Pergamon

PII: S0005-1098(98)00060-0

Automatica, Vol. 34, No. 9, pp. 1125–1133, 1998

Printed in Great Britain
0005-1098/98 \$—see front matter

Brief Paper

Control Curve Design for Nonlinear (or Fuzzy) Proportional Actions Using Spline-based Functions*

B.-G. HU,†‡ G. K. I. MANN‡§ and R. G. GOSINE‡§

Key Words—Nonlinear PID control; fuzzy PID control; fuzzy proportional actions; control curves; spline functions.

Abstract—This work explores a novel approach for a systematic design of a nonlinear mapping system typically for nonlinear, or fuzzy, PID control applications. The paper investigates a nonlinear design of proportional actions using spline-based functions. Specifically, the controller uses Bézier curves to form the nonlinear mapping in order to emulate fuzzy PID systems. While other researchers have addressed the fuzzy systems for approximations of given functions, we believe that, in general control problems, these approximations should be considered in dealing with the properties of unknown control actions. Proportional action is selected as a basic function for nonlinear *control curve* designs. The reasons for this selection are discussed. Specific *heuristic properties* for the proportional action are defined based on the intuitions in general PID controller applications. The new controller is designed to be compatible with these properties. The nonlinearity variation index is used as a process-independent measure for evaluation of different designs. The system has been shown to improve the conventional fuzzy PID controllers on three aspects. These include a high degree of *transparency* with respect to nonlinear tuning parameters, *versatility* to cover various nonlinear functions, and *simplicity* of nonlinear mapping expressions. © 1998 Elsevier Science Ltd. All rights reserved.

1. Introduction

One of the important features of fuzzy PID control systems is their accommodation of the nonlinear control to the process (Ying, 1993; Zhao *et al.*, 1993; He *et al.*, 1993; Lewis and Liu, 1996; Tan *et al.*, 1997). With nonlinear control actions, the process is regulated according to the error signal to produce better performance than the conventional, or *linear*, PID controllers. The design of a fuzzy PID controller is actually a task of building an inference engine with a nonlinear relationship between crisp input and output using linguistic expressions. This task is related to two basic issues. *First, what kind of nonlinearity of the controller output does a process require with respect to the specific performance criteria? Second, how does one construct a system which produces the desired or approximate nonlinear functions?* To the authors' knowledge, there do not seem to be any analytical findings in an explicit form to solve the first issue. In a previous investigation (Hu *et al.*, 1997), a three-rule fuzzy PID controller was investigated. The single-input–single-output (SISO) fuzzy inference was used to produce nonlinear proportional actions (or curves) with respect to the error. Four types of nonlinear curves, defined within the positive range of the error signal, could be constructed by the controllers (Fig. 1). In the numerical simulations, the two simplest nonlinear, C-type, curves (Fig. 1a and b) were selected for the fuzzy proportional action by using a genetic optimization solver. These C-type curves, approximating desired but unknown controller output for a specific performance, have produced satisfactory results for several processes including first and second-order plants.

Significant studies have been reported in regard to the second issue of using fuzzy systems (Wang and Mendel, 1992; Buckley, 1992; Kosko, 1994; Ying, 1994). Stimulated by the success of approximation using neural networks (Funahashi, 1989; Hornik *et al.*, 1989; Cotter, 1990), Wang and Mendel (1992), Buckley (1992) and Kosko (1994) have derived the Universal Approximation Theorem for using fuzzy systems. The theorem suggests that a fuzzy system would be capable of forming an arbitrarily close approximation to any continuous nonlinear function. Generally, most investigations addressed the approximation of given functions using either fuzzy systems or neural networks. Comparisons were made of "approximation accuracy" when using different paradigms, rules, or parameters in the fuzzy (Ying, 1994; Wang, 1994) and neural network systems (Narendra and Parthsarathy, 1990). Recently, the study of reducing the rule size in fuzzy systems has received attention (Rovatti *et al.*, 1995; Chao *et al.*, 1996; Kóczy and Hirota, 1997). For this study, a term "parsimony" is used to represent the economisation of parameters in neural network or fuzzy system design (Manson and Parks, 1992).

As shown by Hu *et al.* (1997), the nonlinearity of fuzzy proportional action can be changed by a number of parameters, called nonlinear tuning parameters. Ideally, we hope to select sufficient parameters and adjust them to realize the desired nonlinear functions (or nonlinear control laws) for the optimal performance of the controller. In most cases, however, this may never be achieved because an explicit expression of the desired nonlinear function does not exist. If this is the case, the problems encountered in using approximation algorithms for the systematic design of fuzzy controllers are better defined as:

(1) to guess the general (or process-independent) properties of the desired nonlinear functions,
(2) to generate a set of closed-form nonlinear functions compatible with the properties, and
(3) to relate the nonlinear tuning parameters quantitatively to their associated versatility and flexibility to cover various nonlinear functions.

This work is a further study of our previous investigation (Hu *et al.*, 1997). Although those systems, using two nonlinear tuning parameters, have shown superior performance in the processes controlled, we recognize a number of weaknesses in their nonlinear

*Received 9 July 1997; revised 25 January 1998; received in final form 20 March 1998. This paper was not presented at any IFAC meeting. This paper was recommended for publication in revised form by Associate Editor K.W. Lim under the direction of Editor C.C. Hang. *Corresponding author* Dr Baogang Hu. Tel. + 86 10 62542971; Fax + 86 10 62551993; E-mail hubg@prlsun4.ia.ac.cn.

† National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academic of Sciences, P.O. Box 2728, Beijing, 100080, China

‡ Also at C-CORE, Memorial University of Newfoundland, St. John's, Canada.

§ Faculty of Engineering and Applied Science Memorial University of Newfoundland, St. John's A1B 3X5 Canada.

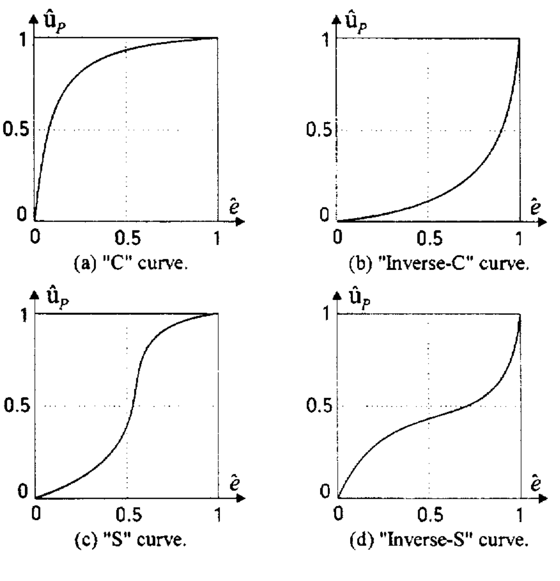


Fig. 1. Four types of simple nonlinear curves.

design. Firstly, their closed-form fuzzy nonlinear mapping requires multiple and non-uniform expressions, which cause the nonlinear analysis to be relatively complex and tedious. Secondly, their associated nonlinear variations is limited in covering the whole spectrum of the C-type curves. This constrains the searching space of optimal solutions to the process. Finally, the system becomes more difficult to design if additional tuning parameters are required.

Motivated to improve the previous fuzzy PID system, we develop a systematic design strategy in this work to synthesize the nonlinear controllers using spline-based approximations. The proposed system constructs a *"control curve"* in a two-dimensional plane. This SISO fuzzy inference scheme simplifies analysis and implementation over a conventional fuzzy system which involves construction of a *"control surface"* (Ying, 1993; Kosko, 1997). The advantages in using low-dimensional fuzzy PID systems have been discussed elsewhere (Hu *et al.*, 1998).

The paper is organized as follows. Section 2 describes why we study proportional actions. In Section 3, we propose the heuristic properties for the proportional actions. The properties are general for any fuzzy PID controller design, rather than process-dependent. In Section 4, the definition of Béziecr curves, which fall into a general form of spline-based functions, is introduced. Section 5 presents the procedures used to generate nonlinear functions using one, two or four parameters. For each case of the selection of parameters, the associated nonlinear flexibility of the system is discussed. The procedures for using more parameters are described in Section 6. Section 7 discusses the nonlinear functions under symmetric considerations. The linguistic representation is discussed in Section 8. Finally, the summary and discussions are given in Section 9.

2. *Why study proportional actions*

In this work, the basic structure of a fuzzy PID controller is the same as in Hu *et al.* (1997). This is an SISO fuzzy system with a cascade structure similar to a conventional PID controller. The scaled discrete-time controller output, $\hat{u}(r)$, is the sum of three terms, represented by

$$\hat{u}(r) = \hat{K}_P \hat{u}_P(r) + \hat{K}_I \sum_{i=0}^{r} \hat{u}_P(i)\Delta t + \hat{K}_D \frac{\Delta \hat{u}_P(r)}{\Delta t},$$

$$r = 0, 1, 2, \ldots;\ \hat{K}_P,\ \hat{K}_I,\ \hat{K}_D \in [0,1], \qquad (1)$$

where $\hat{u}_P$ is a nonlinear (or fuzzy) proportional output; its change is $\Delta\hat{u}_P$ for the sampling period Δt; and $\hat{K}_P$, $\hat{K}_I$ and $\hat{K}_D$ are the normalized proportional, integral and derivative gains, respectively. The overall control output of the fuzzy system is the multiple of a denormalized factor s_u and $\hat{u}$. In comparison, the output of a conventional PID controller is given by

$$u(r) = K_P e(r) + K_I \sum_{i=0}^{r} e(i)\Delta t + K_D \frac{\Delta e(r)}{\Delta t}, \quad r = 0, 1, 2, \ldots, \qquad (2)$$

where e is the error signal, and three gains are regular without normalization. Note that the controller output, eqation (1), in the proposed fuzzy PID system is directly related to $\hat{u}_P$ instead of e. Rewriting equation (1) in the form of equation (2) gives

$$\hat{u} = \left(\hat{K}_P \frac{\hat{u}_P}{\hat{e}}\right)\hat{e} + \hat{K}_I \frac{\Sigma \hat{u}_P}{\Sigma \hat{e}}\right)\Sigma \hat{e}\Delta t + \left(\hat{K}_D \frac{\Delta\hat{u}_P}{\Delta\hat{e}}\right)\frac{\Delta\hat{e}}{\Delta t}$$

$$= (K_P)_{eq}\hat{e} + (K_I)_{eq}\Sigma\hat{e}\Delta t + (K_D)_{eq}\frac{\Delta\hat{e}}{\Delta t}, \qquad (3)$$

where we define $\hat{e}$ to be the normalized error, and $(K_P)_{eq}$, $(K_I)_{eq}$ and $(K_D)_{eq}$ to be the *equivalent* (or *nonlinear*) *proportional, integral and derivative gains* of a conventional PID controller, respectively. Note that the denormalized factor, s_u, is neglected for the equivalency since this does not influence the nonlinear behavior of the gains. For static fuzzy controllers, we drop the notation of time instant r in the expressions. Thus, a nonlinear mapping is constructed between $\hat{e}$ and $\hat{u}_P$ using the form:

$$\hat{u}_P = f(\hat{e}, \mathbf{z}), \qquad (4)$$

where $\mathbf{z}$ is the data set of nonlinear tuning parameters, in the form of either a matrix or vector, and f is a "$\mathscr{R} \to \mathscr{R}$" nonlinear function of its input $\hat{e}$. The function is time invariant and exhibits non-dynamical behavior (Mohler, 1991) unless the tuning parameters are a function of time.

It is understandable that the design of nonlinear functions for the overall controller output is difficult since this output may show a high degree of complexity. This is also true for the nonlinear PID gains. We select the proportional action, $\hat{u}_P$, as the basic nonlinear function for the following reasons. Firstly, any feedback control should include the proportional action. This means whenever an error occurs, there is a control force to regulate the process. Sometimes, the process is possibly controlled without resorting to the integral or derivative actions. Secondly, the fuzzy controller readily simulates a linear PID controller by setting $\hat{u}_P = \hat{e}$, which provides control engineers the simplest and theoretically solid starting point for nonlinear control design. Thirdly, the proportional action may possess the simplest nonlinear curve in the error domain (Hu *et al.*, 1997). The conditions for $\hat{u}_P(\hat{e} = 0)$ and $\hat{u}_P(\hat{e} = 1)$ are predictable (see the next section) but may not be for integral or derivative actions. The gain-scheduling scheme (Åström and Wittenmark, 1995; Zhao *et al.*, 1993; He *et al.*, 1993; Tan *et al.*, 1997), which produces nonlinear gains directly, may require more parameters due to the complex nature of gain changes and their varied ranges. Finally, the proportional action also influences, or controls, the integral and derivative actions in an associated means by $e(r)$. We believe that the selection of proportional actions for the nonlinear design will be a crucial step toward the simplicity of fuzzy, or nonlinear, PID control systems.

3. *Heuristic properties and preferred features*

The knowledge of strict expressions of nonlinear control functions is usually unknown to most processes, and the desired nonlinearity of the proportional actions strongly depends on the process dynamics. Therefore, we will derive the heuristic properties of nonlinear or fuzzy proportional actions by intuitions based on the general applications for such nonlinear design. All properties are presumed to be process- or problem-independent. For this static system, we will present the properties as well as their associated heuristic reasons as follows:

$\mathscr{P}1$ "$\hat{u}_P$ vs. $\hat{e}$" has to be within a normalized compact region, *i.e.*, $-1 \le \hat{e} \le 1$, and $-1 \le \hat{u}_P \le 1$. Boundedness is a necessary condition for a controller to be stable. Normalization is used for realizing standard design procedures.

$\mathscr{P}2$ $\hat{u}_P(\hat{e} \ne 0) \ne 0$ is a necessary condition for a non-zero controller output when an error exists. Otherwise, the steady-state error may occur even for a zero-steady-state-error process.

$\mathscr{P}3$ $\hat{u}_P(\hat{e}=0)=0$ is a necessary condition for a zero steady-state error of process response.

$\mathscr{P}4$ $|\hat{u}_P(\hat{e}=\pm 1)| = \max(|\hat{u}_P|) = 1$ indicates a maximum proportional controller output for a fast rise-up or fall down response when an error signal is at an extremum.

$\mathscr{P}5$ $\hat{e}$ and $\hat{u}_P$ have a *one-to-one* (or two-way) correspondence. A unique controller output is a necessary condition for a mapping from $\hat{e}$ to $\hat{u}_P$.

$\mathscr{P}6$ $\hat{u}_P(\hat{e})$ is a continuous (C^0 *continuity*) function (Driankov *et al.*, 1996). Discontinuity will result in an abrupt change of the response, which may influence the controller performance negatively (Driankov *et al.*, 1996; Lewis and Liu, 1996) or even damage the plants.

$\mathscr{P}7$ $\hat{u}_P(\hat{e})$ has the first derivative (C^1 *continuity*). This indicates the *normalized sensitivity* function, NS $= \partial\hat{u}_P/\partial\hat{e}$ is continuous; and the equivalent derivative gain in equation (3), $(K_D)_{eq} = \hat{K}_D(\partial\hat{u}_P/\partial\hat{e})$, is also continuous.

$\mathscr{P}8$ $\hat{u}_P(\hat{e})$ is a monotonic function. This condition is implied from $\mathscr{P}5$.

$\mathscr{P}9$ $\hat{u}_P/\hat{e}$ has the C^0 continuity. This property is for smooth, non-abrupt, variation of the equivalent proportional gain in equation (3), $(K_P)_{eq} = \hat{K}_P(\hat{u}_P/\hat{e})$.

$\mathscr{P}10$ $\partial\hat{u}_P/\partial\hat{e} > 0$ states that the equivalent derivative gain is positive if $\hat{K}_D \neq 0$. This property implies the curve, $\hat{u}_P(\hat{e})$, is a monotonically increasing function.

$\mathscr{P}11a$ $\hat{u}_P(\hat{e})$ is a non-symmetric function in the situation that a process has non-symmetric behaviors in dynamic responses, say, different overshoot and undershoot characteristics. The necessity for applying this property can also be found out by checking process arrangement. For example, a non-symmetric saturation range, $[0, u_{max}]$ (where u_{max} denotes the upper saturation bound), of the heater to a temperature control process suggests a need to employ this property for a high performance of controllers in both rise-up and fall-down response situations.

$\mathscr{P}11b$ $\hat{u}_P(\hat{e})$ could be an antisymmetric function, $\hat{u}_P(-\hat{e}) = -\hat{u}_P(\hat{e})$, to reduce the total number of tuning parameters although sometimes this may result in poorer performance to the process than using non-symmetric functions.

$\mathscr{P}12$ The nonlinearity of $\hat{u}_P(\hat{e}, \mathbf{z})$ is changeable by tuning $\mathbf{z}$. This is necessary for a nonlinear PID system to adjust the control performance through an optimal solver or human interactive operations. The tuning parameters, $\mathbf{z}$, should be within a compact range, $\mathbf{z} \in [0, 1]$. This is properly for using a genetic-optimization solver where the searching resolution, say, $1/(2^8-1)$, is fixed for the given bits of the coding string, say, eight, on the parameters.

$\mathscr{P}13$ $\hat{u}_P$ is able to form a linear function, $\hat{u}_P = \hat{e}$, by tuning $\mathbf{z}$. The system which satisfies this condition is called a *Guaranteed-PID-Performance* (**GPP**) system (Hu *et al.*, 1997). In this case, the performance analysis for the corresponding linear PID systems will provide a safe (or lower) bound of the specified performance criteria (say, time response error, stability, robustness, etc.) for the GPP system which consists of an optimal solver for tuning the system.

$\mathscr{P}14$ $0 < \hat{u}_P/\hat{e} < \infty$ suggests that the equivalent proportional gain should be positive and finite. This property also states that $\hat{u}_P(\hat{e})$ is a "first and third quadrant nonlinear function", because the curve lies in the first and third quadrants of the normalized compact region.

$\mathscr{P}15$ $\hat{u}_P(\hat{e}_i) \neq \hat{u}_P(\hat{e}_j)$ if $\hat{e}_i \neq \hat{e}_j$. This indicates that there is no "*perfect flat*" zone in the curve. A curve including a flat segment needs more tuning parameters than its approximation.

All properties are basically performance-related, but the properties, $\mathscr{P}1$, $\mathscr{P}4$ and $\mathscr{P}12$, are also considered for the reason of implementation. Some properties are inter-related and can be derived from others. We list them separately since these properties present apparent guidelines for systematic design of fuzzy control systems. While the controllers are designed to include the heuristic properties listed above, three preferred features are also suggested below to the systems from the viewpoint of applications:

$\mathscr{F}1$ The nonlinearity of the system should exhibit *transparency* with respect to tuning parameters, $\mathbf{z}$. This requires explicit expressions of $\hat{u}_P(\hat{e}, \mathbf{z})$. This feature, proposed by Brown and Harris (1994), is important for systematic design as well as for manual tuning during human interactive operations. A closed-form relationship of fuzzy nonlinear mapping is a key step to integrate fuzzy control theory and conventional/modern control theories.

$\mathscr{F}2$ The system should offer a high degree of *versatility* (*flexibility* or *effectiveness*) by using a finite number of nonlinear tuning parameters to generate a wide spectrum of nonlinear functions. Fuzzy controllers, typically high dimensional systems due to their multiple parameters, usually suffer from the difficulty "curse of dimensionality" (Kosko, 1997). This suggests that fuzzy systems should contain a small number of tuning parameters, which can be effectively used to form various nonlinear functions. The feature of versatility should be examined based on a quantitative index, like the *Nonlinearity Variation Index* (**NVI**) defined in Section 5 for nonlinear curves. The index can be used as a process-independent measure for design comparisons of different control systems.

$\mathscr{F}3$ The system should be implemented with *simplicity*. This feature can be evaluated on the following aspects: Firstly, it should have a simple mathematical expression of $\hat{u}_P(\hat{e}, \mathbf{z})$ (say, low-order polynomials). This will simplify nonlinear analysis, algorithm implementation and computations. Secondly, the system should be able to produce uniform representations (say, unchanged forms when adding or removing the parameters). Thirdly, it should be possible to implement the algorithm with high modularity, parallelism, and/or recursiveness. Finally, manipulation with the tuning parameters should be simple and straightforward for an approximation of any given function.

4. *Definition of Bèzier curves* (*Su and Liu*, 1989, *Farin*, 1990)

In this work, we will use *Bézier* curves in the design of fuzzy nonlinear mapping. Let $n+1$ control points $\mathbf{P}_i$ ($i = 0, 1, \ldots, n$), whose coordinates are represented in a column vector form, be given in space. The parametric curve segment of degree n,

$$\mathbf{Q}(s) = \sum_{i=0}^{n} \mathbf{P}_i B_i^n(s), \quad 0 \le s \le 1, \tag{5}$$

is called the *Bézier curve*, which is expressed by a column vector in terms of *Bernstein polynomials*:

$$B_i^n(s) = C_i^n s^i (1-s)^{n-i}, \qquad C_i^n = \frac{n!}{i!(n-i)!}. \tag{6}$$

If $\mathbf{P}_i$ is a point in three (or two) dimensional spaces, a three (or two) dimensional curve will be formed. Equation (5) indicates that the Bézier curve is the sum of the control points weighted by the polynomials. This curve approximates the *control* (or *Bézier*) *polygon* formed by $\mathbf{P}_0, \ldots, \mathbf{P}_n$. Several important properties of Bernstein functions are (Su and Liu, 1989):

$$B_i^n(s) \in [0, 1], \tag{7a}$$

$$\sum_{i=0}^{n} B_i^n(s) \equiv 1, \quad s \in [0, 1], \tag{7b}$$

$$B_i^n(s) = B_i^{n-1}(1-s), \tag{7c}$$

$$\frac{d}{ds} B_i^n(s) = n(B_{i-1}^{n-1}(s) - B_i^{n-1}(s)), \tag{7d}$$

$$B_i^n(s) = (1-s)B_i^{n-1}(s) + sB_{i-1}^{n-1}(s). \tag{7e}$$

For $n = 3$, the four Bernstein basis functions are illustrated in Fig. 2. The properties in equation (7) enable the Bézier curves to possess many appealing features in the application of curve design which may not be shared by B-spline functions. These features include (Su and Liu, 1989): (a) the Bézier curve lies within the convex hull which is defined by control points, (b) the curve interpolates the endpoints, (c) it can be represented by

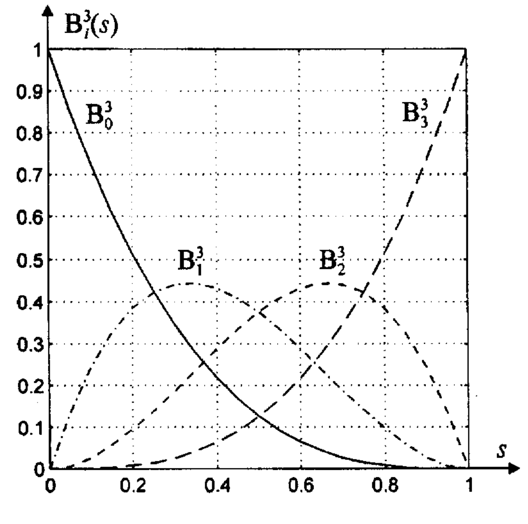


Fig. 2. The Bernstein polynomials: the cubic case.

a uniform and recursive expression for a nonlinear function with any number of parameters. On the other hand, this type of curve may lose two merits compared to B-spline (Su and Liu, 1989): local control and closed approximation to the control polygon. In general, Bézier curves are considered to fall into the category of spline-based functions and they can be represented interchangeably (Farin, 1990).

5. *Nonlinear functions with one, two or four parameters*

In this section, we will use Bézier forms to generate two-dimensional nonlinear curves, $\hat{u}_P(\hat{e}, \mathbf{z})$, compatible with the heuristic properties. For convenience, we consider first the tuning parameters, no more than four, for the curve generation.

Given a positive compact region of $0 \le \hat{e} \le 1$, and $0 \le \hat{u}_P \le 1$ for construction of nonlinear functions, the nonlinear tuning parameters are considered initially by four *control points*, $\mathbf{z}_0 = [\mathbf{P}_0, \mathbf{P}_1, \mathbf{P}_2, \mathbf{P}_3]$, located within the region. A general form of a planar curve, $\mathbf{Q}(s)$, is represented by a recursive form of Bézier curves using these four points:

$$\mathbf{Q}(s) = \begin{Bmatrix} \hat{e}(s) \\ \hat{u}_P(s) \end{Bmatrix} = \mathbf{P}_0^3(s) = (1-s)\mathbf{P}_0^2(s) + s\mathbf{P}_1^2(s),$$

where

$$\mathbf{P}_0^2(s) = (1-s)\mathbf{P}_0^1(s) + s\mathbf{P}_1^1(s), \mathbf{P}_1^2(s) = (1-s)\mathbf{P}_1^1(s) + s\mathbf{P}_2^1(s), \tag{8}$$

$$\mathbf{P}_0^1(s) = (1-s)\mathbf{P}_0(s) + s\mathbf{P}_1(s), \mathbf{P}_1^1(s) = (1-s)\mathbf{P}_1(s) + s\mathbf{P}_2(s),$$

$$\mathbf{P}_2^1(s) = (1-s)\mathbf{P}_2(s) + s\mathbf{P}_3(s).$$

The data flow for calculation of the final function, in a triangular form of *de Casteljau schemes* (Farin, 1990), is shown in Fig. 3. The recursive feature of the calculation is suitable for using neural net implementation. Specifically in our problem, two *endpoints* are fixed at two corner points of the region, $\mathbf{P}_0 = \{0, 0\}^T$ to satisfy $\mathscr{P}3$ (Property 3) and $\mathbf{P}_3 = \{1, 1\}^T$ for $\mathscr{P}4$, respectively. These are considered to be the *endpoint parameter constraints*. The other two *interior* points can be placed at any position within the compact region. Therefore, the nonlinearity of the curves is controlled by four independent tuning parameters, $(P_r)_i$, $r = x, y$; $i = 1, 2$.

Figure 4 shows the functional procedures for the calculation and tuning in the controllers. After the parameter constraints on $\mathbf{z}_0$, the independent tuning parameter data, $\mathbf{z}$, can be represented with two independent row vectors, $\mathbf{z}_x$ and $\mathbf{z}_y$. For the known tuning parameters, the controller output is calculated from the first mapping: "$\hat{e}$ to s", then the second mapping: "s to $\hat{u}_P$". The first mapping needs to solve the cubic

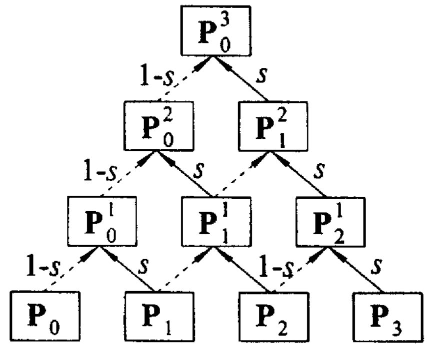


Fig. 3. Data flow for the calculation of Bézier curves.

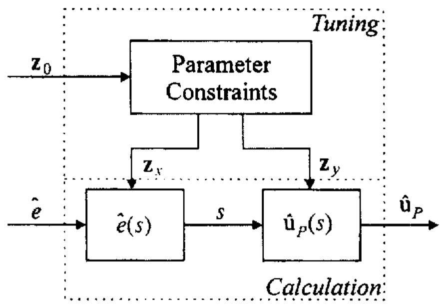


Fig. 4. Control calculation and tuning of nonlinear controllers.

polynomial equation (9a), and the second mapping, equation (9b), is a direct substitution.

$$\hat{e}(s) = 3s(1-s)^2(P_x)_1 + 3s^2(1-s)(P_x)_2 + s^3, \tag{9a}$$

$$\hat{u}_P(s) = 3s(1-s)^2(P_y)_1 + 3s^2(1-s)(P_y)_2 + s^3. \tag{9b}$$

A standard linear equation solver, say, the Newton's method (Golub and Ortega, 1992), can be used for the solution. Since equation (9a) is a monotonic function for any given $\mathbf{P}_1$, and $\mathbf{P}_2 \in [0, 1]$ (see Appendix A), the solutions for s within the interval [0, 1] is guaranteed to be unique. The curve, $\hat{u}_P(\hat{e})$, is readily drawn from the *cross plots* (Farin, 1990) of two curves represented in equation (9).

Now, we define a simple and intuitive method of measuring the flexibility of the system in producing the nonlinearity variations in nonlinear design of two-dimensional curves. Let θ_0 and θ_1 (Fig. 5a) be the curve slopes in radians at $\hat{e} = 0$ and $\hat{e} = 1$, respectively. These slopes also represent the normalized sensitivity (NS), $NS_0 = \tan(\theta_0)$ and $NS_1 = \tan(\theta_1)$, at the two points, $\mathbf{P}_0$ and $\mathbf{P}_1$, respectively. They provide useful information about the control. The low value of the slope (or angle) indicates a low degree of sensitivity to white noise of the system. The high value means the greater derivative action generated by the system. For a linear controller, the NS is a constant, although, a fuzzy controller can produce a varied NS with respect to the error signal. To examine the nonlinearity variations approximately, we define the admissible area/line of the nonlinearity diagram (Fig. 5) on the "θ_0 and θ_1" plane. Here, we call θ_0 and θ_1 *nonlinearity examination parameters*. The points within this area/line mean that their associated θ_0 and θ_1 can be produced by the system. The larger the admissible area, the greater the flexibility of the system in generating the nonlinear functions. It is desirable for a control system, either implemented by fuzzy or neural network version, to have a large area in the diagram but to employ only a small number of tuning parameters. For a dimensionless reason, we proposed the Nonlinearity Variation Index (NVI) in a relative form:

$$\mathrm{NVI}(N_v, N_t, N_e) = \frac{\text{Admissible region in } N_e \text{ dimensional space}}{\text{Whole region in } N_e \text{ dimensional space}}, \tag{10}$$

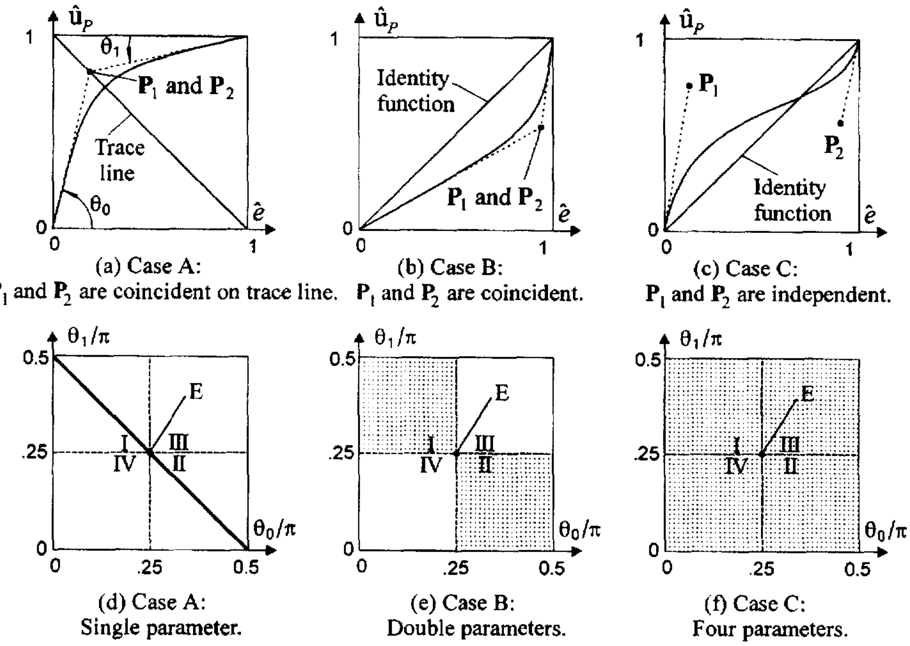


Fig. 5. Bézier curves and their associated admissible area/line of nonlinearity diagrams in three cases. (Admissible line: heavy solid line. Admissible area: gray area. E: point corresponding to a perfectly linear function.) (a) Case A: $\mathbf{P}_1$ and $\mathbf{P}_2$ are coincident on trace line. (b) Case B: $\mathbf{P}_1$ and $\mathbf{P}_2$ are coincident. (c) Case C: $\mathbf{P}_1$ and $\mathbf{P}_2$ are independent. (d) Case A: Single parameter. (e) Case B: Double parameters. (f) Case C: Four parameters.

where N_v, N_t and N_e are the total numbers of input variables, nonlinear tuning parameters and nonlinearity examination parameters, respectively. In order to represent the NVI sufficiently in characterizing the system, N_e should be selected to be equal to N_t. However, this will become very complex for the calculation of NVI when $N_e \geq 3$.

One of the advantages in using Bézier curves is the simple derivation of θ_0 and θ_1. Based on eq. (7d), there have the relationships:

$$\theta_0 = \angle(\mathbf{P}_1 - \mathbf{P}_0) = \tan^{-1}\left[\frac{(P_y)_1 - (P_y)_0}{(P_x)_1 - (P_x)_0}\right] = \tan^{-1}\left[\frac{(P_y)_1}{(P_x)_1}\right], \tag{11a}$$

$$\theta_1 = \angle(\mathbf{P}_3 - \mathbf{P}_2) = \tan^{-1}\left[\frac{(P_y)_3 - (P_y)_2}{(P_x)_3 - (P_x)_2}\right] = \tan^{-1}\left[\frac{1 - (P_y)_2}{1 - (P_x)_2}\right]. \tag{11b}$$

These equations show that θ_0 is simply controlled by the location of $\mathbf{P}_1$ (since $\mathbf{P}_0$ is fixed), and θ_1 by $\mathbf{P}_2$ ($\mathbf{P}_3$ is fixed).

Due to simple equations expressed in equation (9), the nonlinear analysis of the system becomes straightforward and easy. For examples, the cubic polynomial functions satisfy C^0 and C^1 continuities. The functions do not have any flat zone. The nonlinear function, $\hat{u}_P(\hat{e})$, is always within the compact region due to the convex hull property of Bézier curves. The monotonecity of $\hat{u}_P(\hat{e})$ within the compact region is shown in Appendix A. The equivalent proportional gain is readily obtained from equation (9) and $\mathscr{P}9$. For simplicity, we still use the parametric form to express the gain:

$$\hat{e}(s) = 3s(1-s)^2(P_x)_1 + 3s^2(1-s)(P_x)_2 + s^3, \tag{12a}$$

$$(K_P)_{eq}(s) = \hat{K}_P\frac{\hat{u}_P(s)}{\hat{e}(s)} = \hat{K}_P\frac{3(1-s)^2(P_y)_1 + 3s(1-s)(P_y)_2 + s^2}{3(1-s)^2(P_x)_1 + 3s(1-s)(P_x)_2 + s^2}. \tag{12b}$$

In general, $(K_P)_{eq}$ is a nonlinear function, which can be shown from the cross plot of equation (12). A constant $(K_P)_{eq}$ is obtained only when $(P_x)_1 = (P_y)_1$, and $(P_x)_2 = (P_y)_2$. This case corresponds to a liner PID controller with constant gains. Similarly, The equivalent derivative gain can be obtained from equation (9) and $\mathscr{P}7$. The equivalent integral gain can be explicitly represented only when e (or $\hat{e}$) is a known function with respect to time. Based on the proof in Appendix A, we find that the heuristic properties of $\mathscr{P}7$, $\mathscr{P}10$ and $\mathscr{P}14$ are also satisfied by the present system. Therefore, we conclude that the nonlinear functions of the present controller satisfy all the heuristic properties proposed in Section 3. Further investigation is made on the Nonlinear Variation Index of the controllers. We will discuss three cases associated with the different tuning parameter numbers as follows:

Case A: One tuning parameter.

By imposing constraints on $\mathbf{z}_0$, the number of the independent tuning parameters can be reduced. Here is one example of how to impose constraints:

$$(P_x)_1 = (P_x)_2 = 1 - (P_y)_1 = 1 - (P_y)_2 = c, \quad c \in [0, 1], \tag{13}$$

to obtain a single tuning parameter controller. Equation (13) indicates that the two interior control points are coincident and follow a *trace line* of $\hat{u}_P(\hat{e}) = 1 - \hat{e}$ (Fig. 5a). Based on equation (11), we immediately find that the curves have the relation: $\theta_0 = \pi/2 - \theta_1$. This results in a single admissible line in the nonlinearity diagram (Fig. 5d). Denoting $A_l(\theta_0)$ to be a projected length of the admissible line onto the θ_0 axis, we obtain:

$$\text{NVI}(1, 1, 1) = \frac{A_l(\theta_0)}{\pi/2} = 1.0. \tag{14}$$

Using the constraints in equation (13), the controller produces symmetric C-type curves (Fig. 5a). The mirror plane is coincident with the tracking function of the control points. When $c < 0.5$, the controller forms "C-curves" (Fig. 1a). If $c > 0.5$, we get "inverse-C-curves" (Fig. 1b). A perfectly linear function will be obtained when $c = 0.5$, which corresponds to Point E in Fig. 5d. This point divides the nonlinearity diagram into four regions. Each region corresponds to a specific type of curve (Table 1). Note that the shape and location of the admissible line are dependent on the trace line. In this case, equation (14) only

Table 1. Relationship between the regions of nonlinearity diagrams (Fig. 5d–f) and the nonlinear curve types (Fig. 1).

Region number	I	II	III	IV
Curve type	"Inverse-C" (Fig. 1b)	"C" (Fig. 1a)	"Inverse-S" (Fig. 1d)	"S" (Fig. 1c)

gives partial information for the evaluation of the controller. In control applications, if a single nonlinear tuning parameter is used, the location of the admissible line is also important for a control process; this is related to the first issue discussed in the introduction.

Case B: Two tuning parameters.

The constraints for this case are:

$$(P_x)_1 = (P_x)_2 \in [0, 1] \quad \text{and} \quad (P_y)_1 = (P_y)_2 \in [0,1]. \quad (15)$$

This equation suggests that the two interior control points should be coincident (Fig. 5b). Whenever the points are located at the identity function of $\hat{u}_P(\hat{e}) = \hat{e}$, a perfectly linear function is generated. This perfect linear function corresponds to a single point, Point E, in the nonlinearity diagram (Fig. 5e). If the control points are only allowed to vary above the identity function of the compact region, C-curves are formed. Variation below the identity function produces only inverse-C-curves. Now, we will examine the Nonlinear Variation Index of the controller. The two independent admissible areas (Fig. 5e), A_a, are readily obtained from equation (11). Then, we have

$$\mathrm{NVI}(1,2,2) = \frac{A_a(\theta_0, \theta_1)}{(\pi/2)^2} = 0.5. \quad (16)$$

It is interesting to note that the present controller with only two nonlinear tuning parameters has a greater value of NVI than that of the fuzzy controller with the same number of tuning parameters in Hu *et al.* (1997) (where, NVI (1, 2, 2) = 0.366). This means that if a "C" type curve is suggested by the process, the present controller will produce improved flexibility of nonlinearity variations in the process. While that fuzzy controller involved a cumbersome derivation for the nonlinearity variation analysis, this controller requires only simple and straightforward design efforts.

Case C: Four tuning parameters.

This case corresponds to a controller without constraints imposed on the two interior control points (Fig. 5c). The curve analysis is similar to Case B. A perfectly linear function is obtained when the points are located at the identity function. Four types of curves, as shown in Fig. 1, can be generated by this controller. The difference between "C" type and "S" type curves is the occurrence of an *inflection point* on the curves. While the "C" type curve has zero inflection point, the "S" type curve has a single inflection point (Fig. 1). A strict derivation of forming what type of curves with respect to the tuning parameters is quite complex and difficult in this case. For approximation and convenience, we observe that if $\mathbf{P}_1$ is within the region below (or above) the generated curve and $\mathbf{P}_2$ is above (or below) the curve, an "S- (or inverse-S-) curve" is obtained. θ_0 and θ_1 can be varied independently in a range $[0, \pi/2]$.

In this case, for simplification, we will use the nonlinearity diagram with respect to θ_0 and θ_1, for calculations of NVI (1, 4, 2). Therefore, the controller covers a full admissible area in the nonlinearity diagram (Fig. 5f).

$$\mathrm{NVI}(1,4,2) = \frac{A_a(\theta_0, \theta_1)}{(\pi/2)^2} = 1.0. \quad (17)$$

The system is able to span a complete spectrum of nonlinearity in terms of NVI (1, 4, 2). We use equation (17) to indicate that any one of the four types of simple curves in Fig. 1 can be generated by applying four tuning parameters. To realize the same NVI value, we found it was very complex to design the fuzzy controller using the same tuning parameters. The strictly defined NVI (1, 4, 4), represented in four dimensions, is quite complex and it involves a calculation of hyper-volumes. However, it should be understood that adding two more tuning parameters enhances the controller not only by encompassing the S-type curves, but also by enlarging the spectrum of C-type curves. Figure 6 demonstrates two different C-type curves for the fixed θ_0 and θ_1. The length between two associated control points changes the curvature of the curve. Therefore, the variations of C-type curves are greatly enriched in this case.

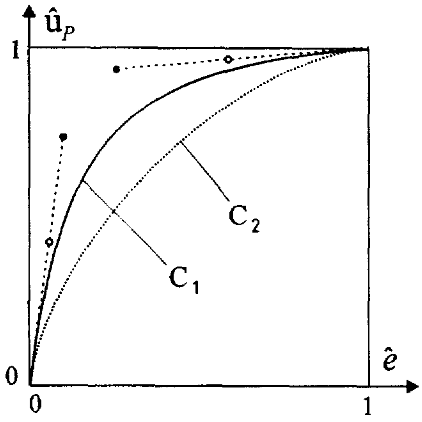
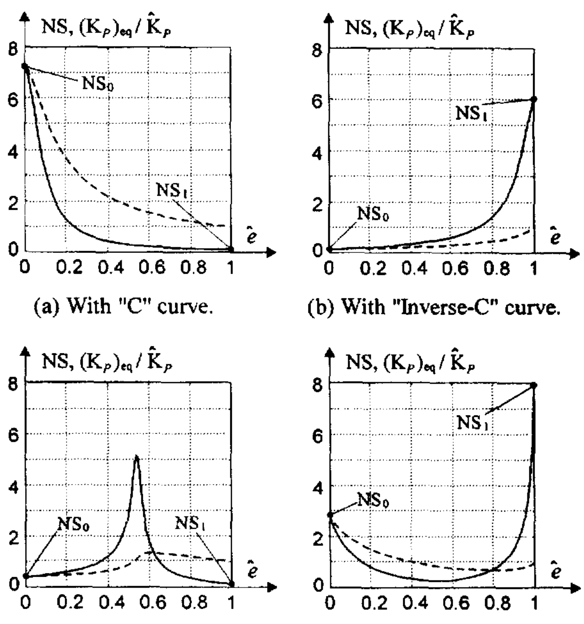


Fig. 6. Different C-type curves for the fixed θ_0 and θ_1 when using four tuning parameters. (Solid points: control points corresponding to C_1. Hollow points: Control points corresponding to C_2.)



(a) With "C" curve. (b) With "Inverse-C" curve. (c) With "S" curve. (d) With "Inverse-S" curve.

Fig. 7. Nonlinear curves of NS$(=(K_D)_{eq}/\hat{K}_D)$ and $(K_P)_{eq}/\hat{K}_P$ associated with four types of simple curves in Fig. 1. (Solid line: NS(e). Dash line: $(K_P)_{eq}/\hat{K}_P$). (a) With "C" curve. (b) With "Inverse-C" curve. (c) With "S" curve. (d) With "Inverse-S" curve.

When using four tuning parameters, at least four types of nonlinear curves are formed. More complicated curves are possible, but we believe that the "C" and "S" types are basic to approximate any complex curve piecewisely with efficiency. While the curve of $\hat{u}_P(\hat{e})$ is generated, the functions for the normalized sensitivity, NS($\hat{e}$), and equivalent proportional gain, $(K_P)_{eq}$, will be obtained readily. Figure 7 shows the nonlinear curves for NS$(=(K_D)_{eq}/\hat{K}_D$, if $\hat{K}_D \neq 0)$ and $(K_P)_{eq}/\hat{K}_P$ in the normalized error domain associated with the four curves in Fig. 1. Note that the NS($\hat{e}$) for "C" type curves is monotonic, but has a peak or valley for "S" type curves. The curves of $(K_P)_{eq}/\hat{K}_P$ always start at the height of NS_0 and end at a unit. The curve features of both equivalent proportional and derivative gains have confirmed our discussions in Section 2.

At this point, it is interesting to note that the parametric representations of the spline-based functions, using an indirect mapping scheme ($x \to s \to y$), offer unique characteristics to the nonlinear approximators when compared to other approximators. For example, a loop-type curve (say, circle, ellipsoid, or spiral) can be generated by simple polynomial functions (Su and Liu, 1989), which seems very difficult, if not impossible, by using direct-mapping ($x \to y$) approximators, say, the fuzzy or neural networks (Ying, 1994; Wang, 1994; Brown and Harris, 1994; Kosko, 1997). Note that Brown and Harris also applied B-spline functions for their systems, which have the local control feature when adjusting the tuning parameters. This discussion shows the powerfulness of the proposed method in generating the various nonlinear curves even they may be not the case in the control applications.

6. *Nonlinear functions using more than four parameters*

For generating nonlinear functions more complex than the four basic curves in the compact region (Fig. 1), additional parameters are needed. This section illustrates a general method which present a uniform expression in generation of nonlinear functions regardless of the number of parameters. As discussed before, the heuristic properties are still applied for those nonlinear functions.

The inclusion of more parameters starts with the division of the compact region into subdivisions. The first simplest case occurs when there are two subdivisions (Fig. 8). It is interesting to see that this case is similar to a "two-stage" fuzzy PI controller developed by Li and Lau (1989), where two sets of rules have been used in "coarse" and "fine" error ranges, respectively. In Fig. 8, the first subdivision is "a-b-c-d", and the second is "d-e-f-g". The total number of control points will be seven, $\mathbf{P}_0, \ldots, \mathbf{P}_6$. Imposing the endpoint constraints eliminates $\mathbf{P}_0$ and $\mathbf{P}_6$ from the tuning parameter data. Another constraint is necessitated by $\mathscr{P}7$, as described in Section 3 around the *junction point*, $\mathbf{P}_3$:

$$\angle(\mathbf{P}_3 - \mathbf{P}_2) = \angle(\mathbf{P}_4 - \mathbf{P}_3) \quad \text{and} \quad |\mathbf{P}_3 - \mathbf{P}_2| = |\mathbf{P}_4 - \mathbf{P}_3|, \tag{18}$$

to realize the C^1 continuity between the two subdivisions (Farin, 1990). This constraint indicates that the location of $\mathbf{P}_2$ implies the location of $\mathbf{P}_4$. These two control points are asymmetrically aligned with $\mathbf{P}_3$ as shown in Fig. 8. Therefore, the independent tuning parameters for the two subdivisions are:

$$\mathbf{z} = [\mathbf{P}_1, \mathbf{P}_2, \mathbf{P}_3, \mathbf{P}_5]. \tag{19}$$

The total number of the tuning parameters becomes eight. Note that if an optimal solver is applied for searching the optimal tuning parameters, the two subdivisions will be readily determined from the parameters. The final function, represented in a triangular form of the de Casteljau schemes (Fig. 9), always keeps a uniform expression when adding or removing the parameters. For convenience, however, the calculation of the final function is made on two *intermediate functions*, $\mathbf{P}_0^3(s)$ and $\mathbf{P}_3^3(s)$ (Fig. 9). The procedure for the calculations is as follows:

(1) Set the subdivisions according to the locations of endpoints and junction points, say, "a-b-c-d" and "d-e-f-g" as shown in Fig. 8;
(2) Determine the *permitted-location region* (**PLR**) for each control point in equation (19). In this case, we have $\mathbf{P}_3 \in (0, 1)$ (this is an open region), $\mathbf{P}_1 \in$ Region "a-b-c-d", $\mathbf{P}_5 \in$ Region "d-e-f-g", and $\mathbf{P}_2 \in$ Region "c-d-k-j". The PLR for $\mathbf{P}_2$ is calculated from a rectangular region given by:

$$P_2\text{'s PLR: Width} = \text{minimum}(\text{Length "}c\text{-}d\text{"}, \text{Length "}d\text{-}e\text{"})$$

$$\text{Height} = \text{minimum}(\text{Length "}b\text{-}c\text{"}, \text{Length "}g\text{-}d\text{"}),$$

becomes smaller than its associated subdivision due to the consideration that $\mathbf{P}_4$ should be within the compact region too (Fig. 8), that is, $\mathbf{P}_4 \in$ Region "d-i-h-g".

(3) Introduce a *local parameter* q (Farin, 1990) for the interval $c_1 \le s \le c_2$:

$$q = \frac{s - c_1}{c_2 - c_1}, \quad q \in [0, 1]. \tag{20}$$

The interval points c_1 and c_2 are calculated from equation (9a) using the known subdivision data with respect to $\hat{e}$.
(4) Calculate the piecewise Bézier curves for each subdivision from the nonlinear mapping, "$\hat{e}$ to q", "q to $\hat{u}_P$".

The above procedures can be extended to a general case, $N_p = 1, 2, 4, 8, \ldots, 4m$, where m is an integer number. Whenever one subdivision is added in, four additional parameters are augmented to the tuning parameter set. It seems difficult to define an NVI in this case for an overall evaluation of the function. The original definition of NVI can be used in each subdivision.

7. *Nonlinear functions under symmetric considerations*

All derivations above regarding the tuning parameters are valid for positive values of $\hat{e}$ in the interval [0, 1]. If a nonsymmetric function is required in the design, we will need another set of tuning parameters to form the nonlinear mapping for the negative values of $\hat{e}$. The procedures for realizing the nonlinear functions are the same as those described in the previous sections. Two sets of tunning parameters, not necessarily equal, will vary the nonlinear functions in the positive $\hat{e}$ and

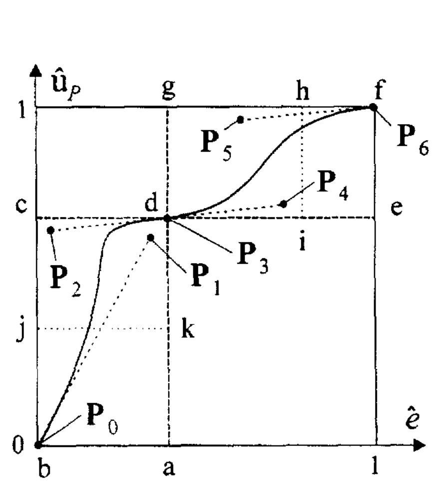


Fig. 8. Generation of nonlinear function from two subdivisions.

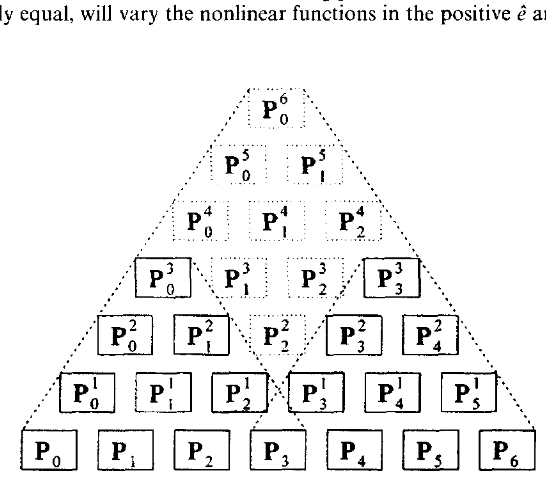


Fig. 9. Calculation for two subdivisions.

negative $\hat{e}$ domains. Though they can be joined in as a single uniform expression, for simplicity, however, we calculate the whole curve separately. If an antisymmetric function is desired ($\mathscr{P}$11b), the calculation of the function in the negative $\hat{e}$ domain can be made by

$$\hat{u}_P(-\hat{e}) = -\hat{u}_P(\hat{e}). \quad (21)$$

8. *Linguistic representation*

Some researchers (Wang, 1994; Driankov *et al.*, 1996) have commented that fuzzy controllers are simply a kind of nonlinear transfer element. The present controller, emulating a fuzzy controller, can be implemented without involving the concept of fuzzy logic. Given the known relationship of the mapping, like equation (9), the system can calculate the controller output directly. An optimization solver can be used for tuning a process according to the specific performance criteria. However, as pointed out by Zadeh (1996), "*the main contribution of fuzzy logic is a methodology for computing with words*". The capability of incorporating human linguistic descriptions has significant importance in applications since many real-world problems are directly solved by human expertise and decision making. In control problems, fuzzy systems produce an elegant solution which makes use of human knowledge in handling many complex processes, which might have failed if traditional control approaches had been used. Therefore, the feature of computing with words is imperatively preserved in the present system.

The fuzzy system is constructed from the fuzzy "IF–THEN" rules using linguistic information to describe the control action from human experts. The simplest fuzzy linguistic representation, based on the heuristic property $\mathscr{P}$11a, for the present controller is

$$\begin{aligned} &\text{R1: } \textit{If}(\hat{e} \textit{ is positive}) \textit{ then } \{\hat{u}_P = (\hat{u}_P)^+\} \\ &\text{R2: } \textit{If}(\hat{e} \textit{ is negative}) \textit{ then } \{\hat{u}_P = (\hat{u}_P)^-\} \end{aligned} \quad (22)$$

where the superscripts of " + " and " − " are used for describing the functions in positive $\hat{e}$ and negative $\hat{e}$ domains, respectively. This set of rules works like the control scheme switchers, and has lost the conventional meaning of fuzzy reasoning. We believe that this is the weakness of the present controllers. However, the benefits are significant for applying this approach since it provides a systematic and simple solution to formulate human knowledge in nonlinear design for PID control applications. For example, some rules can be constructed according to the control curve types (or the number of nonlinear tuning parameters):

R1: *If(performance is unkmown) then* ($\hat{u}_P$ = *linear fuction*),

R2: *If (performance for linear fuction is unsatisfied) then* ($\hat{u}_P$ = *C-type fuctions*),

R3: *If (performance for C-type Fuction is unsatisfied) then* ($\hat{u}_P$ = *S-type fuctions*),

R4: *If (performance for S-type fuction is unsatisfied) then* ($\hat{u}_P$ = *subdivisional functions*). (23)

This set of rules represents the "*simple-first*" strategy for fine tuning of model-free controls where little or no knowledge is available to the plant. We consider a linear function as a starting point since the corresponding linear PID controllers have a well-developed control technique and are familiar to the control engineers. This strategy, different from the strategy of "rule (or complexity) reduction", requires less design efforts and is more compatible with engineering practice. The optimization in using a rule-reduction design method may involve two conflicting criteria with respect to simplicity and accuracy, respectively. This makes it difficult to define the appropriate criteria. However, a simple-first design method will apply a single accuracy criterion without involving a trade-off consideration of two conflicting criteria.

In equation (23) we assume that the control performance is improved by adding more tunning parameters. Note that Rule 2 in equation (23) was proved to be numerically true in Hu *et al.* (1997) for the proposed properties. However, the assumption becomes invalid if the heuristic properties of the desired nonlinear functions are not properly defined. At least we can find that if the number of tuning parameters (or subdivisions) is close to infinite, a C-type curve will be obtained. This is against the assumption. This discussion suggests that equation (23), as well as the proposed heuristic properties, need to be refined based on further investigations into the relationship between "nonlinearity and performance".

9. *Summary and discussions*

In this work, a nonlinear PID controller is designed using spline-based functions to emulate a fuzzy PID system. The proportional actions are selected as a basic function for the nonlinear "control curve" design. This selection is of great significance in nonlinear control design since this action may provide maximum intrinsic simplicity of nonlinear functions than other control actions and tuning gains. The heuristic properties are proposed for the functions, which can also serve as design guidelines for a single-input fuzzy controller. Due to several distinguished features of spline functions, such as generality, parametrization, parallelism and recursiveness, the present design method has been greatly enhanced. While the associated nonlinearity variations of the previous controller in Hu *et al.* (1997) did not cover the whole range of the C-type curves, the present controller has encompassed a wider spectrum of C-type curves when using two tuning parameters. At the same time, the nonlinear analysis becomes straightforward and simple for the control design, and the system is readily extended to include more nonlinear tuning parameters (say, a complete spectrum of C- and S-type curves in terms of NVI (1, 4, 2) when using four tuning parameters). A perfect linear function can be realized to generate a guaranteed-PID-performance system, which is considered to be the starting point for tuning of the present system.

In PID control applications, we view any controller as a nonlinear approximator. It works to approximate the nonlinear functions of the controller output which are implicitly suggested by the process. These functions, varying with the specific performance criteria, are generally unknown to control engineers. For this reason, any approximator, either fuzzy, neural network or spline-based version, should be evaluated using three features: transparency, versatility and simplicity. We have used the nonlinearity variation index as a process-independent measure, not the approximation accuracy of the specific function, to evaluate the control systems. This work explores a new direction for a systematic design of a nonlinear approximator in control applications. We believe that further investigations of the approach are needed. An improved definition of the nonlinearity variation index is required to evaluate a system's nonlinear characteristics sufficiently. Additional studies are also expected regarding the implementation of the present approach with a true fuzzy reasoning approach, as well as integration with neural networks and genetic optimization techniques for applications of high-performance controllers.

Acknowledgements—The authors gratefully acknowledge the support of Natural Sciences and Engineering Research Council of Canada, the Canadian Space Agency, and Petro-Canada Resources. The editorial assistance from Ms E. Nesbitt and Ms P. LeFeuvre is appreciated.

References

Åström, K. J. and B. Wittenmark (1995). *Adaptive Control*, 2nd edn. Addison-Wesley, New York.

Brown, M. and C. Harris (1994). *Neurofuzzy Adaptive Modelling and Control*. Prentice-Hall, New York.

Buckley, J. J. (1992). Universal fuzzy controllers. *Automatica*, **28**, 1245–1248.

Chao, C. T., Y. J. Chen and C. C. Teng (1996). Simplification of fuzzy-neural systems using similarity analysis. *IEEE Trans. Systems, Man, Cybern.*, **B26**, 344–354.

Cotter, N. (1990). The Stone-Weierstrass theorem and its application to neural networks. *IEEE Trans. Neural Networks*, **1**, 290–295.

Driankov, D., H. Hellendoorn and M. Reinfrank, M. (1996). *An Introduction to Fuzzy Control*, 2nd edn. Springer, Berlin.

Farin, G. (1990). *Curves and Surfaces for Computer Aided Geometric Design, A Practical Guide*, 2nd edn. Academic Press, Boston,

Funahashi, K. (1989) On the approximate realization of continuous mapping by neural networks. *Neural Networks*, **2,** 183–192.

Golub, G. H. and J. M. Ortega (1992). *Scientific Computing and Differential Equations, An Introduction to Numerical Methods*. Academic Press, Boston.

Harris, C. J., C. G. Moore and M. Brown (1993). *Intelligent Control, Aspects of Fuzzy Logic and Neural Nets*. World Scientific, Singapore.

He, S.-Z., S. Tan, F.-L. Xu and P.-Z. Wang (1993). Fuzzy self-tuning of PID controllers. *Fuzzy Sets and Systems*, **56,** 37–46.

Hornik, K., M. Stinchcombe and H. White (1989). Multilayer feedforward networks are universal approximators, *Neural Networks*, **2,** 359–366.

Hu, B.-G., G. K. I. Mann and R. G. Gosine (1997). New methodology for analytical and optimal design of fuzzy PID controllers. *IEEE Trans. Fuzzy Systems*, (submitted).

Hu, B.-G., G. K. I. Mann and R. G. Gosine (1998). Study of dimensionality of fuzzy inference for fuzzy PID controllers. *Acta Automatica Sinica*, **24** (5).

Lewis, F. L. and K. Liu (1996). Towards a paradigm for fuzzy logic control. *Automatica*, **32,** 167–181.

Li, Y. F. and C. C. Lau (1989). Development of fuzzy algorithms for servo systems. *IEEE Control System Mag*. April, 1989, 65–72.

Kóczy, L. T. and K. Hirota (1997). Size reduction by interpolation in fuzzy rule bases. *IEEE Trans. Systems, Man, Cybern.-Part B: Cybern.*, **27,** 14–25.

Kosko, B. (1994). Fuzzy Systems as Universal Approximators. *IEEE Trans. Comput.*, **43,** 1329–1333.

Kosko, B. (1997). *Fuzzy Engineering*. Prentice-Hall, New Jersey.

Manson, J. C. and P. C. Parks (1992). Selection of neural network structures: some approximation theory guidelines. In *Neural Networks for Control and Systems*, eds K. Warwick, G. W. Irwin and K. J. Hunt, pp. 151–180. Peter Peregrinus, London.

Mohler, R. R. (1991). *Nonlinear Systems, Vol. 1: Dynamics and Control*. Prentice-Hall, Englewood Cliffs, New Jersey.

Narendra, K. S. and K. Parthsarathy (1990). Identification and control of dynamical systems using neural networks. *IEEE Trans. Neural Networks*, **1,** 4–27.

Rovatti, R., Guerrieri, R. and Baccarani, G. (1995). An enhanced two-level Boolean synthesis methodology for fuzzy rules minimization. *IEEE Trans. Fuzzy Systems*, **3,** 288–299.

Su, B.-Q. and D.-Y. Liu (1989). *Computer Geometry, Curves and Surface Modeling*. Academic Press, Boston.

Tan, S., C. C. Hang and J. S. Chai (1997). Gain scheduling: from conventional to neuro-fuzzy. *Automatica*, **33,** 411–419.

Wang, L.-X. and J. M. Mendel (1992). Fuzzy basic functions, Universal approximation, and Orthogonal Least-squares learning. *IEEE Trans. Neural Networks*, **3,** 807–814.

Wang, L.-X. (1994). *Adaptive Fuzzy systems and Control, Design and Stability Analysis*. PTR Prentice-Hall, New Jersey.

Ying, H. (1993). A nonlinear fuzzy controller with linear control rules is the sum of a global two-dimensional multilevel relay and a local nonlinear proportional-integral control. *Automatica*, **29,** 499–505.

Ying, H. (1994). Sufficient conditions on general fuzzy systems as function approximations. *Automatica*, **30,** 521–525.

Zadeh, L. A. (1996). Fuzzy logic = Computing with words. *IEEE Trans. Fuzzy Systems*, **4,** 103–111.

Zhao, Z.-Y., M. Tomizyoshi and S. Isaka (1993). Fuzzy gain scheduling of PID controllers. *IEEE Trans. System Man, Cybernet.*, **23,** 1392–1398.

Appendix A—Proof of the monotonecity of $\hat{u}_P(\hat{e})$ in the compact region

The proof of the monotonecity of $\hat{u}_P(\hat{e})$ can be made on their associated parametric functions of equation (9). If both $\hat{e}(s)$ and $\hat{u}_P(s)$ are monotonic to s in the interval [0, 1], respectively, the curve of $\hat{u}_P$ will be monotonic with respect to $\hat{e}$ in its compact region. Since $\hat{e}(s)$ has the same form as $\hat{u}_P(s)$, the conclusion on $\hat{e}(s)$ will be valid for $\hat{u}_P(s)$. Therefore, the derivation is only given to the $\hat{e}(s)$. The derivation is conducted on the differential of $\hat{e}(s)$, which is obtained from equation (9):

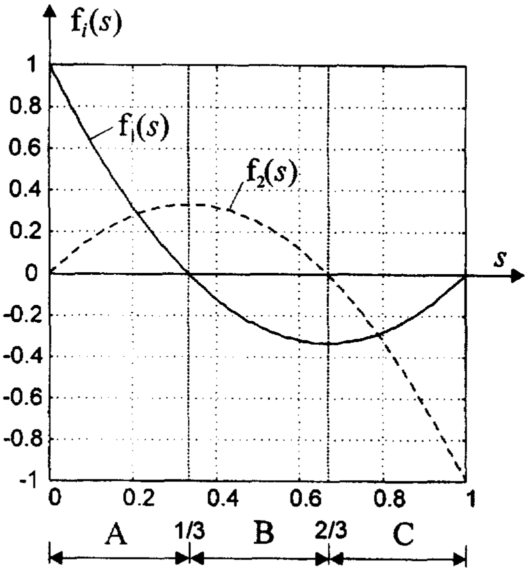


Fig. A1. The curves of $f_1(s)$ and $f_2(s)$.

$$\frac{d\hat{e}(s)}{ds} = 3\{(P_x)_1 f_1(s) + (P_x)_2 f_2(s) + s^2\}, \qquad (A.1)$$

where

$$f_1(s) = 1 - 4s + 3s^2 \qquad (A.2a)$$

$$f_2(s) = 2s - 3s^2, \qquad (A.2b)$$

and $(P_x)_1$ and $(P_x)_2$ are the independent tuning parameters within the range of [0, 1]. For simplifying the analysis, two functions, $f_1(s)$ and $f_2(s)$, are illustrated in Fig. A1, and they are examined according to three ranges (which are divided by two cross-zero points of the functions) as below:

Range A: $0 \le s < 1/3$.

Since $f_1(s) \ge 0$ and $f_2(s) \ge 0$ in this range, the differential of $\hat{e}(s)$, in equation (A.1), will always be non-negative:

$$\frac{d\hat{e}(s)}{ds} \ge 0. \qquad (A.3)$$

Range B: $1/3 \le s < 2/3$.

In this range, $f_1(s) \le 0$ and $f_2(s) \ge 0$. The minimum of equation (A.1) within this range is when $(P_x)_1 = 1$, and $(P_x)_2 = 0$. We can find a relation for equation (A.1):

$$\left[\frac{d\hat{e}(s)}{ds}\right]_{min} = 12\left(s - \frac{1}{2}\right)^2 \ge 0. \qquad (A.4)$$

Range C: $2/3 \le s \le 1$.

Both functions are negative in this range (Fig. A.1). The minimum of equation (A.1) is obtained when $(P_x)_1 = (P_x)_2 = 1$. Then, equation (A.1) has the relation:

$$\left[\frac{d\hat{e}(s)}{ds}\right]_{min} = 3(s - 1)^2 \ge 0. \qquad (A.5)$$

Equations (A.3)–(A.5) confirm that $\hat{e}(s)$, having non-negative differential, is a monotonically increasing function with respect to s in the interval.