Proceedings of the American Control Conference
Chicago, Illinois • June 2000

Self-Tuning of PID Controllers by Adaptive Interaction. [1]

Feng Lin, Robert D. Brandt, and George Saikalis

Abstract

We propose a new self-tuning or adaptation algorithm for PID controllers based on a theory of adaptive interaction. The theory develops a simple and effective way to perform gradient descent in the parameter space. One version of the tuning algorithm requires no knowledge of the plant to be controlled. This makes the algorithm robust to changes in the plant. It also makes the algorithm universally applicable to linear and nonlinear plants. The algorithm achieves the tuning objective by minimizing an error function. Because of its simplicity, the overhead for adding self-tuning is negligible. We applied this algorithm in an automotive product manufactured by Hitachi to satisfy performance requirements for both cold start and normal operation. Simulation results are presented to show the validity of the approach.

Keywords: PID controller, adaptive control, self-tuning

1 Introduction

Although control theory has made great advance in the last few decades, which has led to many sophisticated control schemes, PID control still remains the most popular type of control being used in industries today. This popularity is partly due to the fact that PID controllers have simple structures and very well understood principles. Furthermore, a well-tuned PID controller can have excellent performance. Here, the words "well-tuned" must be emphasized because the performance of a PID controller is crucially dependent on the tuning process.

For the convenience of discussion, we would like to classify PID tuning into two (perhaps overlapping) classes: (1) initial "off-line" tuning and (2) continuous "on-line" self-tuning. Since our approach is more likely to be used for the second class, we will focus our discussions on this class. We believe that there are at least two reasons for having an on-line self-tuning. First, the objectives and hence requirements of a PID controller often changes during the different stages of control. This is quite evident from our experience with automotive control systems. The control objectives during the "cold start" is often different from the"normal operation". We believe that this is also true for many general systems. For example, we often want a system to have a fast response time initially, but then put more emphasis on reducing steady-state error. Fast response time and small steady-state error are often conflicting objectives and require different set of parameters for the PID controller. Therefore, PID control can be improved greatly if we will set the parameters initially to ensure fast response time and then tune the parameters to reduce steady-state error. We indeed applied this self-tuning PID controller in an automotive product manufactured by Hitachi. The result is a much improved controller. The second reason for on-line self-tuning is that the plant to be controlled often changes from time to time. This is especially true if the plant is a nonlinear system with changing operation points. When this is the case, our approach provides a simple and effective method for adaptation of such changes.

Our approach is based on a recently developed theory of adaptive interaction [5]-[8]. Using this theory, the controlled system is decomposed into four subsystems consisting the plant, the proportional, integral and derivative control. The parameters of the PID control, K_P, K_I, and K_D are viewed as the interactions between these four subsystems. A general adaptation algorithm developed in the theory of adaptive interaction is applied to self-tuning these coefficients. The algorithm is simple and effective.

To apply this self-tuning algorithm, the only information needed about the plant is its Fréchet derivative. For linear systems, the Fréchet derivatives can be easily calculated. Furthermore, simulation results show that in many cases, the Fréchet derivative can be replaced by a constant that is then absorbed into the adaptation coefficient. Using this approximated self-tuning algorithm, we can eliminate any dependence on the plant model and hence make the algorithm universal to a large class of systems.

Another way to investigate our self-tuning algorithm is to to view the self-tuning PID controller as a nonlinear controller because the parameters K_P, K_I, and K_D are changing continuous according to the adaptation dynamics. In general, we do not require that K_P, K_I, and K_D convergent to some constants. In fact, we will let them change as the inputs or disturbances change. Because of this property, our self-tuning PID controllers can do more than conventional PID controllers. For example, they can stabilize systems than cannot be stabilized by conventional PID controllers.

We investigated the effectiveness of our self-tuning PID controllers by simulation for a large class of systems, including linear and nonlinear plants, stable and unstable plants, and plants with delays. We also simulated sys-

[1]This research is supported in part by the National Science Foundation under grant ECS-9315344. Feng Lin is with the Department of Electrical and Computer Engineering, Wayne State University, Detroit, MI 48202; Tel: 313-5773428, Fax: 313-5771101, Email: flin@ece.eng.wayne.edu. Robert D. Brandt is with Intelligent Devices, Inc., 465 Whittier Avenue, Glen Ellyn, IL 60137. George Saikalis is with Hitachi, 34500 Grand River Ave., Farmington Hills, MI 48335.

tems with noise and saturation. In the simulations, we used both the Fréchet self-tuning algorithm and the approximated self-tuning algorithm. In all these simulations, we have not found any case where the closed-loop systems is unstable. We encourage the readers to try for themselves.

This introduction will not be completed without giving references to other approaches to PID tuning. They are listed at the end of the paper. To compare our approache to other approaches is however a difficult task for two reasons: First, there are just too many approaches to PID tuning; and secondly, our approach is quite different from conventional approaches. Therefore, any omission in this regard is not intentional.

This paper is organized as follows. In Section 2, we will briefly present the theory of adaptive interaction that forms the basis of our approach. In Section 3, we derive the Fréchet self-tuning algorithm and the approximated self-tuning algorithm. We also prove the stability of the closed-loop system with the self-tuning PID controller. In Section 4, we present some simulation results for various types of systems.

2 Theory of Adaptive Interaction

The theory of adaptive interaction considers a complex system consisting of N subsystems which we called devices. Each device (indexed by $n \in \mathcal{N} := \{1, 2, ..., N\}$) has an integrable output signal y_n and an integrable input signal x_n. The dynamics of each device is described by a (generally nonlinear) causal[1] functional

$$\mathcal{F}_n : \mathcal{X}_n \rightarrow \mathcal{Y}_n, \quad n \in \mathcal{N},$$

where $\mathcal{X}_n$ and $\mathcal{Y}_n$ are the input and output spaces respectively. That is, the output $y_n(t)$ of the nth device relates to its input $x_n(t)$ by

$$y_n(t) = (\mathcal{F}_n \circ x_n)(t) = \mathcal{F}_n[x_n(t)], \quad n \in \mathcal{N},$$

where $\circ$ denotes composition.

We assume the Fréchet derivative of $\mathcal{F}_n$ exists[2]. We further assume that each device is a single-input single-output system[3].

An interaction between two devices consists of a (generally non-exclusive) functional dependence of the input of one of the devices on the outputs of the others and is mediated by an information carrying connections denoted by c. The set of all connections is denoted by $\mathcal{C}$.

[1]A functional $\mathcal{F}_n : \mathcal{X}_n \rightarrow \mathcal{Y}_n$ is causal if $y_n(t)$ depends only on the previous history of x_n, $\{x_n(\tau) : \tau \leq t\}$.

[2]the Fréchet derivative [15], $\mathcal{F}'_n[x]$, of $\mathcal{F}_n[x]$, is defined as a functional such that

$$lim_{||\Delta|| \rightarrow 0} \frac{||\mathcal{F}_n[x+\Delta] - \mathcal{F}_n[x] - \mathcal{F}'_n[x] \circ \Delta||}{||\Delta||} = 0.$$

[3]The assumption of single-input single-output is not as restrictive as it may seem. This is because the partition of system into devices is arbitrary and up to the designer. Therefore, one can often partition a multi-input multi-output system into several single-input single-output systems.

We assume that there is at most one connection from one device to another. Let pre_c be the device whose output is conveyed by connection c and $post_c$ the device whose input depends on the signal conveyed by connection c. We denote the set of input interactions for the nth device by $I_n = \{c : pre_c = n\}$ and the set of output interactions by $O_n = \{c : post_c = n\}$. A typical system is illustrated in Figure 1. In the figure, for example, the set of input interactions of Device 2 is $I_2 = \{c_1, c_3\}$ and the set of output interactions is $O_2 = \{c_4\}$. Also, c_1 connects Device 1 to Device 2, therefore $pre_{c_1} = 1, post_{c_1} = 2$.

For the purpose of this paper, we consider only linear interactions, that is, we assume that the input to a device is a linear combination of the output of other devices via connections in I_n and possibly an external input signal $u_n(t)$:

$$x_n(t) = u_n(t) + \sum_{c \in I_n} \alpha_c y_{pre_c}(t), \quad n \in \mathcal{N},$$

where α_c is the connection weights.

With this linear interaction, the dynamics of the system is described by

$$y_n(t) = \mathcal{F}_n[u_n(t) + \sum_{c \in I_n} \alpha_c y_{pre_c}(t)], \quad n \in \mathcal{N}.$$

To simplify the notation, in the rest of the paper, we will eliminate when appropriate the explicit reference to time t.

The goal of our adaptation algorithm is to adapt the connection weights α_c so that some performance index $E(y_1, ..., y_n, u_1, ..., u_n)$ as a function of the external inputs and outputs will be minimized. The algorithm is given in the following theorem [8].

Theorem 1 *For the system with dynamics given by*

$$y_n = \mathcal{F}_n[u_n + \sum_{c \in I_n} \alpha_c y_{pre_c}], \quad n \in \mathcal{N},$$

if connection weights α_c are adapted according to

$$\dot{\alpha}_c = \Big(\sum_{s \in O_{post_c}} \alpha_s \dot{\alpha}_s \frac{\frac{dE}{dy_{post_s}} \circ \mathcal{F}'_{post_s}[x_{post_s}]}{\frac{dE}{dy_{post_s}} \circ \mathcal{F}'_{post_s}[x_{post_s}] \circ y_{post_c}} - \gamma \frac{\partial E}{\partial y_{post_c}} \Big) \circ \mathcal{F}'_{post_c}[x_{post_c}] \circ y_{pre_c}, \quad c \in \mathcal{C},$$

and the above equation has a unique solution, then the performance index E will decrease monotonically with time. In fact, the following is always satisfied

$$\dot{\alpha}_c = -\gamma \frac{dE}{d\alpha_c}, \quad c \in \mathcal{C},$$

where $\gamma > 0$ is some adaptation coefficient.

The above theorem can be applied to a very general class of systems. For example, its application to neural networks was reported in [5]-[8]. Using this algorithm, a neural network can adapt without the need of a feedback network to back-propagate errors. The algorithm hence provides a biologically plausible mechanism for adaptation in biological neurons.

Since the PID control system is special case of the systems amendable to the above adaptation algorithm, the algorithm can be significantly simplified as shown in the next section.

3 Tuning Algorithm

For a PID control system, we decompose the system into four devices as shown in Figure 2: Device 1 is the proportional part with transfer function 1; Device 2 is the integral part with transfer function s^{-1}; Device 3 is the derivative part with transfer function s; and Device 4 is the plant. In some implementations, the differentiation and integration are often modified to improve the performances. For example, differentiation is often preceded by a low-pass filter. As we shall see, our algorithm applies equally well to such modifications. In any case, there are three adaptive connections: $\alpha_c = K_P, K_I$, or K_D. Since for all these connections, $O_{post_c} = O_4 = \emptyset$, the adaptation algorithm of the previous section reduces to

$$\dot{\alpha}_c = -\gamma \frac{\partial E}{\partial y_{post_c}} \circ \mathcal{F}'_{post_c}[x_{post_c}] \circ y_{pre_c}.$$

We take our goal as to minimize the error[4]

$$E = e^2 = (u - y_4)^2.$$

We then obtain the following Fréchet tuning algorithm

$$\begin{aligned} \dot{K}_P &= -2\gamma(y_4 - u)\mathcal{F}'_4[x_4] \circ y_1 \\ &= -2\gamma e \mathcal{F}'_4[x_4] \circ y_1. \end{aligned}$$

Similarly, we have

$$\begin{aligned} \dot{K}_I &= -2\gamma e \mathcal{F}'_4[x_4] \circ y_2 \\ \dot{K}_D &= -2\gamma e \mathcal{F}'_4[x_4] \circ y_3. \end{aligned}$$

Note that the self-tuning algorithm for P, I and D all have the same form: It depends on the error e, the Fréchet derivative $\mathcal{F}'_4[x_4]$, and the output of the device $y_i, i = 1, 2, 3$. It is independent of the natural of the device, whether it is P, I, D, or anything else. Therefore, any modification of differentiation or integration will not change the adaptation algorithm.

To calculate the Fréchet derivative, let us consider the functional $y = \mathcal{F}[x]$ of the following form

$$\mathcal{F}[x] = \int_0^t f(x(\tau), \tau) d\tau.$$

[4]Other performance index can also be used, resulting in different self-tuning algorithms.

It can be shown ([15], page 175) that the Fréchet differential of $\mathcal{F}$ is equal to its Gateaux differential which is given by

$$\delta\mathcal{F}(x; h) = \int_0^t f_x(x(\tau), \tau) h(\tau) d\tau,$$

where $f_x = \frac{\partial f}{\partial x}$. Therefore, the Fréchet derivative of $\mathcal{F}$ at x is given by

$$\mathcal{F}'[x] \circ h = \int_0^t f_x(x(\tau), \tau) h(\tau) d\tau.$$

For a linear time-invariant plant with transfer function $G(s)$, $\mathcal{F}$ is given by the convolution

$$\mathcal{F}[x] = g(t) * x(t) = \int_0^t x(\tau) g(t - \tau) d\tau,$$

where $g(t)$ is the impulse response. Therefore the Fréchet derivative

$$\mathcal{F}'[x] \circ h = \int_0^t g(t - \tau) h(\tau) d\tau = g(t) * h(t).$$

By simulation, we found that for many practical systems the Fréchet derivative can be approximated by

$$\mathcal{F}'[x] \circ h = \beta h,$$

where β is some constant.

Substitute the above approximation into the Fréchet tuning algorithm, we obtain the following modified tuning algorithm.

$$\begin{aligned} \dot{K}_P &= -\gamma e y_1 \\ \dot{K}_I &= -\gamma e y_2 \\ \dot{K}_D &= -\gamma e y_3. \end{aligned}$$

Here we have absorbed 2β into the adaptation coefficient γ. This modified algorithm can be implemented as shown in Figure 3.

4 Simulation Results

We performed various simulations which show excellent properties of our tuning algorithm. All the simulations are performed using SIMULINK.

Stable Plant

we started our simulation with a linear stable plant having the following transfer function

$$G_1(s) = \frac{5000}{(s+1)(s+5)(s+100)}.$$

we first applied the modified tuning algorithm, which worked very well. Figure 4 shows the results of a simulation with the input u being a square wave of frequency $\omega = 1$ and magnitude=1. The adaptation coefficient

$\gamma = 0.03$. As shown in the figure, after tuning, the rise time and overshoot are excellent.

For comparison, we also simulated the system using the Fréchet tuning algorithm. We found that the results are very similar. In other words, the modified algorithm worked very well.

In the simulation, we randomly picked the initial PID gains K_P, K_I, K_D. In fact, for the values we picked, the closed-loop system is not stable initially. However, since our algorithm converges fast, it stabilizes the system very shortly as shown in Figure 4. This is significant because it may be difficult to determine an initially stable PID gains if the systems is complex and/or nonlinear.

Adaptation under noise

In order to study the effectiveness of our tuning algorithm under noisy environment, we superimposed a white noise signal with power=0.01 to the input x_4 of the plant. Simulation results show that noise has only small effect on the tuning process.

Unstable Plant

We also simulated a plant with the following transfer function

$$G_2(s) = \frac{5000}{s(s+5)(s+100)}.$$

This plant is open-loop unstable because of the pole at the origin. For unstable plants, we cannot use the Fréchet tuning algorithm, because the Fréchet derivative $g(t) * h(t)$ is unstable. However, using the modified tuning algorithm, we still obtained excellent results.

Systems with Time Delay

Our tuning algorithm also applies to systems with time delay. By adding a delay of 0.2 second before the plant with transfer function $G_1(s)$, we obtained simulation results shown in Figure 5, where the input is a sine wave of frequency $\omega = 1$ and magnitude=1. The adaptation coefficient $\gamma = 0.3$. We used the modified tuning algorithm in our simulation. As shown in the figure, the error decreases as tuning taking place.

Nonlinear Systems

If the plant is nonlinear, then it may be difficult to find its Fréchet derivative. However, our modified tuning algorithm can still be used. In Figure 6, we shown simulation results of a nonlinear plant with the following dynamics.

$$\dot{y}_4 = -5y_4 + y_4^2 + x_4.$$

Again the input is a sine wave of frequency $\omega = 1$ and magnitude=1. The adaptation coefficient $\gamma = 10$.

5 Conclusion

The PID tuning algorithm proposed in this paper has many advantages in applications, most notably, its simplicity and independence of the plant model. The simulation results shows that it performs very well under various situations: linear or nonlinear plants; with or without noise; stable or unstable plants; and with or without time delay. In all these cases, the tuning mechanism remains unchanged, a further proof of this applicability. Because of the proprietary nature of the information, we cannot report the application of this approach to the Hitachi product at this time. However, we hope to do this in the near future.

References

[1] K.J. Astrom, T. Hagglund, 1984. Automatic tuning of simple regulators with specifications on phase and amplitude margins. *Automatica, 20*, 645-651.

[2] K.J. Astrom, T. Hagglund, 1988. *Automatic Tuning of PID Controllers*, ISA.

[3] K.J. Astrom, C.C.Hang, P.Persson and W.K. Ho, 1992. Towards intelligent PID control. *Automatica, 28*, 1-9.

[4] K.J. Astrom, T. Hagglund,C.C.Hang and W.K. Ho, 1993. Automatic tuning and adaptation for PID controllers - survey. *Control Engineering Practice, 1*, 699-714.

[5] R. D. Brandt and F. Lin, 1994. Supervised learning in neural networks without explicit error backpropagation. *Proceedings of the 32nd Annual Allerton Conference on Communication, Control and Computing*, pp. 294-303.

[6] R. D. Brandt and F. Lin, 1996. Can supervised learning be achieved without explicit error backpropagation? *Proceedings of the International Conference on Neural Networks*, pp. 300-305.

[7] R. D. Brandt and F. Lin, 1996. Supervised learning in neural networks without feedback network. *1996 IEEE International Symposium on Intelligent Control*, pp. 86-90.

[8] R. D. Brandt and F. Lin, 1998. *Theory of Adaptive Interaction*, AFI Press, to appear.

[9] D.W. Clark, 1986. Automatic tuning of PID regulators. In *Expert Systems and Optimization in Process Control*, Technical Press, 85-104.

[10] C.C. Hang, K.J. Astrom and W.K. Ho, 1993. Relay auto-tuning in the presence of static load disturbance. *Automatica, 29*, 563-564.

[11] W.K. Ho, C.C. Hang and L.S. Cao, 1995. Tuning of PID controllers based on gain and phase margin specification. *Automatica, 31*, 497-502.

[12] J. Lieslehto, J.T. Tanttu and H. N. Koivo, 1988. An expert systems for tuning PID controllers. *Proc. American Control Conference*, 261-262.

[13] J. Litt, 1991. An expert system to perform on-line controller tuning. *IEEE Control Systems Magazine, 11*, 18-23.

[14] L. Loron, 1997. Tuning of PID controllers by the non-symmetrical optimum method. *Automatica, 33*, 103-107.

[15] D. G. Luenberger,1968. *Optimization by Vector Space Methods*, John Wiley & Sons.

[16] Y. Nishikawa, N. Sannomiya, T. Ohta and H. Tanaka, 1984. A method for auto-tuning of PID control parameters *Automatica, 20*, 321-332.

[17] Z. Shafiet and A.T. Shenton, 1994. Tuning of PID-type controllers for stable and unstable systems with time delay. *Automatica, 30*, 1609-1615.

[18] Y. Tang and R. Ortega, 1993. Adaptive tuning to frequency response specification. *Automatica, 29*, 1557-1563.

[19] A.A. Voda and I.D. Landau, 1995. A method for the auto-calibration of PID controllers. *Automatica, 31*, 41-53.

[20] J.G. Ziegler and N.B. Nichols, 1942. Optimal setting for automatic controllers. *ASME Trans., 64* 759-768.

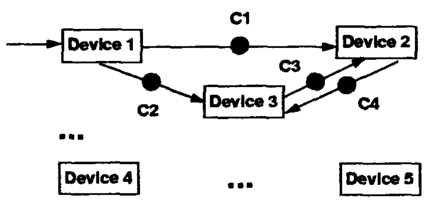

Figure 1: Devices and interactions

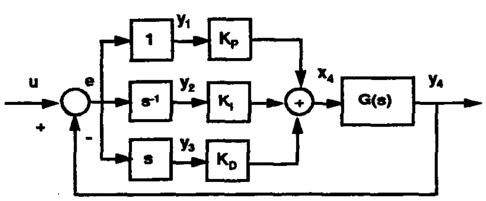

Figure 2: PID controller

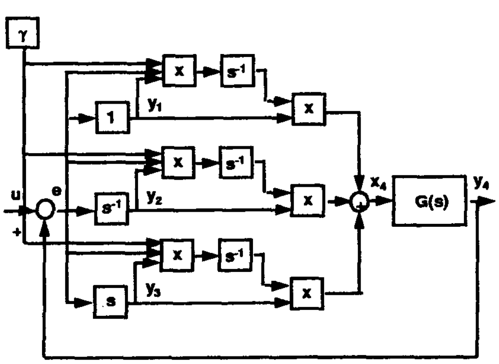

Figure 3: PID Self-tuning

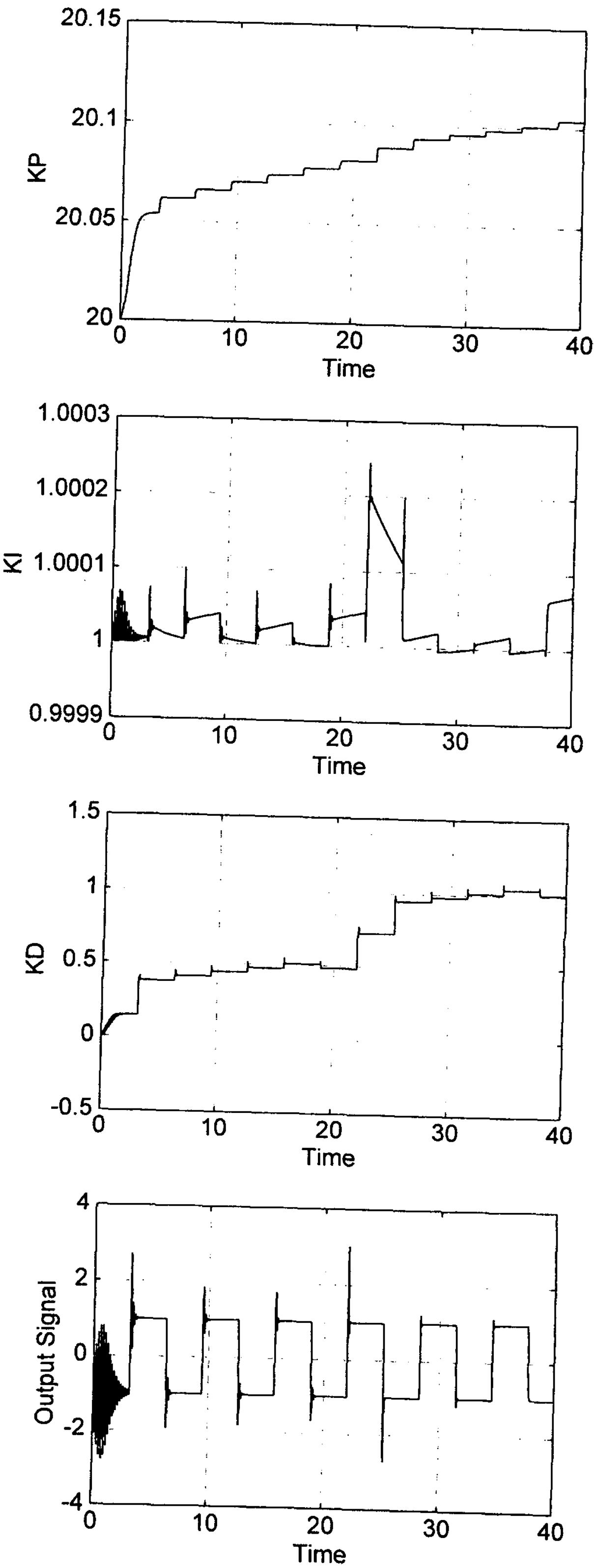

Figure 4: Stable plant

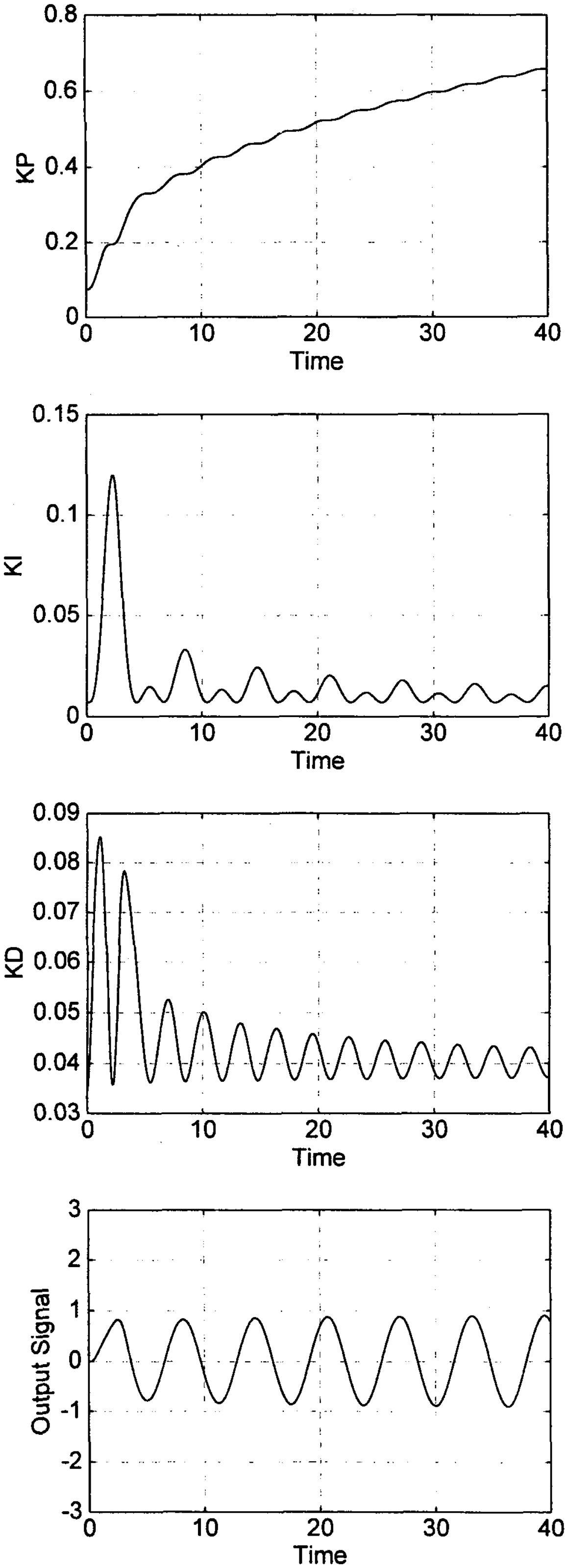

Figure 5: System with time delay

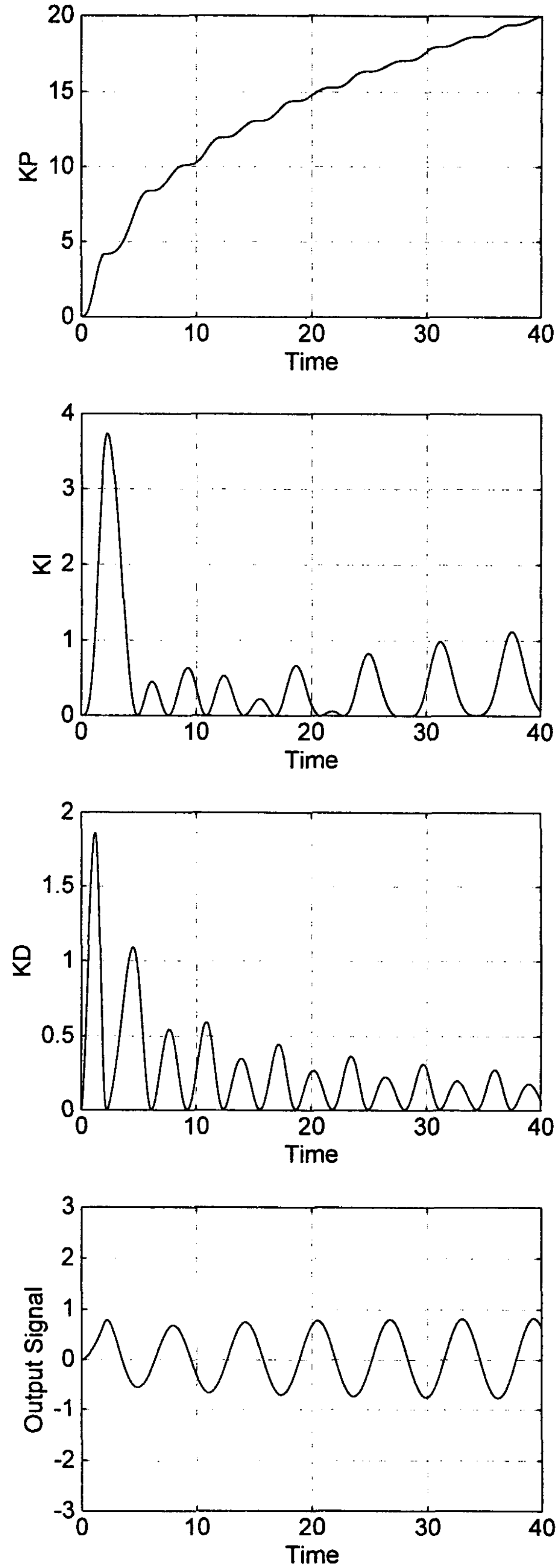

Figure 6: Nonlinear systems