))$$

The above is the standard PID controller law, used widely in practice.

To observe the behavior of the system in Figure 5, we used the plant given in the previous example, with the following parameter values: $K_P = 1.284$, $K_I = 0.0325$,

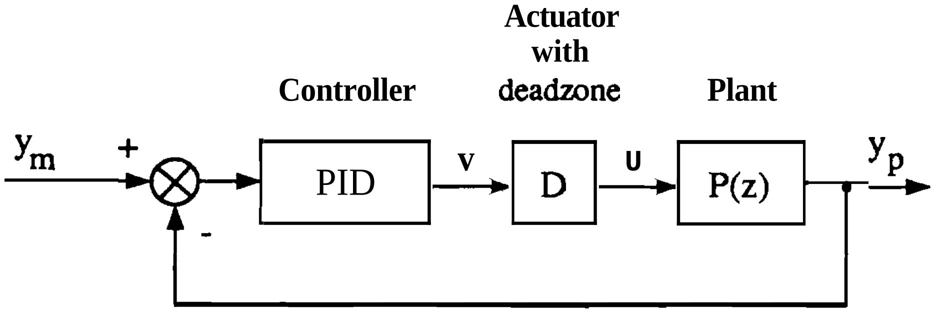


Figure 5: PID controller for system with deadzone

and $K_D = 46.8$. As before, we used a sampling time of 0.025 seconds. The output responses are shown in Figure 6. As we can see, the steady-state error is eliminated. However, the transient response is sensitive to the deadzone width, and is increasingly poor as the deadzone width is increased. By tuning the parameters of the PID controller to the specific deadzone width, we may improve the transient response (although our experience with simulations of the system suggests that the improvement is not substantial). Nonetheless, the fact remains that the PID scheme is sensitive to variations in the deadzone width, and is therefore not a practical approach to the deadzone problem.

III Controller with Fuzzy Precompensator

In this section we describe a novel controller structure based on fuzzy logic precompensation. Our aim is to eliminate the steady-state error and improve the performance of the output response for PD control systems with deadzones by introducing a fuzzy logic controller in front of the PD controller. As we shall see, our proposed scheme is indeed insensitive to deadzones, and exhibits good transient and steady-state behavior.

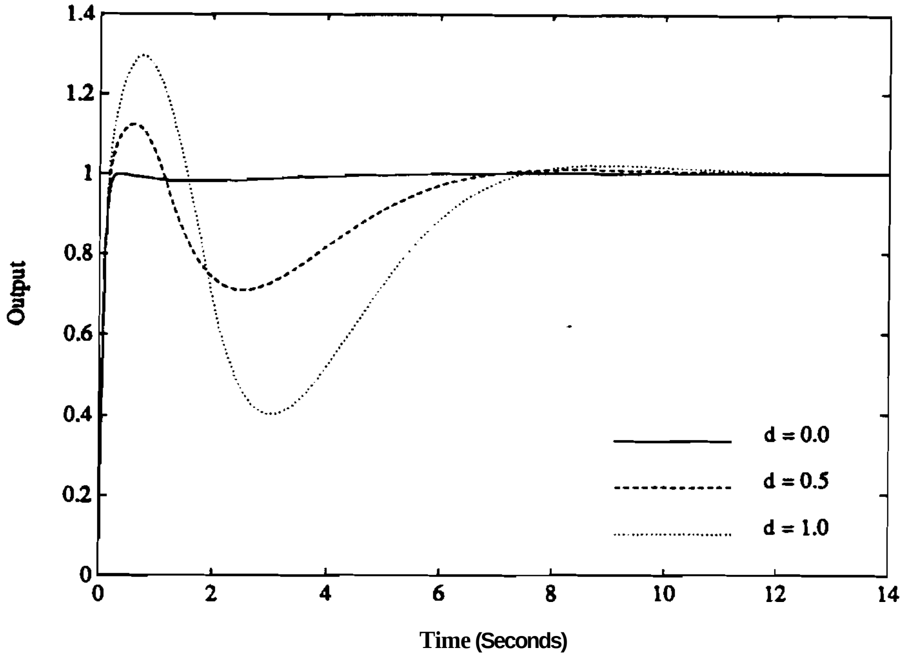


Figure 6: Output responses of plant with PID controller

III.1 Basic Control Structure

We use a graphical approach to describe the idea underlying our proposed controller. Consider Figure 3(b), which illustrates the source of the steady-state error for the conventional PD control system. Suppose we shift the graph of $K_sD[K_1y_m+K_Pe]-y_m$ to the left by an amount equal to η (the intersection point of the graph with the e-axis). Then, it is clear that the steady-state error (the point of intersection of the two graphs in Figure 3(b)) becomes zero. Shifting the graph of $K_sD[K_1y_m+K_Pe]-y_m$ to the left by an amount η is equivalent to adding η to e. In other words, the graph of $K_sD[K_1y_m+K_P(e+\eta)]-y_m$ intersects the graph of $-e$ at the origin. The key idea underlying our proposed controller is to shift the curve of $K_sD[K_1y_m+K_P(e+\eta)]-y_m$ as described above so that the steady-state error is zero. Note that instead of adding η to e to shift the curve, we can achieve a similar effect by adding some other constant μ to the reference input y_m. In our control scheme we use fuzzy logic rules to calculate the appropriate value of μ to be added to the reference input. Note that in the above scheme we have deliberately avoided using explicit knowledge of the values K_s or of the **deadzone** parameters d and m. In fact, as we shall see later, our approach is robust to variations in these parameter values.

We now proceed to describe our proposed control scheme. First, we define the variables $y'_m(k)$ and $e^t(k)$ as follows:

$$\begin{aligned} y'_m(k) &= y_m(k) + \mu(k) \\ e'(k) &= e(k) + \mu(k) \end{aligned}$$

where $\mu(k)$ is a compensating term that is generated using a fuzzy logic scheme (described below). The proposed control scheme is shown in Figure 7. As we can see, the overall control structure consists of two "layers": a fuzzy precompensator, and a conventional PD controller. The error $e(k)$ and change of error $\Delta e(k)$ are inputs to the precompensator. The output of the precompensator is $\mu(k)$. The dynamics of overall system is then described by the following equations:

$$\begin{aligned} e(k) &= y_m(k) - y_p(k) \\ \Delta e(k) &= e(k) - e(k-1) \\ \mu(k) &= F[e(k), \Delta e(k)] \\ y'_m(k) &= y_m(k) + \mu(k) \\ e'(k) &= y'_m(k) - y_p(k) \\ \Delta e'(k) &= e'(k) - e'(k-1) \\ C[e'(k), \Delta e'(k)] &= K_P e'(k) + K_D \Delta e'(k) \\ v(k) &= K_1 y'_m(k) + C[e'(k), \Delta e'(k)] \\ u(k) &= D[v(k)] \\ y_p(k) &= P(z)[u(k)] \end{aligned}$$

In the next two sections we describe in detail the two layers of our proposed controller structure.

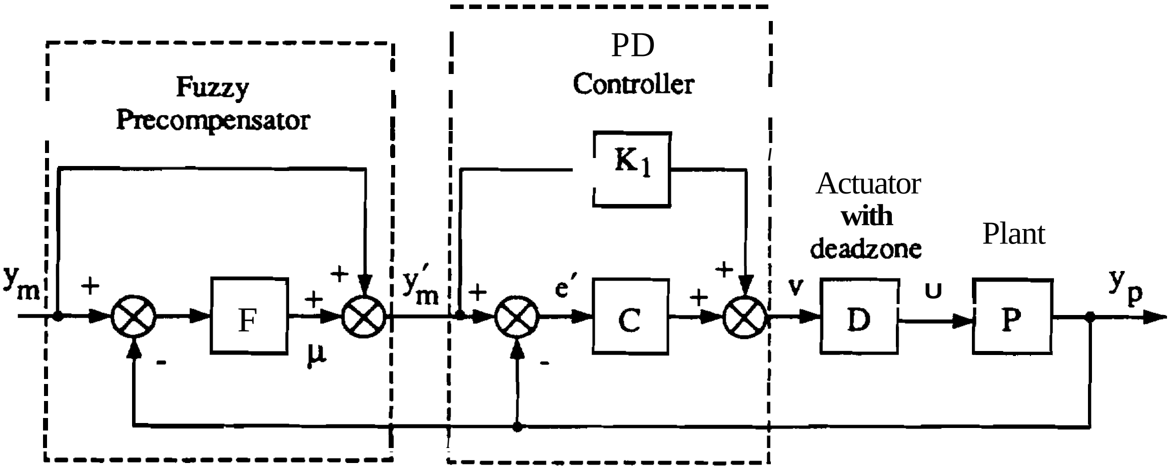


Figure 7: Proposed Control Structure

111.2 First Layer: Fuzzy Precompensator

We now describe the first layer in our two-layered controller structure, which consists of the fuzzy logic-based precompensator. The fuzzy logic control law is based on standard fuzzy logic rules—for details on fuzzy logic controllers we refer the reader to [7]. We think of $e(k)$ and $\Delta e(k)$ as inputs to the controller, and $\mu(k)$ as the output. As we already know, $e(k)$ is the output error $y_m(k)-y_p(k)$, and $\Delta e(k)$ is the change in output error $e(k)-e(k-1)$. The output $\mu(k)$ is generated via the dynamic equation

$$\mu(k) = \mu(k-1) + F[e(k), \Delta e(k)]$$

where $F[e(k), \Delta e(k)]$ is a nonlinear mapping implemented using fuzzy logic. In the following we describe how $F[e(k), \Delta e(k)]$ is implemented.

Associated with the function $F[e(k), \Delta e(k)]$ is a collection of *linguistic values*

$$L = \{NB, NM, NS, ZO, PS, \mathrm{PM}, PB)$$

and an associated collection of *membership functions*

$$M = \{M_{NB}, M_{NM}, M_{NS}, M_{ZO}, M_{PS}, M_{PM}, M_{PB}\}$$

Each membership function is a map from the real line to the interval $[0,1]$; Figure *8* shows a plot of the membership functions. The "meaning" of each linguistic value should be clear from its mnemonic; for example, N B stands for "negative-big", NM stands for "negative-medium^n, N S stands for "negative-small^n, ZO stands for "zero^n, and likewise for the "positive^n (P) linguistic value.

The realization of the function $F[e(k), \Delta e(k)]$ is based on a fuzzy logic method, consists of three stages: fuzzification, decision making fuzzy logic, and defuzzification. The process of fuzzification transforms the inputs $e(k)$ and $\Delta e(k)$ into the setting of linguistic values. Specifically, for each linguistic value $l \in$ L, we assign a pair of numbers $n_e(l)$ and $n_{\Delta e}(l)$ to the inputs $e(k)$ and $\Delta e(k)$ via the associated membership function M_l, by

$$\begin{aligned} n_e(l) &= M_l(C_e e(k)) \\ n_{\Delta e}(l) &= M_l(C_{\Delta e} \Delta e(k)) \end{aligned}$$

where C_e and $C_{\Delta e}$ are scale factors. The numbers $n_e(l)$ and $n_{\Delta e}(l)$, $l \in$ L, are used in the fuzzy logic decision process, which we describe next.

Associated with the fuzzy logic decision process is a set of fuzzy rules R $=$ $\{R_1, R_2, \ldots, R_r\}$. Each R;, i $= 1, \ldots, r$, is a triplet $(l_e, l_{\Delta e},$ I,), where $l_e, l_{\Delta e}, l_\mu \in$ L. The first two linguistic values are associated with the input variables $e(k)$ and $\Delta e(k)$, while the third linguistic value is associated with the output. An example of a rule is the triplet (NS, PS, ZO). Rules are often written in the form: "if $e(k)$ is l_e and $\Delta e(k)$ is $l_{\Delta e}$, then μ is l_μ" (here we think of μ as the output of the fuzzy logic rule). For example, in the rule represented by the triplet (NS, PS, ZO), the idea of the rule is that if $e(k)$ is "negative-small^n and $\Delta e(k)$ is "positive-small^n, then output "zero^n. The rules for our fuzzy precompensator are given in Table 1. In this case, we used 26 rules (i.e., $r = 26$). Our rules were derived by using a combination of experience, "trial and error", and our knowledge of the response of the system. These are common approaches to the design of fuzzy logic rules, as described in [7]. We refer the reader to [7] for a discussion of advantages and tradeoffs in methods for selecting fuzzy rules.

Specifically, each rule $R_i = (l_e, l_{\Delta e}, l_\mu)$ takes a given pair $e(k)$ and $\Delta e(k)$ and assigns to it a function $p_i(e(k), \Delta e(k), \mu)$, $\mu \in [-1, 1]$, as follows:

$$\begin{aligned} N_{min} &= \min(n_e(l_e), n_{\Delta e}(l_{\Delta e})) \\ p_i(e(k), \Delta e(k), \mu) &= \min(N_{min}, M_{l_\mu}(\mu)) \end{aligned}$$

We combine the functions p_i, $i = 1, \ldots, \mathbf{26}$ to get an overall function q by

$$q(e(k), \Delta e(k), \mu) = \max(p_1(e(k), \Delta e(k), \mu), \ldots, p_{26}(e(k), \Delta e(k), \mu)), \quad \mu \in [-1, 1]$$

The defuzzification process maps the result of the fuzzy logic rule stage to a real number output $F[e(k), \Delta e(k)]$. Specifically, we use the Center of Area (COA) method, given by

$$F[e(k), \Delta e(k)] = C_F \frac{\int_{-1}^{1} \mu q(e(k), \Delta e(k), \mu)\, d\mu}{\int_{-1}^{1} q(e(k), \Delta e(k), \mu)\, d\mu}$$

where C_F is a scale factor. Note that the ratio in the right hand side of the above equation is simply the center of area of the function $q(e(k), \Delta e(k), \mu)$ (as a function of μ).

Finally, as mentioned before, the actual control law for the precompensator is given by the equation:

$$\mu(k) = \mu(k-1) + F[e(k), \Delta e(k)]$$

Note that the precompensator is not simply a memoriless nonlinearity, but a nonlinear dynamical system.

111.3 Second Layer: Conventional PD Controller

The second layer of our controller structure consists of a conventional PD controller, which is essentially identical to that described in Section II.2. The only difference in this case is that instead of using $e(k)$ and $\Delta e(k)$ as inputs to the PD controller, we use e^1(k) and Ae^1(k), where e^1(k) = $e(k) + \mu(k)$, Ae^1(k) = e^1(k) – e^1(k – 1), and $\mu(k)$ is the output of the precompensator. In particular, as indicated by the dynamics equations previously, the output of the PD controller is given by

$$v(k) = K_1 y_m'(k) + C[e'(k), \Delta e'(k)]$$

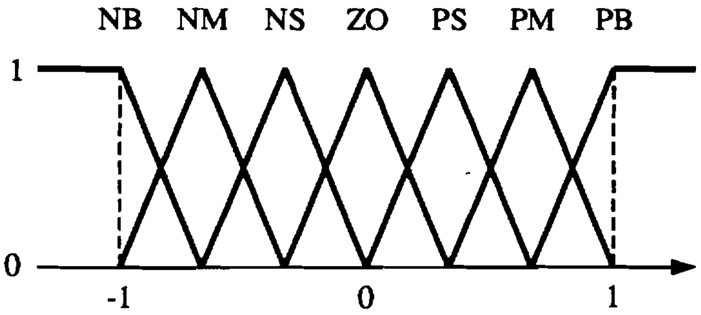


Figure 8: Membership Functions

	$e(k)$							
$\Delta e(k)$		NB	NM	NS	ZO	PS	PM	PB
	NB				NM	NS		
	NM	NB			NM	NS		
	NS	NB			NS	ZO		PM
	ZO	NB	NM	NS	ZO	PS	PM	PB
	PS	NM		ZO	PS	PM		
	PM				PM	PB	PB	
	PB			PM	PM	PB		

Table 1: Fuzzy logic rules for precompensator

where $y'_m(k) = y_m(k) + \mu(k)$.

111.4 Example

We consider again the plant of Section 11.3. We now apply the proposed two-layered fuzzy logic controller to the plant; as before we use a sampling time of 0.025 seconds. The scale factors used in the fuzzy precompensator (first layer) are as follows: $C_e = 4.5/y_m$, $C_{\Delta e} = 49.5/y_m$, $C_F = 0.2y_m$. The parameters of the PD controller (second layer) are the same as in the previous example. Here, we once again set $y_m = 1$, and $K_1 = 0.1$.

Figure 9(a) shows output responses of the plant for m = 1 and three values of d (as before): 0.0, 0.5, 1.0. The output responses in Figure 9(a) show considerable improvement over those of Figures 4 and 6. Not only is the steady-state error reduced to virtually zero, but the transient response is also dramatically improved. In Note that in Figure 9(a), the same values for the "internal variables" (e.g., scale factors, membership functions) As we can see, the performance of the controller does not deteriorate significantly for deadzone widths of d = 0.5 and 1.0. Therefore, we conclude that our controller is robust to variations in the deadzone width. In practice, we can use the same values of interval variables for a whole range of deadzone widths, without having to "retune" the controller.

Figure 9(b) shows output responses of the plant for d = 0.5 and three values of m: 2.0, 3.0, 6.0. In all three plots, the same values for the internal variables of the fuzzy precompensator were used as before. The parameter values used for the PD controller were $K_P = 0.3$ and $K_D = 9.6$. As we can see, the controller performs well in all three cases. Hence we conclude that the controller is also robust to variations in slope.

In the above examples we used $K_1 = 0.1 = K_s^{-1}$, which means that K_1 is "well-tuned" to the steady-state gain of the plant. Figures 10(a) and (b) show output responses of the plant with values of K_1 which are not well-tuned; in Figure 10(a) we used $K_1 = 2.0$ (20 times K_s^{-1}), and in Figure 10(b) we used $K_1 = 0.005$ (1/20

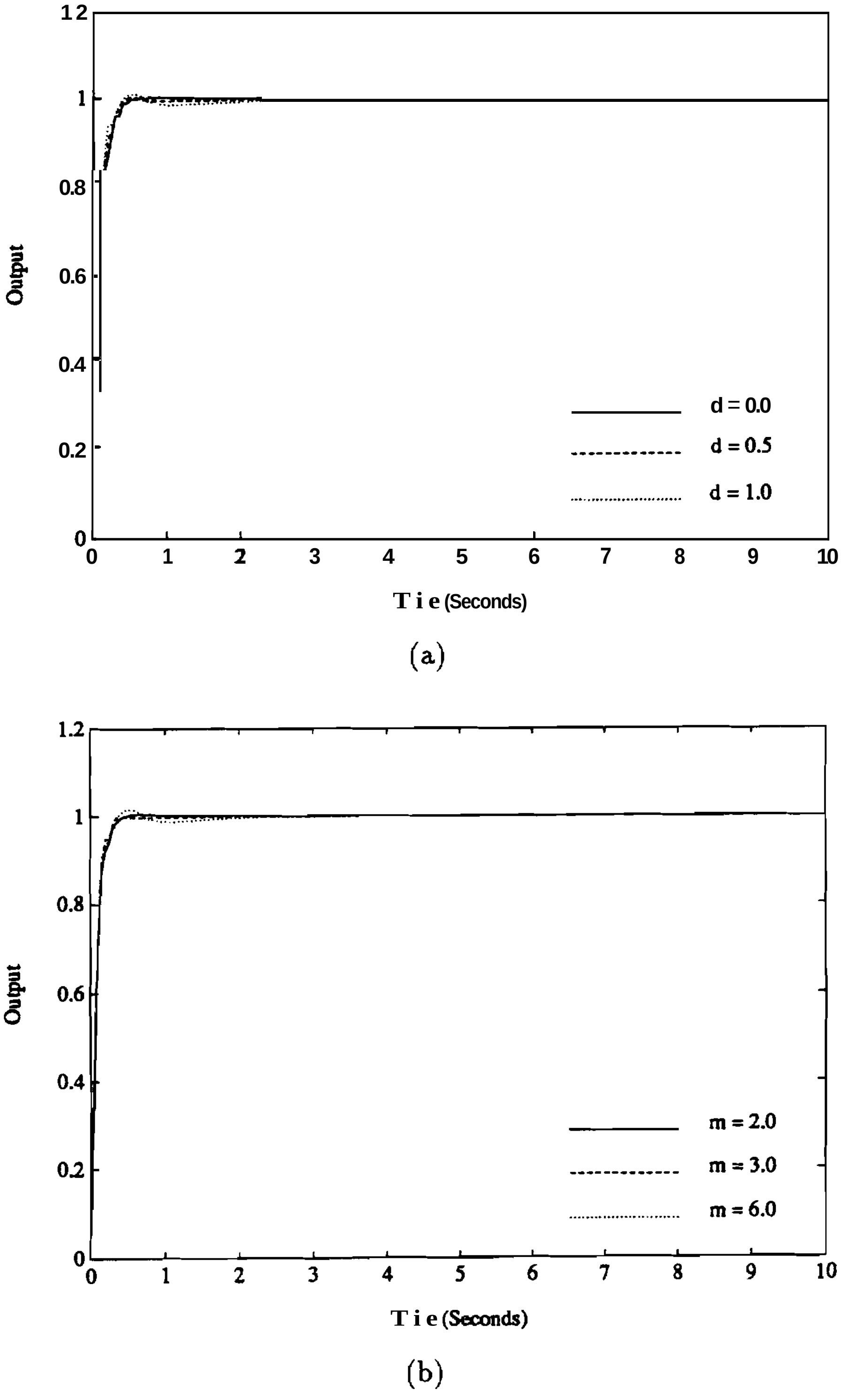


Figure 9: Output responses of plant with proposed control scheme with $K_1 = 0.1$ and **(a)** m $= 1$, and (b) d $= 0.5$

times K_s^{-1}). The parameters for the PD controller in these plots are the same as in Figure 9(a). We can see that the performance is relatively robust to the choice of K_1. Naturally, with fixed values of K_1 and the internal variables, we expect the performance to deteriorate with increasing deadzone widths, as illustrated in Figure 10. The performance for large deadzone widths may be improved if we retune the internal variables of the fuzzy precompensator.

To observe the behavior of our fuzzy precompensator with a PID controller (instead of a PD controller), we plotted the output response of the system with the fuzzy precompensation scheme and a PID controller (with no feedforward term). Note that this set up is equivalent to using a PD controller (with $K_1 = 0$) applied to a Type 1 system, namely the Type 0 system considered before with an additional pole at $z = 1$. Figure 11(a) shows output responses with deadzone slope m = 1 and deadzone widths of d = 0, 0.5, and 1.0. We used a PID controller with the same parameters as the one used in Section 11.4, Figure 6, namely, $K_P = 1.284$, $K_I = 0.0325$, and $K_D = 46.8$. We can see that the output responses for the system is virtually identical to those of Figure 9(a). In Figure 11(b), we show output responses of the system with a fixed deadzone width of d = 0.5, and deadzone slopes of m = 1, 2, and 3. The PID parameter values used in this case were $K_P = 0.39$, $K_I = 0.02$, $K_D = 22.4$. The output responses of Figure 11(b) show slight overshoots for the case where m = 1 (but not for m = 2 and 3). This indicates that the system is more sensitive to variations in deadzone slope than to variations in deadzone width. Comparing Figure 11(b) with Figure 9(b), we see that the precompensator with a PID controller is more sensitive (with respect to slope variations) than the precompensator with a PD controller.

IV Conclusions

In this report, we proposed a fuzzy logic-based precompensation scheme for controlling systems with deadzones. Our approach consists of a fuzzy precompensator and a conventional PD controller. The proposed control scheme has superior steady-state

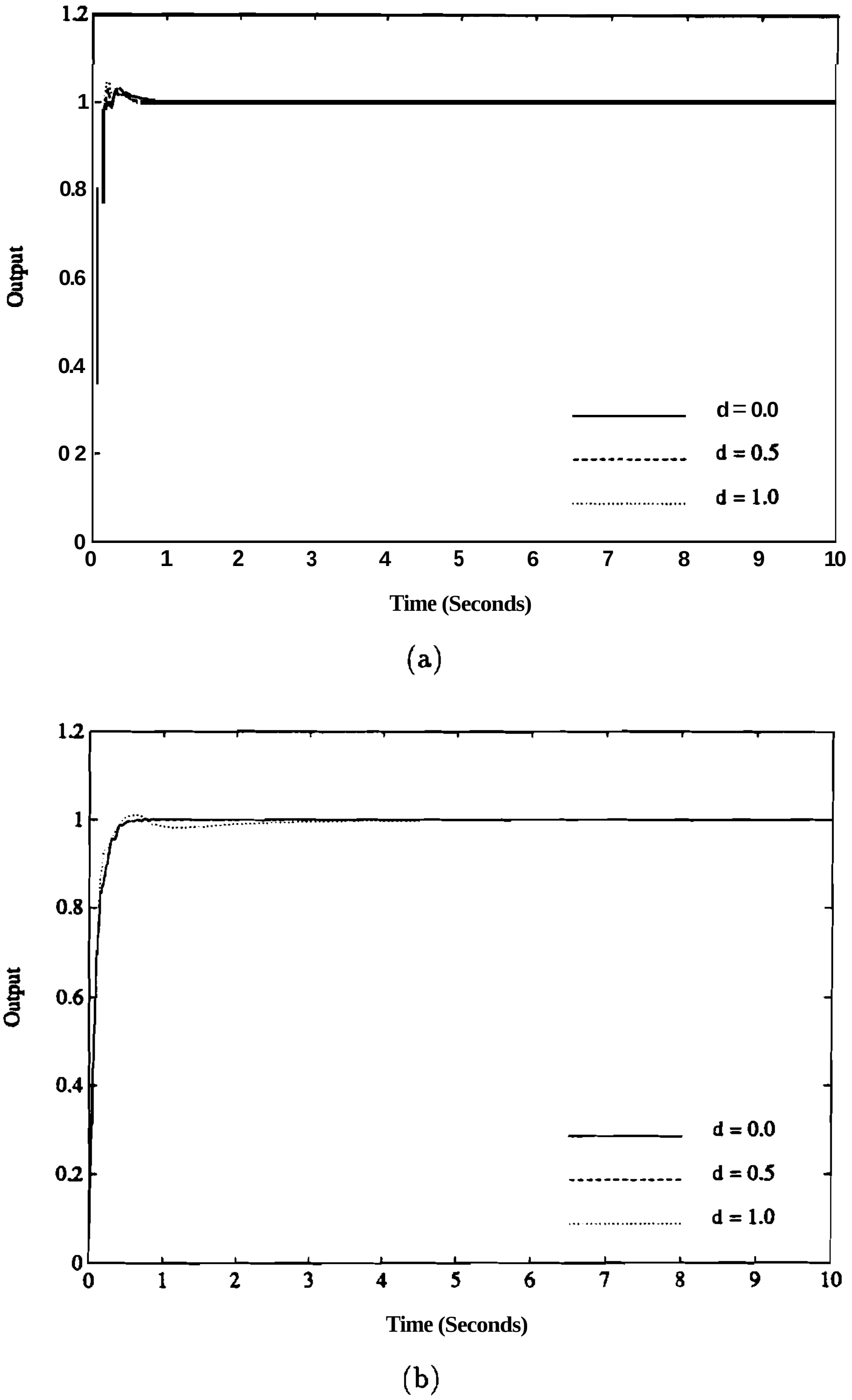
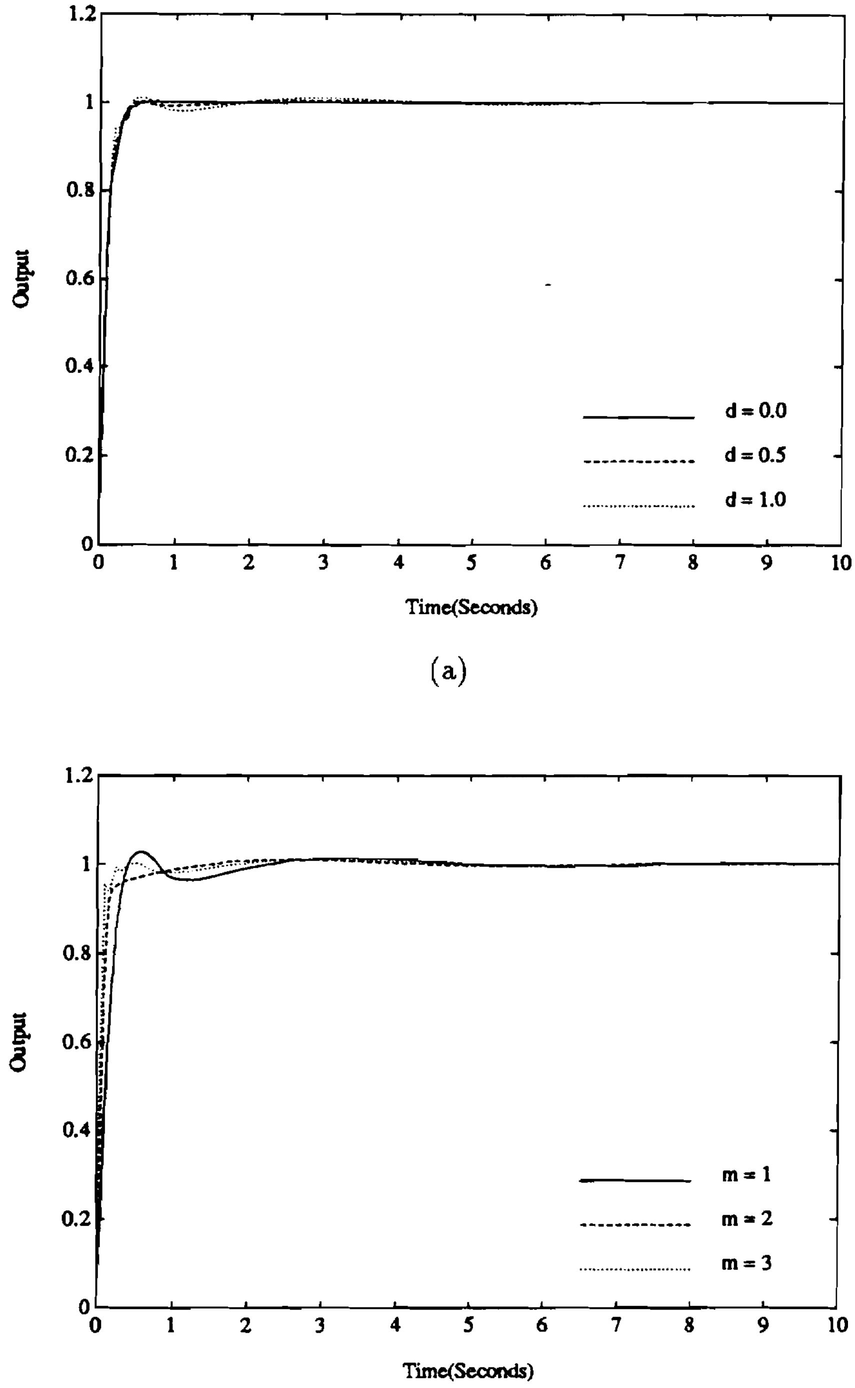


Figure 10: Output responses of plant with proposed control scheme with (a) $K_1 = 2.0$, and (b) $K_1 = \mathbf{0.005}$



(a)

(b)

Figure 11: Output responses of plant with fuzzy precompensator and PID controller with (a) $m = 1$, and (b) $d = 0.5$

and transient performance, compared to a conventional PD controller, as well as a PID controller. An advantage of our present approach is that an existing PD controller can be easily modified into our control structure by simply adding a fuzzy precompensator. In addition, the control structure is robust to variations in the deadzone nonlinearities (width and slope), as well as the steady-state gain of the plant. We demonstrated the performance of our controller via several computer simulation examples.

In this report, we do not address the important problem of stability of the control scheme. As for many other fuzzy logic based control schemes, a mathematical analysis of the stability of our scheme is an intractable problem, due to the highly nonlinear nature of the fuzzy precompensator. This difficult but important problem is a topic of ongoing research.

References

[1] V. I. Utkin, *Sliding Modes and Their Application in Variable Structure Systems.* Moscow:Mir, 1978.

[2] C. A. Desoer and S. M. Shahruz, "Stability of dithered nonlinear systems with backlash or hysteresis," *Int. J. Control,* vol. 43, no. 4, pp. 1045–1060, 1986.

[3] D. A. Recker, P. V. Kokotovic, D. Rhode and J. Winkelman, "Adaptive nonlinear control of systems containing a dead-zone," in *Proc. of the IEEE Conf. on Dec. and Contr.*, pp. 2111–2115, Brighton, U.K., Dec. 1991.

[4] G. Tao and P. V. Kokotovic,"Adaptive control of Plants with unknown dead-zones," Report No. CCEC-91-1006, Univ. of Cal., Santa Barbara, U.S.A., Sept. 1991.

[5] E. H. Mamdani and B. R. Gaines, "Fuzzy *reasoning and its Applications"* London:Academic. 1981.

[6] Y. F. Li and C. C. Lau, "Development of fuzzy algorithms for servo systems," IEEE Contr. Syst. Magazine, vol. 9, pp. 65–72, 1989.

[7] C. C. Lee, "Fuzzy Logic in Control Systems: Fuzzy Logic Controller — Part I, Part II," IEEE Trans. on Sys. Man and Cyber., vol. 20, no. 2, pp. 404–435,1990.

[8] D. P. Kwok, P. Tam, C. K. Li and P. Wang, "Linguistic PID controllers," in Proc. of IFAC 1lth *Triennial* World Congress, Tallinn, USSR, vol. 4, pp. 205–210, Aug. 1990.

[9] B. Kosko, Neural Networks and Fuzzy Systems, Englewood Cliffs, New Jersey: Prentice Hall, 1992.

[10] J.-H. Kim, J.-H. Park, S.-W. Lee, E. K. P. Chong, "A Novel Approach to Fuzzy Logic Controller Design for Systems With Deadzones," Tech. Report TR-EE 92-44, School of Electrical Engineering, PurdueUniversity, October 1992.