An Optimal Fuzzy PID Controller

K. S. Tang, *Member, IEEE*, Kim Fung Man, *Senior Member, IEEE*, Guanrong Chen, *Fellow, IEEE*, and Sam Kwong, *Member, IEEE*

***Abstract*—This paper introduces an optimal fuzzy proportional–integral–derivative (PID) controller. The fuzzy PID controller is a discrete-time version of the conventional PID controller, which preserves the same linear structure of the proportional, integral, and derivative parts but has constant coefficient yet self-tuned control gains. Fuzzy logic is employed only for the design; the resulting controller does not need to execute any fuzzy rule base, and is actually a conventional PID controller with analytic formulas. The main improvement is in endowing the classical controller with a certain adaptive control capability. The constant PID control gains are optimized by using the multiobjective generic algorithm (MOGA), thereby yielding an optimal fuzzy PID controller. Computer simulations are shown to demonstrate its improvement over the fuzzy PID controller without MOGA optimization.**

***Index Terms*—Fuzzy control, genetic algorithm, optimization, proportional–integral–derivative controller.**

I. Introduction

CONVENTIONAL proportional–integral–derivative (PID) controllers have been well developed and applied for about half a century [4], and are extensively used for industrial automation and process control today. The main reason is due to their simplicity of operation, ease of design, inexpensive maintenance, low cost, and effectiveness for most linear systems. Recently, motivated by the rapidly developed advanced microelectronics and digital processors, conventional PID controllers have gone through a technological evolution, from pneumatic controllers via analog electronics to microprocessors via digital circuits [4], [7].

However, it has been known that conventional PID controllers generally do not work well for nonlinear systems, higher order and time-delayed linear systems, and particularly complex and vague systems that have no precise mathematical models. To overcome these difficulties, various types of modified conventional PID controllers such as autotuning and adaptive PID controllers were developed lately [1]–[3], [7]. Also, a class of nonconventional type of PID controller employing fuzzy logic has been designed and simulated for this purpose [6], [7], [16]–[18], [21]–[23]. This fuzzy PID controller has the following special features.

1) It has the same linear structure as the conventional PID controller, but has constant coefficient, self-tuned control gains: the proportional, integral, and derivative gains are nonlinear functions of the input signals.
2) The controller is designed based on the classical discrete PID controller, from which the fuzzy control law is derived.
3) Membership functions are simple triangular ones with only four fuzzy logic IF–THEN rules. The fuzzification, control-rule execution, and defuzzification steps are all embedded in the final formulation of the fuzzy control law. The resulting control law is an explicit conventional formula, so the controller works just like a conventional PID controller, while the fuzzification–rules–defuzzification routine is not needed throughout the entire control process.

Stability of these fuzzy PID controllers has also been analyzed and is guaranteed [6]–[9], [17], [18], [21], [22]. Many simulation and practical examples have been given to show the superior performance of this class of fuzzy PID controllers [7], [8]. However, despite the significant improvement of the fuzzy PID controllers over their classical counterparts, it is noted that these fuzzy PID controllers do not meet specific optimality criteria. The constant control gains of these controllers are tuned manually, so generally do not achieve their best possible performance due to the lack of optimization. Therefore, how to incorporate optimality into these successful controllers remains an interesting and important issue to be further addressed.

This paper aims to equip the fuzzy PID controllers with a certain optimality by using the multiobjective generic algorithm (MOGA), so as to obtain an optimal fuzzy PID controller. Computer simulations are shown to demonstrate its improvement over the fuzzy PID controller without MOGA optimization.

The remainder of the paper is organized as follows. In Sections II and III, a representative fuzzy PI+D controller is introduced, which is used as a platform for the description of the GA-based optimization method proposed in the present paper. This is followed by a detailed description of the GA approach for the optimization of the fuzzy PI+D controller in Section IV. Simulations are then given in Section V to demonstrate the improvement of the GA-based optimization on control gain determination, as compared to the fuzzy PI+D control systems without using GA optimization. Conclusions are drawn in Section VI, with some comments and discussion.

II. Fuzzy PI+D Controller

The fuzzy PI+D controller is a digital controller, which contains a fuzzy PI+D control units arrangement, called the deriva-

Manuscript received February 26, 2001. Abstract published on the Internet June 6, 2001. This work was supported by the City University of Hong Kong under Grant 7000796.

K. S. Tang, K. F. Man, and G. Chen are with the Department of Electronic Engineering, City University of Hong Kong, Kowloon, Hong Kong (e-mail: kstang@ee.cityu.edu.hk).

S. Kwong is with the Department of Computer Science, City University of Hong Kong, Kowloon, Hong Kong.

Publisher Item Identifier S 0278-0046(01)06276-1.

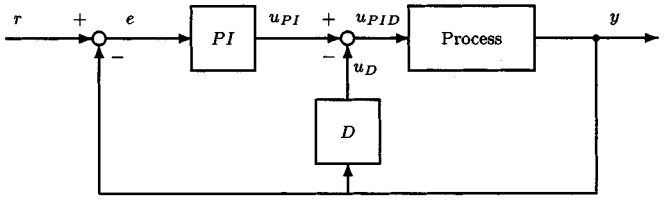

Fig. 1. Conventional continuous-time PI+D control system.

tive of output, as shown in Fig. 1. This arrangement is often desirable if the reference input contains discontinuities [24].

In the design phase [8], [21], we start with the continuous conventional PI+D controller and then use the standard bilinear transform to convert it to the corresponding digital controller. The next two sections discuss this procedure.

A. Fuzzy PI Controller

The output of the conventional analog PI controller in the frequency s domain, as can be verified easily from Fig. 1, is given by

$$u_{\rm PI}(s) = \left(K_p^c + \frac{K_i^c}{s}\right) E(s)$$

where K_p^c and K_i^c are the proportional and integral gains, respectively, and $E(s)$ is the tracking error signal.

This equation can be transformed into the discrete version by applying the bilinear transformation $s = (2/T)((z+1)/(z-1))$, where $T > 0$ is the sampling period, which results in the following form:

$$u_{\rm PI}(z) = \left(K_p^c - \frac{K_i^c T}{2} + \frac{K_i^c T}{1 - z^{-1}}\right) E(z).$$

By letting

$$K_p = K_p^c - \frac{K_i^c T}{2} \quad \text{and} \quad K_i = K_i^c T$$

and then taking the inverse z transform, we obtain

$$u_{\rm PI}(nT) - u_{\rm PI}(nT-T) = K_p[e(nT) - e(nT-T)] + K_i T e(nT). \quad (1)$$

Dividing (1) by T and then rearranging terms, we have

$$u_{\rm PI}(nT) = u_{\rm PI}(nT-T) + T\Delta u_{\rm PI}(nT) \quad (2)$$

where

$$\Delta u_{\rm PI}(nT) = K_p e_v(nT) + K_i e_p(nT) \quad (3)$$

with

$$e_v(nT) = \frac{e(nT) - e(nT-T)}{T}, \quad (4)$$

$$e_p(nT) = e(nT). \quad (5)$$

Here, $\Delta u_{\rm PI}(nT)$ is the incremental control output of the PI controller, $e_p(nT)$ is the error signal, and $e_v(nT)$ is the rate of change of the error signal.

By further replacing the term $T\Delta u_{\rm PI}(nT)$ with a fuzzy control action $K_{u\rm PI}\Delta u_{\rm PI}(nT)$, we arrive at

$$u_{\rm PI}(nT) = u_{\rm PI}(nT-T) + K_{u\rm PI}\Delta u_{\rm PI}(nT) \quad (6)$$

in which $K_{u\rm PI}$ is a constant control gain to be determined.

B. Fuzzy D Controller

The D controller in the PI+D control system, as shown in Fig. 1, satisfies

$$u_{\rm D}(s) = sK_d^c Y(s) \quad (7)$$

where K_d^c is the control gain and $Y(s)$ is the output signal. Under the bilinear transformation, (7) becomes

$$u_{\rm D}(z) = K_d^c \frac{2}{T}\frac{1 - z^{-1}}{1 + z^{-1}} Y(z) \quad (8)$$

so that

$$u_{\rm D}(nT) + u_{\rm D}(nT-T) = \frac{2K_d^c}{T}[y(nT) - y(nT-T)]. \quad (9)$$

Dividing (9) by T and then rearranging terms yields

$$u_{\rm D}(nT) = -u_{\rm D}(nT-T) + K_{u\rm D}\Delta u_{\rm D}(nT) \quad (10)$$

where $K_{u\rm D}$ is a constant control gain to be determined,

$$\Delta u_{\rm D}(nT) = K_d \Delta y(nT) \quad (11)$$

is the incremental control output of the fuzzy D controller,

$$\Delta y(nT) = \frac{y(nT) - y(nT-T)}{T} \quad (12)$$

is the rate of change of the output y, and

$$K_d = \frac{2K_d^c}{T}. \quad (13)$$

To enable better performance of this D controller, we have slightly modified (11) by adding the signal $y_d(nT)$ to its right-hand side, where

$$y_d(nT) = y(nT) - r(nT) = -e(nT) \quad (14)$$

so as to obtain

$$\Delta u_{\rm D}(nT) = K_d \Delta y(nT) + y_d(nT). \quad (15)$$

C. Fuzzy PI+D Controller

Finally, the overall fuzzy PI+D control law can be obtained by algebraically summing the fuzzy PI control law (6) and the fuzzy D law (10) together. The result is

$$u_{\rm PID}(nT) = u_{\rm PI}(nT-T) + K_{u_{\rm PI}}\Delta u_{\rm PI}(nT) + u_{\rm D}(nT-T) - K_{u\rm D}\Delta u_{\rm D}(nT). \quad (16)$$

Equation (16) will be referred to as the fuzzy PI+D control law throughout the paper.

The overall conventional PI+D control system is shown in Fig. 2. To this end, the fuzzy PI and fuzzy D controllers will be inserted into the figure, resulting in the configuration shown in Fig. 3.

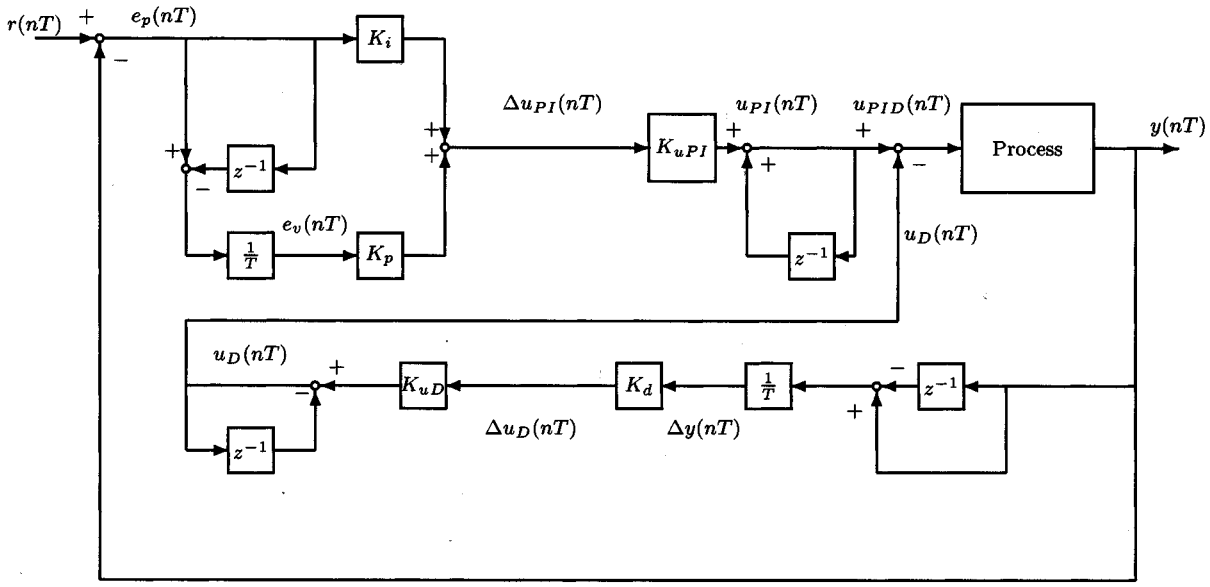

Fig. 2. Conventional digital PI+D control system.

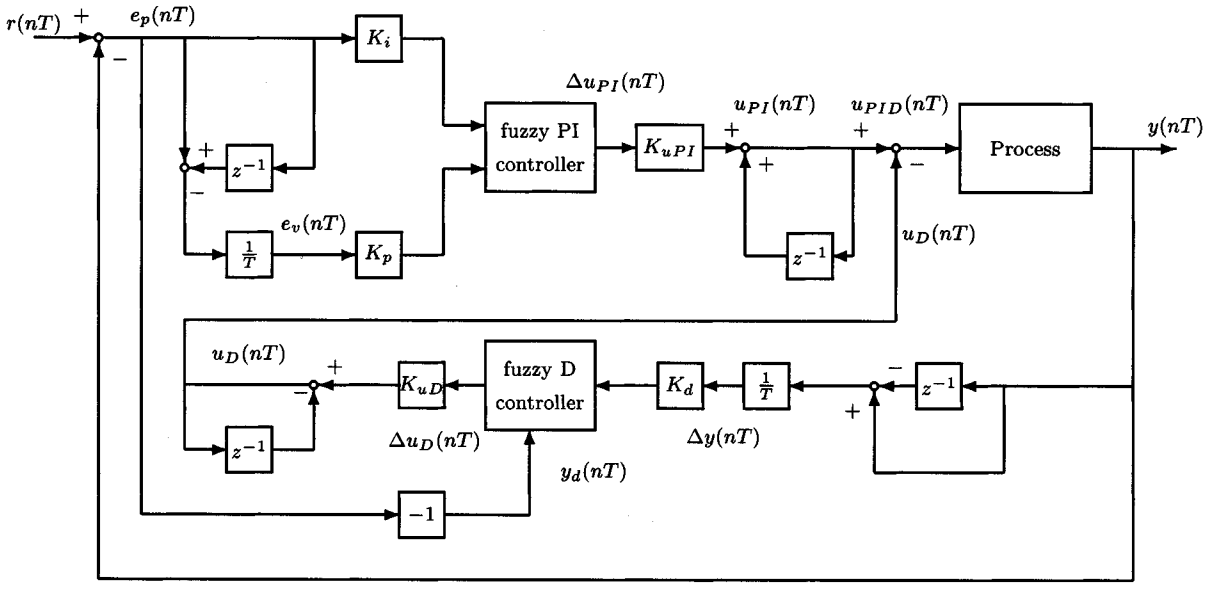

Fig. 3. Fuzzy PI+D control system.

III. Fuzzification, Control Rule Base, and Defuzzification

The fuzzy PID controller was designed by following the standard procedure of fuzzy controllers design, which consists of fuzzification, control rule base establishment, and defuzzification.

A. Fuzzification

We fuzzify the PI and D components of the PI+D control system individually and then combine the desired fuzzy control rules for each of them, taking into consideration the overall PI+D fuzzy control law given in (16). The input and output membership functions of the PI component are shown in Fig. 4(a) and (b), respectively, while those for the D component, in Fig. 5(a) and (b), respectively.

The fuzzy PI controller employs two inputs, the error signal $\widetilde{e_p}(nT) = K_i e_p(nT)$ with $e_p := r - y$ and the rate of change of the error signal $\widetilde{e_v}(nT) = K_p e_v(nT)$ with $e_v = \dot{e}_p = 0 - \dot{y} = -\dot{y}$ and has a single output $\Delta u_{\rm PI}(nT)$, as shown in Fig. 4, where the constant $L > 0$.

Similarly, the fuzzy D controller has two weighted inputs $y_d(nT)$ and $\widetilde{\Delta y}(nT) = K_d \Delta y(nT)$ and its output is denoted as $\Delta u_{\rm D}(nT)$.

It should be noted that a single constant L is used in these membership functions since the inputs and outputs will be weighted by the gains K_p, K_i, K_d, $K_{u\rm PI}$, and $K_{u\rm D}$, where the

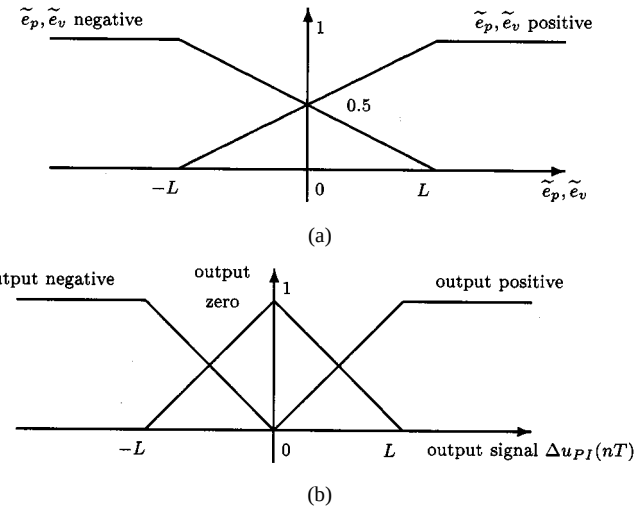

Fig. 4. Membership functions for the PI component. (a) Input membership functions. (b) Output membership functions.

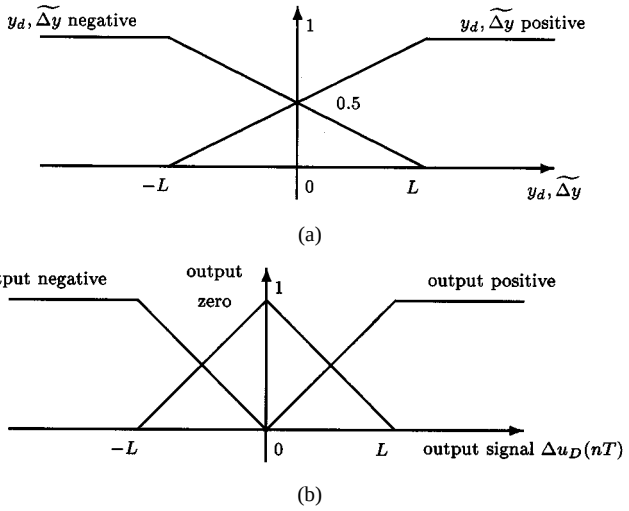

Fig. 5. Membership functions for the D component. (a) Input membership functions. (b) Output membership functions.

gains and the constant $L > 0$ are determined by MOGA later in specific applications.

B. Fuzzy Control Rules

Using the aforementioned membership functions, the following control rules are established for the fuzzy PI controller.

(R1) IF $\widetilde{e_p} = \widetilde{e_p} \cdot n$ AND $\widetilde{e_v} = \widetilde{e_v} \cdot n$, THEN PI output $= o \cdot n$.
(R2) IF $\widetilde{e_p} = \widetilde{e_p} \cdot n$ AND $\widetilde{e_v} = \widetilde{e_v} \cdot p$, THEN PI output $= o \cdot z$.
(R3) IF $\widetilde{e_p} = \widetilde{e_p} \cdot p$ AND $\widetilde{e_v} = \widetilde{e_v} \cdot n$, THEN PI output $= o \cdot z$.
(R4) IF $\widetilde{e_p} = \widetilde{e_p} \cdot p$ AND $\widetilde{e_v} = \widetilde{e_v} \cdot p$, THEN PI output $= o \cdot p$.

"PI output" is the fuzzy PI control output $\Delta u_{\rm PI}(nT)$, "$\widetilde{e_p} \cdot p$" means "error positive," and "$o \cdot p$" means "output positive," etc. Also, the logical "AND" takes the minimum.

Similarly, from the membership functions of the fuzzy D controller, the following control rules are used for the D component.

(R5) IF $y_d = y_d \cdot p$ AND $\widetilde{\Delta y} = \widetilde{\Delta y} \cdot p$, THEN D output $= o \cdot z$.
(R6) IF $y_d = y_d \cdot p$ AND $\widetilde{\Delta y} = \widetilde{\Delta y} \cdot n$, THEN D output $= o \cdot p$.
(R7) IF $y_d = y_d \cdot n$ AND $\widetilde{\Delta y} = \widetilde{\Delta y} \cdot p$, THEN D output $= o \cdot n$.
(R8) IF $y_d = y_d \cdot n$ AND $\widetilde{\Delta y} = \widetilde{\Delta y} \cdot n$, THEN D output $= o \cdot z$.

In the above rules, "D output" is the fuzzy D control output $\Delta u_{\rm D}(nT)$, and the other terms are defined similarly to the PI component.

These eight rules altogether yield the control actions for the fuzzy PI+D control law.

The formulation of these rules can be understood as follows. If we look at Rule 1 (R1) for the PI controller, condition $\widetilde{e_p} \cdot n$ (the error is negative) implies that the systems output y is above the setpoint, and $\widetilde{e_v} \cdot n$ (rate of error negative) implies $\dot{y} > 0$ (meaning that the controller at the previous step is driving the system output upward). Since the $\Delta u_{\rm PI}(nT)$ component of (16) contains more control terms with gain parameters than the D controller, we set this term to be negative and set the $\Delta u_{\rm D}(nT)$ component to be zero. Thus, the combined control action will drive the system output downward by Rules (R1) and (R5) of both controllers. Rules 2, 3, and 4 are similarly determined.

C. Defuzzification

In the defuzzification step, for both fuzzy PI and D controllers, the centroid formula is employed to defuzzify the incremental control of the fuzzy control law (16) as shown in (17) at the bottom of the page.

For the fuzzy PI controller, the value ranges of the two inputs, the error and the rate of change of the error, are actually decomposed into 20 adjacent input-combination (IC) regions, as shown in Fig. 6(a). This figure is understood as follows. We put the membership function of the error signal [given by the curves for $\widetilde{e_p}$ in Fig. 4(a)] over the horizontal axis in Fig. 6(a), and put the membership function of the rate of change of the error signal [given by the same curves in Fig. 4(a) for $\widetilde{e_v}$] over the vertical axis in Fig. 6(a). These two membership functions then overlap and form the third-dimensional picture [which is not shown in Fig. 6(a)] over the two-dimensional regions shown in Fig. 6(a). When we look at region IC1, for example, if we look upward to the $K_i e_p(nT)$ -axis, we see the domain $[0, L]$ and the membership function (in the third dimension) over $[0, L]$ of the error signal; if we look leftward to the $K_p e_v(nT)$ axis, we see the domain $[-L, 0]$ and the membership function (in the third dimension) over $[-L, 0]$ of the rate of change of the error signal.

The control rules for the fuzzy PI controller [(R1)–(R4)], with membership functions and IC regions together, are used to evaluate appropriate fuzzy control laws for each region.

In so doing, we consider the locations of the error $K_i e_p(nT)$ and the rate $K_p e_v(nT)$ in the regions IC1 and IC2 [see Fig. 6(a)]. Let us look at region IC1, for example, where $K_i e_p(nT)$ is in the range $[0, L]$ and $K_p e_v(nT)$ in $[-L, 0]$. For

$$\Delta u(nT) = \frac{\sum \text{membership value of input} \times \text{output corresponding to the membership value of input}}{\sum \text{membership value of input}} \tag{17}$$

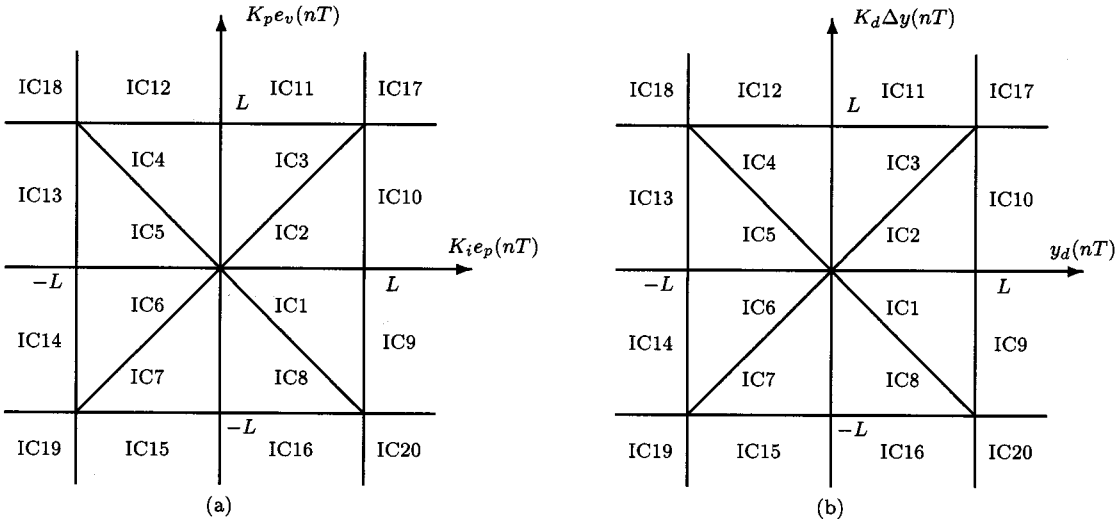

Fig. 6. Regions of (a) fuzzy PI and (b) fuzzy D controller input-combination values.

these two signals, we have $\widetilde{e_v}\cdot n > 0.5 > \widetilde{e_p}\cdot n$ [see Fig. 4(a)]. Hence, Fig. 4(a) and (R1), where the logical "AND" is used, together lead to

$$\{\text{"error} = \widetilde{e_p}\cdot n \text{ AND rate} = \widetilde{e_v}\cdot n\text{"}\} = \min\{\widetilde{e_p}\cdot n, \widetilde{e_v}\cdot n\} = \widetilde{e_p}\cdot n.$$

Therefore, Rule 1 (R1) yields

(R1) $\begin{cases}\text{the selected input membership value is } \widetilde{e_p}\cdot n;\\ \text{the corresponding output membership value is } o\cdot n.\end{cases}$

Similarly, in region IC1, Rules 2–4, (R2)–(R4), and the logical "AND" used in (R2)–(R4) together yield

(R2) $\begin{cases}\text{the selected input membership value is } \widetilde{e_p}\cdot n;\\ \text{the corresponding output membership value is } o\cdot z.\end{cases}$

(R3) $\begin{cases}\text{the selected input membership value is } \widetilde{e_v}\cdot n;\\ \text{the corresponding output membership value is } o\cdot z.\end{cases}$

(R4) $\begin{cases}\text{the selected input membership value is } \widetilde{e_v}\cdot p;\\ \text{the corresponding output membership value is} o\cdot p.\end{cases}$

It can be verified that the above are true for the two regions IC1 and IC2. Thus, in regions IC1 and IC2, it follows from the defuzzification formula (17) that

$$\Delta u_{\rm PI}(nT) = \frac{\widetilde{e_p}\cdot n\times o\cdot n + \widetilde{e_p}\cdot n\times o\cdot z + \widetilde{e_v}\cdot n\times o\cdot z + \widetilde{e_v}\cdot p\times o\cdot p}{\widetilde{e_p}\cdot n + \widetilde{e_p}\cdot n + \widetilde{e_v}\cdot n + \widetilde{e_v}\cdot p}.$$

To this end, by applying $o\cdot p = L$, $o\cdot n = -L$, $o\cdot z = 0$ [obtained from Fig. 4(b)], and the following straightline formulas from the geometry of the membership functions associated with Fig. 6(a):

$$\widetilde{e_p}\cdot p = \frac{K_ie_p(nT)+L}{2L} \quad \widetilde{e_p}\cdot n = \frac{-K_ie_p(nT)+L}{2L}$$
$$\widetilde{e_v}\cdot p = \frac{K_pe_v(nT)+L}{2L} \quad \widetilde{e_v}\cdot n = \frac{-K_pe_v(nT)+L}{2L}$$

we obtain

$$\Delta u_{\rm PI}(nT) = \frac{L}{2(2L - K_ie_p(nT))}\,[K_ie_p(nT) + K_pe_v(nT)].$$

Here, we note that $e_p(nT)\geq 0$ in regions IC1 and IC2. In the same way, one can verify that in regions IC5 and IC6 we have

$$\Delta u_{\rm PI}(nT) = \frac{L}{2(2L + K_ie_p(nT))}\,[K_ie_p(nT) + K_pe_v(nT)]$$

where it should be noted that $e_p(nT)\leq 0$ in regions IC5 and IC6. Hence, by combining the above two formulas we arrive at the following result for the four regions IC1, IC2, IC5, and IC6:

$$\Delta u_{\rm PI}(nT) = \frac{L[K_ie_p(nT) + K_pe_v(nT)]}{2(2L - K_i|e_p(nT)|)}.$$

Working through all regions in the same way, we obtain the following formulas for the 20 IC regions:

$$\Delta u_{\rm PI}(nT) = \begin{cases}\dfrac{L[K_ie_p(nT)+K_pe_v(nT)]}{2(2L-K_i|e_p(nT)|)} & \text{IC 1, 2, 5, 6}\\ \dfrac{L[K_ie_p(nT)+K_pe_v(nT)]}{2(2L-K_p|e_v(nT)|)} & \text{in IC 3, 4, 7, 8}\\ \frac{1}{2}\,[K_pe_v(nT)+L] & \text{in IC9, 10}\\ \frac{1}{2}\,[K_ie_p(nT)+L] & \text{in IC11, 12}\\ \frac{1}{2}\,[K_pe_v(nT)-L] & \text{in IC13, 14}\\ \frac{1}{2}\,[K_ie_p(nT)-L] & \text{in IC15, 16}\\ 0 & \text{in IC18, 20}\\ L & \text{in IC17}\\ -L & \text{in IC19.}\end{cases} \tag{18}$$

Similarly, defuzzification of the fuzzy D controller follows the same procedure as described above for the PI component, except that the input signals in this case are different. The IC combinations of these two inputs are decomposed into twenty similar regions, as shown in Fig. 6(b).

Similarly, we obtain the following formulas for the D controller in the 20 IC regions:

$$\Delta u_{\mathrm{D}}(nT) = \begin{cases} \dfrac{L[Ky_d(nT) - K_d\Delta y(nT)]}{2(2L - |y_d(nT)|)} & \text{in IC 1, 2, 5, 6} \\ \dfrac{L[Ky_d(nT) - K_d\Delta y(nT)]}{2(2L - K_d|\Delta y(nT)|)} & \text{in IC 3, 4, 7, 8} \\ \frac{1}{2}\left[-K_d\Delta y(nT) + L\right] & \text{in IC9, 10} \\ \frac{1}{2}\left[y_d(nT) - L\right] & \text{in IC11, 12} \\ \frac{1}{2}\left[-K_d\Delta y(nT) - L\right] & \text{in IC13, 14} \\ \frac{1}{2}\left[y_d(nT) + L\right] & \text{in IC15, 16} \\ 0 & \text{in IC17, 19} \\ -L & \text{in IC18} \\ L & \text{in IC20.} \end{cases} \tag{19}$$

An important final remark is that the stability of this fuzzy PI+D controller has been thoroughly analyzed, with sufficient conditions derived, in [8], [21].

IV. Optimization of Control Gains Based on GA

The basic principles of GAs were first proposed by Holland [15]. The GA is inspired by the mechanism of natural selection, where stronger individuals would likely be the winners in a competing environment. Here, GA uses a direct analogy of such natural evolution.

The GA presumes that a potential solution of a problem is an individual and can be represented by a set of parameters. These parameters are regarded as the genes of a chromosome. A positive value, generally known as the fitness value, is used to reflect the degree of "goodness" of the chromosome for the problem that would be highly related with its objective value.

Throughout a genetic evolution, the fitter chromosome has the tendency to yield good quality offspring, which means a better solution to the problem. Initially, a population pool of chromosomes is randomly installed. In each cycle of genetic operation termed as evolving process, a group of those chromosomes, generally called "parents" or a collection term "mating pool," are selected via a specific fitness proportionate selection routine. The genes of the parents are to be mixed and recombined for the production of offspring in the next generation. It is expected from this process of evolution (manipulation of genes) that the "better" chromosome will create a larger number of offspring, and thus has a higher chance of surviving in the subsequent generation, emulating the survival-of-the-fittest mechanism in nature.

The cycle of evolution is repeated until a desired termination criterion is reached. This criterion can also be set by the number of evolution cycles (computational runs), or the amount of variation of individuals between different generations, or a predefined value of fitness.

Detailed design of the GA can be referred to in [13], [19], and [20].

A. Chromosome Representation

Referring to Section III, there are seven control parameters $\{K_p, K_i, K_d, L, K_{u\mathrm{D}}, K_{u\mathrm{PI}}, K\}$ to be determined for an optimal fuzzy PI+D controller. Hence, the chromosome I can be defined as

$$I = \{K_p, K_i, K_d, L, K_{u\mathrm{D}}, K_{u\mathrm{PI}}, K\} \tag{20}$$

with real-number representation.

B. Genetic Operations

The specialized genetic operations developed in GENOCOP [20] for real-number-represented chromosome are adopted.

For crossover, the jth gene of the offspring I' can be determined by

$$I'_j = \beta I_j^{(x)} + (1 - \beta) I_j^{(y)} \tag{21}$$

where $\beta \in [0, 1]$ are uniformly distributed random numbers, $I^{(x)}$ and $I^{(y)}$ are selected parents.

Mutation is performed within the confined region of the chromosome by applying Gaussian noise to the genes [20].

C. Objective Functions

For the general control problem, it is desirable to optimize a number of different system performances.

Consider a step input $r(t)$ and the output response $y(t)$. The following objectives are stated for our design.

1) Minimizing the maximum overshoot of the output

$$O_1 = \max_t y(t). \tag{22}$$

2) Minimizing the settling time of the output

$$O_2 = t_s \tag{23}$$

such that $0.98r \le y(t) \le 1.02r, \forall t \ge t_s$.

3) Minimizing the rise time of the output

$$O_3 = t_r = t_1 - t_2 \tag{24}$$

such that $y(t_1) = 0.1r$ and $y(t_2) = 0.9r$.

D. Pareto-Based Fitness Assignment

Instead of aggregating the objectives with a weighting function, a multiobjective approach [11] is applied.

Definition: For an n-objective minimization problem, u is dominated by v if

$$\forall i = 1, \ldots, n, \quad f_i(u) \ge f_i(v)$$

and

$$\exists j = 1, \ldots, n, \quad \text{s.t.} \quad f_j(u) > f_j(v). \tag{25}$$

The chromosome I can then be ranked with

$$\text{rank}(I) = 1 + p \tag{26}$$

if I is dominated by other p chromosomes in the population. Hence, a Pareto-based fitness can be assigned to each chromosome according to its rank in the population.

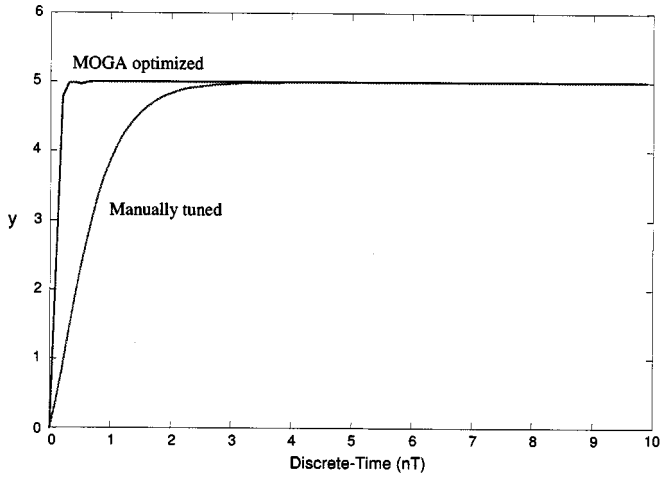

Fig. 7. Output response of the fuzzy PID controller.

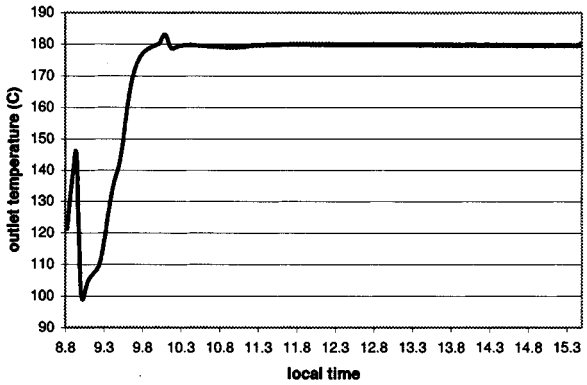

Fig. 8. Output response and reference.

Pareto-based ranking can correctly assign all nondominated chromosomes with the same fitness. However, the Pareto set may not be uniformly sampled. Usually, the finite populations will converge to only one or some of these, due to stochastic errors in the selection process. Such phenomenon is known as genetic drift. Therefore, fitness sharing [14] is adopted to prevent the drift and promote the sampling of the whole Pareto set by the population. The individual is penalized due to the presence of other individuals in its neighborhood. The number of neighbors governed by their mutual distance in objective spaces is counted and the raw fitness value of the individual is then weighted by this niche count. Eventually, the total fitness in the population is redistributed favoring those regions with less chromosomes located in them.

V. Results

A. Nonlinear Model

The examples chosen here for simulation and comparison are taken from [21], where they were simulated and compared to the classical PID controllers.

The first example is a nonlinear process with the following simple model:

$$\dot{y}(t) = 0.0001|y(t)| + u_{\mathrm{PID}}(t)$$

for which fuzzy PI+D parameters were determined manually as: $T = 0.1$, $K = 1.0$, $K_p = 1.5$, $K_d = 1.0$, $K_i = 2.0$, $K_{u\mathrm{PI}} = 1.1$, $K_{u\mathrm{D}} = 1.0$, $L = 45.0$, and the set oint $r = 5.0$ in order to obtain similar responses to that given in [21].

The output response obtained is shown in Fig. 7, where it clearly reveals that the fuzzy PI+D controller tracks the setpoint without any oscillation or steady-state error. On the contrary, the conventional PI+D controller is not able to track the set oint, no matter how one changes its parameters [21].

Fig.7 also depicts the output performance of the optimal fuzzy PI+D controller, in which the gain is located by MOGA. The obtained parameters are $T = 0.1$, $K_p^* = 0.9090$, $K_i^* = 4.1396$, $K_d^* = 0.9355$, $K_{u\mathrm{D}}^* = 2.8976$, $K_{u\mathrm{PI}} = 4.5329$, nd $L^* = 11.0472$. It clearly reveals that the fuzzy PI+D controller, after optimization via the GA, has generally better steady-state and transient responses, due to the multiobjective optimal criteria formulated for the set-point tracking control tasks.

B. Solar Plant Process

A similar optimal fuzzy controller is designed for the solar plant model at Tabernas, in Almería, Spain [5], [12, Fig. 1]. This solar plant consists of 480 distributed solar ACUREX collectors arranged in 20 rows forming ten parallel loops. Each loop is about 172-m long. The collector uses parabolic mirrors to reflect solar radiation onto a pipe for heating up the oil inside while circulation. A sunlight tracking system is installed to drive the mirrors to revolve around the pipes to achieve a maximum of sun radiation. The cold inlet oil is pumped from the bottom of the storage tank and passes through the field inlet. The heated oil is then transferred to a storage tank for generating the electrical power. The system is provided with a three way valve located in the field outlet to allow the oil to be recycled in the field until its outlet temperature is adequately heated for entering into the top of the storage tank.

The most important objective of this control system is to maintain the outlet oil temperature at a desirable level in spite of disturbances, which may be caused by the changes of solar radiation level, mirror reflectivity, and/or inlet oil temperature.

The following solution set is selected for the control purpose:

$$K_p = 3.3004,\ K_i = 2.7463,\ K_d = 3.1141,\ K = 0.3077$$

$$L = 185.1126,\ K_{u\mathrm{PI}} = 0.0348,\ K_{u\mathrm{D}} = 1.2086.$$

Fig. 8 shows the outlet temperature of the controlled plant for a step set point of 180 °C. It can be observed that the output is well tracking the reference temperature with fluctuation less than 0.6 °C and the overshoot is just about 3 °C, even with a large variation on the solar radiation as shown in Fig. 9. The oil flow is also plotted in Fig. 10 for reference.

It should be noted that the abnormal response in the starting phase of the operation is mainly due to a number of factors.

- The initial temperature profile inside the tubes (including the interconnection tube between the tank and the point in which the inlet oil temperature sensor is placed) is unknown and it causes a wrong result in the numerical integration algorithm in the simulator.
- The oil flow is usually saturated to the minimum value in order to produce the maximum oil heating.

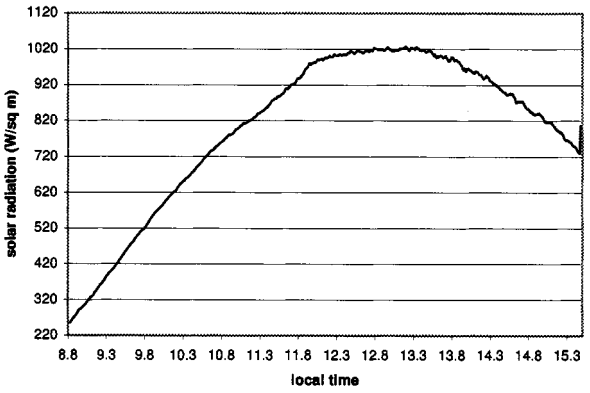

Fig. 9. Solar radiation.

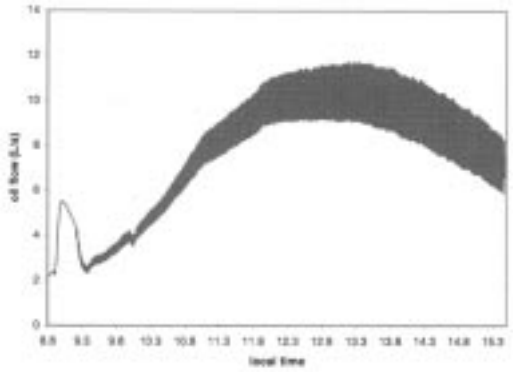

Fig. 10. Oil flow.

VI. Conclusion

An optimal fuzzy PID controller has been proposed in this paper. It is a discrete-time version of the conventional PID controller, with adaptive control capability and optimized via the multiobjective GA. The results demonstrate that its performance is much better than that of the one with manually tuned gains. This optimal fuzzy PID controller is suitable for the control of nonlinear plants in industrial applications, as demonstrated by the examples given in this paper.

References

[1] K. J. Aström, "Intelligent tuning," in *Adaptive Systems in Control and Signal Processing*, L. Dugard, M. M'Saad, and I. D. Landau, Eds. Oxford, U.K.: Pergamon, 1992, pp. 360–370.

[2] K. J. Aström, T. Hägglund, C. C. Hang, and W. K. Ho, "Automatic tuning and adaptation for PID controllers—A survey," in *Adaptive Systems in Control and Signal Processing*, L. Dugard, M. M'Saad, and I. D. Landau, Eds. Oxford, U.K.: Pergamon, 1992, pp. 371–376.

[3] K. J. Aström, C. C. Hang, P. Persson, and W. K. Ho, "Toward intelligent PID control," *Automatica*, vol. 28, pp. 1–9, 1992.

[4] S. Bennett, "Development of the PID controller," *IEEE Contr. Syst. Mag.*, vol. 13, pp. 58–65, Dec. 1993.

[5] E. F. Camacho, F. R. Rubio, and J. A. Gutierrez, "Modeling and simulation of a solar power plant system with a distributed collector system," in *Proc. Int. IFAC Symp. Power Systems Modeling and Control Applications*, 1988, pp. 11.3.1–11.3.5.

[6] J. Carvajal, G. Chen, and H. Ogmen, "Fuzzy PID controller: Design, performance evaluation, and stability analysis," *Inform. Sci.*, vol. 123, pp. 249–270, 2000.

[7] G. Chen, "Conventional and fuzzy PID controllers: An overview," *Int. J. Intell. Control Syst.*, vol. 1, pp. 235–246, 1996.

[8] G. Chen and T. T. Pham, *Introduction to Fuzzy Sets, Fuzzy Logic, and Fuzzy Control Systems*. Boca Raton, FL: CRC Press, 2000.

[9] G. Chen and H. Ying, "BIBO stability of nonlinear fuzzy PI control systems," *Int. J. Intell. Control Syst.*, vol. 5, pp. 3–21, 1997.

[10] H.-J. Cho, K.-B. Cho, and B.-H. Wang, "Fuzzy-PID hybrid control: Automatic rule generation using generic algorithms," *Int. J. Fuzzy Sets Syst.*, vol. 92, pp. 305–316, 1997.

[11] C. M. Fonseca and P. J. Fleming, "Multiobjective optimization and multiple constraint handling with evolutionary algorithms—Part I: A unified formulation," *IEEE Trans. Syst., Man, Cybern. A*, vol. 28, pp. 26–37, Jan. 1998.

[12] M. Geyer and B. Milow, "Annual technical report of Plataforma Solar de Almería," Plataforma Solar de Almería, Tabernas, Spain, Tech. Rep., 1995.

[13] D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*. New York: Addison-Wesley, 1989.

[14] D. E. Goldberg and J. Richardson, "Genetic algorithms with sharing for multimodal function optimization," in *Proceeding of the 2nd Int. Conference on Genetic Algorithms*, J. J. Grefenstette, Ed. Hillsdale, NJ: Lawrence Erlbaum, 1987, pp. 41–49.

[15] J. H. Holland, *Adaptation in Natural and Artificial Systems*. Cambridge, MA: MIT Press, 1975.

[16] H.-X. Li, "A comparative design and tuning for conventional fuzzy control," *IEEE Trans. Syst., Man, Cybern. B*, vol. 27, pp. 884–889, Oct. 1997.

[17] H. A. Malki, D. Feigenspan, D. Misir, and G. Chen, "Fuzzy PID control of a flexible-joint robot arm with uncertainties from time-varying loads," *IEEE Trans. Contr. Syst. Tech.*, vol. 5, pp. 371–378, May 1997.

[18] H. A. Malki, H. Li, and G. Chen, "New design and stability analysis of fuzzy proportional-derivative control systems," *IEEE Trans. Fuzzy Syst.*, vol. 2, pp. 245–254, Nov. 1994.

[19] K. F. Man, K. S. Tang, and S. Kwong, *Genetic Algorithms: Concepts and Designs*. London, U.K.: Springer-Verlag, 1999.

[20] Z. Michalewicz, *Genetic Algorithms + Data Structures = Evolution Program*, 3rd ed. New York: Springer-Verlag, 1996.

[21] D. Misir, H. A. Malki, and G. Chen, "Design and analysis of a fuzzy proportional-integral-derivative controller," *Int. J. Fuzzy Sets Syst.*, vol. 79, pp. 297–314, 1996.

[22] P. Sooraksa and G. Chen, "Mathematical modeling and fuzzy control of flexible robot arms," *Math. Comput. Model.*, vol. 27, pp. 73–93, 1998.

[23] W. M. Tang, G. Chen, and R. D. Lu, "A modified fuzzy PI controller for a flexible-joint robot arm with uncertainties," *Int. J. Fuzzy Sets Syst.*, vol. 118, pp. 109–119, 2001.

[24] W. A. Wolovich, *Automatic Control Systems*. Orlando, FL: Saunders College, 1994.

K. S. Tang (M'96) received the B.Sc. degree from the University of Hong Kong, Hong Kong, and the M.Sc. and Ph.D. degrees from the City University of Hong Kong, Hong Kong, in 1988, 1992, and 1996, respectively.

He is currently an Assistant Professor in the Department of Electronic Engineering, City University of Hong Kong. He is a member of the IFAC Technical Committee on Optimal Control (Evolutionary Optimization Algorithms) His research interests include evolutionary algorithms and chaotic theory.

Prof. Tang is a member of the Intelligent Systems Committee of the IEEE Industrial Electronics Society.

Kim Fung Man (M'91–SM'98) received the Ph.D. degree from Cranfield Institute of Technology, Bedford, U.K., in 1983.

He is currently with the Electronic Engineering Department, City University of Hong Kong, Hong Kong. He is the Editor-in-Chief of the *Real-Time Systems Journal* and a member of the IFAC Technical Committee on Algorithms and Architectures for Real-Time Control and Real-Time Programming. His research interests are genetic algorithms, chaos, and active noise control.

Dr. Man is currently an Associate Editor of the IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS.

Guanrong (Ron) Chen (M'89–SM'92–F'97) received the M.Sc. degree in computer science from Sun Yatsen (Zhongshan) University, Guangzhou, China, and the Ph.D. degree in applied mathematics from Texas A&M University, College Station.

He was a tenured Full Professor at the University of Houston, Houston, TX, prior to his appointment as a Chair Professor at the City University of Hong Kong, Hong Kong. His primary research interests are nonlinear systems control and dynamics. He is the coauthor of 180 refereed journal papers, 150 conference papers, and 11 research monographs and advanced textbooks, published since 1981. He is currently serving as Feature Editor, Advisory Editor, and Associate Editor of five international journals.

Prof. Chen is currently the Chair of the Nonlinear Circuits and Systems Technical Committee of the IEEE Circuits and Systems Society. Among the awards he has received is the 1998 Annual Prize for Outstanding Journal Paper from the American Society of Engineering Education.

Sam Kwong (M'93) received the B.Sc. degree in electrical engineering from the State University of New York, Buffalo, the M.A.Sc. degree in electrical engineering from the University of Waterloo, Waterloo, ON, Canada, and the Ph.D. degree from the University of Hagen, Hagen, Germany, in 1983, 1985, and 1996, respectively.

In 1990, he joined the City University of Hong Kong, Hong Kong, as a Lecturer in the Department of Electronic Engineering. He is currently an Associate Professor in the Department of Computer Science. His research interests are genetic algorithms, speech recognition, and data compression.