

PID Control Design with Exhaustive Dynamic Encoding Algorithm for Searches (eDEAS)

Jong-Wook Kim and Sang Woo Kim

Abstract: This paper proposes a simple but effective design method of PID control using a numerical optimization method. In order to achieve both stability and performance, gain and phase margins and performance indices of step response directly compose of the cost function. Hence, the proposed approach is a multiobjective optimization problem. The main effectiveness of this approach results from the strong capability of the used optimization method. A one-dimensional example concerning gain margin illustrates the practical applicability of the optimization method. The present approach has many degrees of freedom in controller design by only adjusting related weight constants. The attained PID controller is compared with Wang's and Ho's methods, IAE, and ISE for a high-order process, and the simulation result for various design targets shows that the proposed approach achieves desired time-domain performance with a guarantee of frequency-domain stability.

Keywords: Dynamic encoding algorithm for searches, gain tuning, multiobjective optimization, PID control.

1. INTRODUCTION

Proportional-integral-derivative (PID) controllers are widely used in the process control industry owing to their relatively simple structures and robust performances. This popularity has stimulated researchers to develop various methods for the design of PID controllers, which can be roughly classified into two approaches; frequency-based methods and time-based methods. The frequency-based tuning methods focused on system stability encompass most of the conventional methods, e.g., the Ziegler-Nichols rule [1], the Cohen-Coon method [2], the internal model control [3], Ho *et al.*'s tuning rules based on gain and phase margin specifications [4], Wang *et al.*'s method using a model reduction method [5], and so on. The time-based tuning methods concentrated on

control performance, which resort to the tools of soft computing, such as the fuzzy inference system, the neural network, and the genetic algorithm (GA) [6], are recently being researched in addition to the conventional criteria such as the integral absolute error (IAE) [7] and the integral time-weighted squared error (ITSE) [8].

For hybridization of the two design methods, a multiobjective design method was proposed for PID control of a linear brushless DC motor using GA [9]. The approach meets both the frequency-domain specifications directly related to the mixed sensitivity function and time-domain specifications such as rise time, maximum overshoot, and steady-state error of the step responses. However, checking feasibility of the sensitivity constraint for each chromosome makes GA slow with more complicated coding. To this end, the constrained optimization problem was transformed to an unconstrained one by a penalty function [10]. Gain and phase margins served as measures of stability, and desired output response was referred to for optimization of PID gains. Although the desired response was aimed to contain all the information of maximum overshoot, rise time and settling time at once, problems of adjusting dead time and desired rise time still remain.

This paper proposes a more direct and easy way to design PID controllers using the dynamic encoding algorithm for searches (DEAS). The cost function is composed of design errors for both frequency- and time-domain indices. That is, three gains of the PID controller are adjusted to follow desired stability

Manuscript received September 1, 2006; revised May 30, 2007; accepted September 28, 2007. Recommended by Editorial Board member Jietae Lee under the direction of Editor Jae Weon Choi. This research was supported by the MIC (Ministry of Information and Communication), Korea, under the ITRC (Information Technology Research Center) support program supervised by the IITA (Institute of Information Technology Advancement) (IITA-2006-(C1090-0602-0013)).

Jong-Wook Kim is with the Dept. of Electronics Engineering, Dong-A University, Busan 604-714, Korea (e-mail: kjwook@dau.ac.kr).

Sang Woo Kim is with the Electrical and Computer Engineering Division, Pohang University of Science and Technology, Pohang, Gyeongbuk 790-784, Korea (e-mail: swkim@postech.ac.kr).

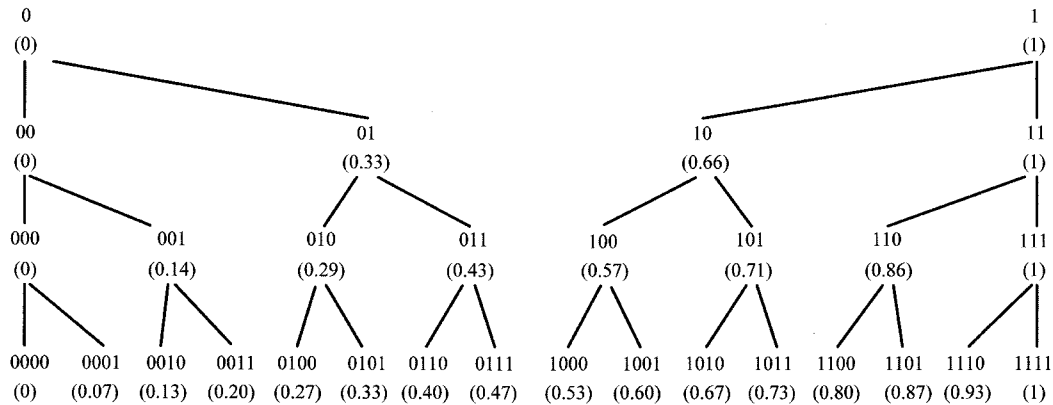


Fig. 1. Binary tree structure together with corresponding decoded values in parenthesis.

indices, i.e., gain and phase margins, and performance indices, i.e., maximum overshoot, rise time, and settling time in step response. Since this design approach requires neither model reduction nor analytical methods, engineers without specific knowledge about PID control are able to handle controllers intuitively. This approach is also to be used as a coarse or initial tuning of controllers. However, owing to the fact that derivative of the proposed cost function is quite hard or impossible to achieve, only computational optimization methods such as GA can deal with this problem at the cost of long execution time. If this computation time is reduced to a few microseconds with the aid of hardware, i.e., SoC (system-on-chip), and software, this design approach can be employed to PID autotuning or on-line tuning.

As a fast direct optimization method replacing GA, this work employs DEAS as a basic PID design tool. The cost function has five degrees of freedom in setting desired stability and performance by adjusting weight factors. The optimization performance of DEAS is compared with GA, and the proposed criterion is compared with those of Wang's and Ho's methods, IAE, and ISE. All the experiments show that the proposed design approach with DEAS gives better results.

Besides this three-dimensional optimization problem, DEAS is also applied to solving an implicit equation, which is a one-dimensional problem. The phase crossover frequency attained from this equation is used for calculation of gain margin. Considering that on-line computation of gain and phase margins are essential for adaptive control or auto-tuning, imprecise values calculated by function approximation [11,12] may deteriorate information on system stability. Both optimization results validate that DEAS is a useful numerical method irrespective of problem types. Source codes of DEAS will be partially released at the dedicated website (<http://deasgroup.net/>).

The paper is organized as follows: Section 2 provides a brief explanation of DEAS. Section 3

describes the proposed design method of PID control with simulation results. Section 4 concludes this work and discusses future work.

2. PRINCIPLES OF DEAS

DEAS is a global optimization method that possesses global and local search strategies. As the global search strategy, DEAS adopts the multistart method where local search is iterated from random points scattered over a search space. After conducting a finite number of local searches from selected random points, DEAS attains a global minimum that is the best local minimum. Since DEAS uses binary representation, i.e., search space is divided by finite grids, a random initial point is one of the intersection points of the grids. Therefore initial points should be checked if they were previously evaluated to avoid unnecessary cost evaluation. To this end, the routine named HISTORY CHECK is easily implemented in DEAS by concatenating all the rows in initial variable matrices into one string, storing them in memory, and comparing them with former strings [13]. The global optimization performance of HISTORY CHECK is quite strong for code simplicity.

The local search strategy in DEAS comprises bisectional search (BSS) and unidirectional search (UDS). BSS is derived from the property that insertion of 0 (or 1) right to LSB of a binary number leads to decrease (or increase) of transformed real values from that of the original binary number [13]. This phenomenon is adopted for local search; in the case the cost computed with a 0-added string is smaller than the cost of a 1-added string, one can obtain a better approximation of a local minimum by adding 0 to the current string as a result. However, the in-depth search of BSS should be balanced with UDS by extending it along an optimal BSS direction while maintaining string length.

For UDS, the simple operations of increment addition (INC) and decrement subtraction (DEC) for a binary string are carried out until a better optimum is

```

Randomly generate an initial binary matrix.
Check if the initial matrix has already been visited (HISTORY CHECK).
For  $m = \text{initLen} : \text{finLen}$ 
  BSS:
    Load the initial matrix or current best matrix attained in the previous UDS.
    Initialize a DV as an all-zero string.
    while  $i \leq 2^n$  do
      Add a current DV as a rightmost column to the current matrix.
      Decode the new matrix and evaluate its cost.
      if the evaluated cost is the lowest at this session then
        Save the matrix, the cost, and DV as optimal values.
        Check the termination condition.
      end if
      INC DV
    end while
  UDS:
    Load the optimal values of BSS as an initialization.
    while No better solution is obtained do
      Load the optimal matrix in a previous transition as a temporary matrix.
      Check the existence of extreme binary strings and exclude them as  $l = n - m$  (LIMIT CHECK).
      Initialize an EV as an all-zero string.
      for  $i \leq 2^l - 1$ 
        INC EV
        Check the redundancy of EV (REDUNDANCY CHECK).
        Conduct INC or DEC to a selected row according to the previous optimal DV.
        Decode the modified matrix and evaluate its cost.
        if the evaluated cost is the lowest then
          Save the matrix and the cost, and EV as optimal values.
          Check the termination condition.
        end if
      end for
    end while
    Check the restart condition.
    Check if the current best matrix has already been visited (HISTORY CHECK).
  end for

```

Fig. 2. Local search routine of eDEAS.

located. An example of UDS, which is started from 01 and preceded by BSS, is

$$011 \xrightarrow{UDS(1)} 100 \xrightarrow{UDS(2)} 101 \xrightarrow{UDS(3)} \dots$$

In this case BSS finds that INC is promising. In terms of directions, BSS searches vertically, while UDS searches horizontally as shown in Fig. 1, in which nodes are represented by binary strings and decoded real values.

For multi-dimensional problems, DEAS deals with binary matrices composed of binary rows for each variable. The number of neighborhood matrices generated during local search exponentially increases such as 2^n , where n is variable dimension. That is, each update requires 2^n times of cost evaluation for the initial version of DEAS, which is named exhaustive DEAS (eDEAS). Experiences evidence that problems of $n \geq 10$ are unfavorable to eDEAS. To this end, uDEAS was developed later to specifically reduce the order of neighborhood points from 2^n to $2n$ by carrying out BSS and UDS for each parameter [14].

The optimal design of PID control is interpreted as

attaining appropriate proportional, integral, and derivative gains whose controlled response suffices required specifications. Thus only three parameters should be optimized, which means that simple eDEAS can cope with this problem. The notion of 'simple eDEAS' implies that hopping or adding intelligence with history information [15,16] is not added to eDEAS.

Fig. 2 shows a brief pseudocode of eDEAS, where *initLen* and *finLen* represent predefined initial and final row lengths, respectively. In general, *initLen*=3 and *finLen*=10 are appropriate numbers. In the code, REDUNDANCY CHECK, which appears in-between UDS, plays a role to prevent UDS from probing points revisited by the previous UDS. The revisiting occurs at multidimensional problems from the fact that extension coordinates at each UDS are translated along optimal transitions. Fortunately, the REDUNDANCY CHECK is easily implemented by referring to the masking technique. See [13] for more details.

3. DESIGN OF PID CONTROL

3.1. Root finding problem

In this section, eDEAS is used for computing exact solutions of implicit equations hard to solve by analytical calculus. This also exemplifies supplementary use of eDEAS as a numerical routine. The ability of eDEAS to compute precise solutions within a few microseconds will be of interest to the engineers who have to derive complex equations but attain only approximate solutions.

For the first application, the PI controller and a first order lag plus time delay model (FOLPD) for process are respectively given by

$$G_c(s) = K_c(1 + \frac{1}{T_i s}), \tag{1}$$

$$G_m(s) = \frac{K_m e^{-sL}}{1 + s\tau}, \tag{2}$$

where K_c and T_i are the proportional gain and the integral time of the controller, respectively, and L and τ are the time delay and the time constant of the process model. Then

$$G_m(j\omega)G_c(j\omega) = \frac{K_m e^{-j\omega L} K_c(j\omega T_i + 1)}{1 + j\omega\tau} \cdot \frac{1}{j\omega T_i}. \tag{3}$$

From the definition of gain and phase margins, the following sets of equations are obtained for the unity feedback closed-loop system:

$$\phi_m = \arg[G_c(j\omega_g)G_m(j\omega_g)] + \pi, \tag{4}$$

$$A_m = \frac{1}{|G_c(j\omega_p)G_m(j\omega_p)|}, \tag{5}$$

where the frequency ω_p , at which the Nyquist curve has a phase of $-\pi$, is known as phase crossover frequency, and the frequency ω_g , at which the Nyquist curve has an amplitude of 1, is known as the gain crossover frequency. Thus ω_g and ω_p are given by

$$|G_c(j\omega_g)G_m(j\omega_g)| = 1 \tag{6}$$

$$\arg[G_c(j\omega_p)G_m(j\omega_p)] = -\pi. \tag{7}$$

From (3),

$$G_m(j\omega)G_c(j\omega) = \frac{K_m K_c \sqrt{1 + \omega^2 T_i^2}}{\omega T_i \sqrt{1 + \omega^2 \tau^2}} \tag{8}$$

$$\angle -0.5\pi + \tan^{-1} \omega T_i - \tan^{-1} \omega \tau - \omega L.$$

Therefore, from (4)

$$\phi_m = 0.5\pi + \tan^{-1} \omega_g T_i - \tan^{-1} \omega_g \tau - \omega_g L \tag{9}$$

with ω_g given by the solution of (6), i.e.,

$$\frac{K_m K_c \sqrt{1 + \omega_g^2 T_i^2}}{\omega_g T_i \sqrt{1 + \omega_g^2 \tau^2}} = 1. \tag{10}$$

Also, from (5) and (8),

$$A_m = \frac{1}{|G_c(j\omega_p)G_m(j\omega_p)|} = \frac{\omega_p T_i \sqrt{1 + \omega_p^2 \tau^2}}{K_m K_c \sqrt{1 + \omega_p^2 T_i^2}}. \tag{11}$$

With ω_p given by the solution of (7), i.e.,

$$0.5\pi + \tan^{-1} \omega_p T_i - \tan^{-1} \omega_p \tau - \omega_p L = 0. \tag{12}$$

From (10), ω_g is determined analytically to be

$$\omega_g = \sqrt{\frac{T_i(K_c^2 K_m^2 - 1) + \sqrt{(K_c^2 K_m^2 - 1)^2 T_i^2 + 4K_c^2 K_m^2 \tau^2}}{2T_i \tau^2}}. \tag{13}$$

Note that an analytical solution of (12) (to determine ω_p) is not possible because of the arctan function. Ho [11] attained an approximate analytical solution using the following approximate function for the arctan function

$$\tan^{-1} x \approx \frac{\pi}{4}x, |x| \leq 1 \text{ and } \tan^{-1} x \approx \frac{\pi}{2} - \frac{\pi}{4x}, |x| > 1. \tag{14}$$

Then four possibilities occur for ω_p :

- i) $\omega_p T_i > 1, \omega_p \tau > 1,$
- ii) $\omega_p T_i > 1, \omega_p \tau \leq 1,$
- iii) $\omega_p T_i \leq 1, \omega_p \tau > 1,$
- iv) $\omega_p T_i \leq 1, \omega_p \tau \leq 1.$

For the first case, solving for ω_p gives

$$\omega_p = \frac{\pi + \sqrt{\pi^2 - 4\pi L(\frac{1}{T_i} - \frac{1}{\tau})}}{4L}. \tag{15}$$

In the case that PID control or a second order process model is used, the number of possibilities of ω_p increases. Ho [12] shows that ω_p should be separately derived for twelve possibilities when the FOLPD model (2) is regulated with PID control

$$G_c(s) = K_c(1 + \frac{1}{T_i s})(\frac{sT_d + 1}{sT_d \alpha + 1}), \tag{16}$$

where T_d is the derivative gain and the constant α

is usually set between 0.05 and 2. Moreover, the equation for ω_p is slightly more complicated as

$$0.5\pi + \tan^{-1} \omega_p T_d + \tan^{-1} \omega_p T_i - \tan^{-1} \omega_p \tau - \tan^{-1} \omega_p \alpha T_d - \omega_p L = 0. \quad (17)$$

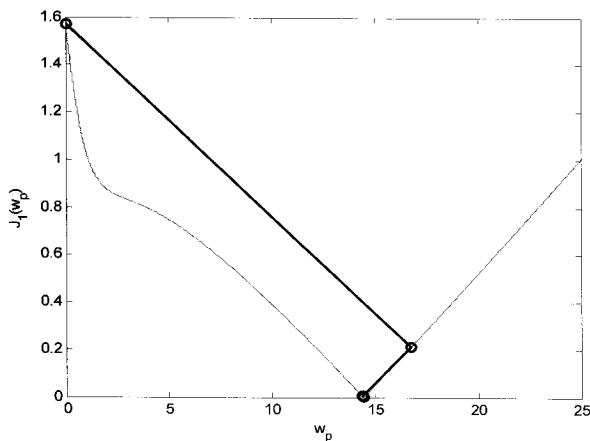
In order to find the solution ω_p^* for (12), the equation should be transformed into an absolute form such that

$$J_1(\omega_p) = |0.5\pi + \tan^{-1} \omega_p T_i - \tan^{-1} \omega_p \tau - \omega_p L|. \quad (18)$$

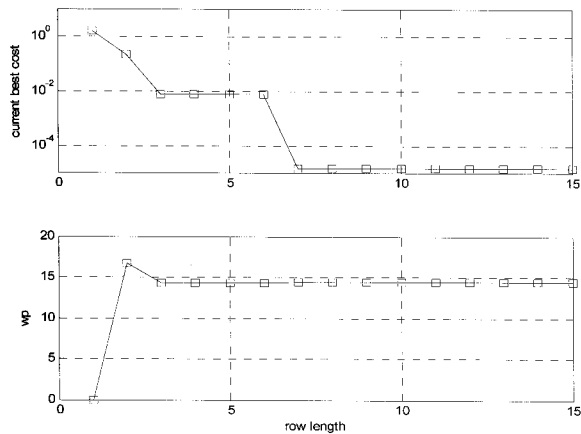
Then the ω_p that minimizes (18) is regarded as a solution. The controller gains and constants are arbitrarily given as

$$K_c = 4.91, T_i = 0.35, K_m = 1, \tau = 1, L = 0.1.$$

Fig. 3(a) presents that the absolute function (18) is unimodal and discontinuous at the solution ω_p^* ,



(a) Minimization of $J_1(\omega_p)$.



(b) Convergence aspect.

Fig. 3. Finding ω_p^* using eDEAS.

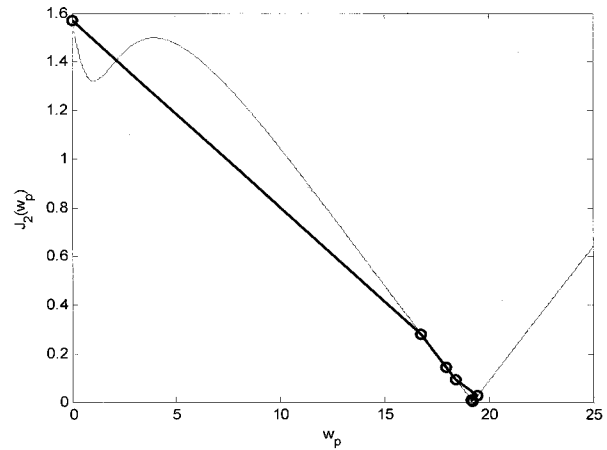


Fig. 4. Minimization of $J_2(\omega_p)$ using eDEAS.

where gradient methods fail to compute derivative. However, eDEAS can find ω_p^* from even the boundary point of 0 ignoring the discontinuity. This fast convergence is due to the unimodal shape of the cost function. The search points depicted with circles represent those attained at every session end. Fig. 3(b) indicates that eDEAS finds the solution $\omega_p^* = 14.4464$ with cost of 1.3791×10^{-5} at row length 7 and function evaluation number 19. Actually (12) yields error of $1.3682 \times 10^{-5} \cong 0$ for the ω_p^* attained. By Ho's method, however, (15) yields $\omega_p^* = 14.7169$ with error of -2.48×10^{-2} .

For the second application, the PID controller (16) with the FOLPD model is considered. Then, finding ω_p^* should minimize the cost function

$$J_2(\omega_p) = |0.5\pi + \tan^{-1} \omega_p T_d + \tan^{-1} \omega_p T_i - \tan^{-1} \omega_p \tau - \tan^{-1} \omega_p \alpha T_d - \omega_p L| \quad (19)$$

with $\alpha = 0.2, T_d = 0.4$. The other parameters and search conditions are the same.

Fig. 4 illustrates that eDEAS quickly finds a global optimum $\omega_p^* = 19.2244$ with error of 7.4397×10^{-5} at the row length 14 (function evaluation number is 43). It should be noted that the cost function has two local minima in this case. However, owing to the fact that a left-hand local minimum has small deviation from unimodal shape, a wide swing generated from setting the initial binary string length at 1 leads to a fast convergence to a global minimum. For the equations that have several solutions, eDEAS has to initiate local search from random strings whose lengths are more than 3 and iterate restart several times. The above results concerning root finding are acceptable in terms of the small function evaluation numbers and quite good accuracies.

3.2. PID control design

In this section, three PID gains are optimized in order to meet both stability and performance of a closed loop system. Gain and phase margins serve as stability measure, while overshoot, rise time, and settling time as performance measure. And for simplification these indices are combined into one cost function which is minimized by eDEAS. Thus this approach can be classified as multiobjective optimization.

Gain and phase margins have recommended values. That is, a closed loop system that has large margins will slow down system response, whereas small margins weaken system stability. Moderate values of 3dB and 60° are generally adopted for gain and phase margins, respectively [5]. On the contrary, the performance measure (overshoot, rise time, and settling time) can be specified arbitrarily by control designers. Considering that the three indices are related with each other, excessive fitting to a single variable may result in unwanted response. For example, in the case that only overshoot is designed to minimize, a system that responds very slowly may be attained. Therefore, three performance indices have to be considered together to properly describe desired response.

The PID controller designed in this section is given by

$$G_c(s) = K_p + K_i \frac{1}{s} + K_d s, \tag{20}$$

where K_p , K_i , and K_d are the proportional, integral, and derivative gains, respectively. The process is selected to be a third order plus time delay model

$$G_m(s) = \frac{1}{(s+1)(s+5)^2} e^{-0.5s}. \tag{21}$$

It should be mentioned that no model reduction is carried out to (21).

A new PID design rule is described with the following unconstrained optimization problem:

$$\begin{aligned} &\text{Minimize } J_3(K_p, K_i, K_d) \\ &= \alpha_1 \frac{|gm - GM^*|}{GM^*} + \alpha_2 \frac{|pm - PM^*|}{PM^*} + \beta_1 \frac{|os - OS^*|}{OS^*} \\ &\quad + \beta_2 \frac{|rt - RT^*|}{RT^*} + \beta_3 \frac{|st - ST^*|}{ST^*}, \end{aligned} \tag{22}$$

where gm , pm , os , rt , and st denote gain margin, phase margin, maximum overshoot, rise time, and settling time, respectively. os is the maximum overshoot defined as $y_{max} - y_{ss}$ where y_{ss} represents the steady state value of y , rt is the rise time defined as the time required for the step response

to rise from 10 to 90 percent of y_{ss} , and st is the settling time defined as the time required for the step response to decrease and stay within $\pm 1\%$ of y_{ss} . The capitalized terms with asterisks represent designed or specified values for corresponding terms. The two kinds of coefficients $\alpha_i, i=1,2$ and $\beta_i, i=1,2,3$ represent weights relevant to stability and performance, respectively. Since each index is normalized in (22), the weight factors play a role to magnify the one most desired.

For optimization, the target indices in (22) are specified as

- $GM^* = 3\text{dB}, PM^* = 60^\circ$
- $OS^* = 0.03, RT^* = 0.9\text{sec}, ST^* = 5\text{sec}$

with weight factors of

$$\alpha_1 = 1, \alpha_2 = 1, \beta_1 = 1, \beta_2 = 1, \beta_3 = 1. \tag{23}$$

Since practical problems have many local minima, eDEAS is run under the following search conditions:

- $initLen = 3$
- $finLen = 10$
- $restartNo = 5$.

where $restartNo$ means the number of restarts initiated from random points at the global search stage.

Fig. 5 illustrates that eDEAS minimizes design error (22) for the weight factor set of (23) that has four local minima during five restarts. One restart is ceased at the fourth row length by HISTORY CHECK. In the case that there exist many local minima, as in Fig. 5, a good strategy for global search is repeating restart as many times as possible. Fortunately, five times of restart was sufficient for our optimization problem. The average function evaluation number per restart is 163, and average error for four restarts is 0.5251 (best one is 0.4003). Therefore, it is evident that eDEAS reliably finds good local minima within short execution time.

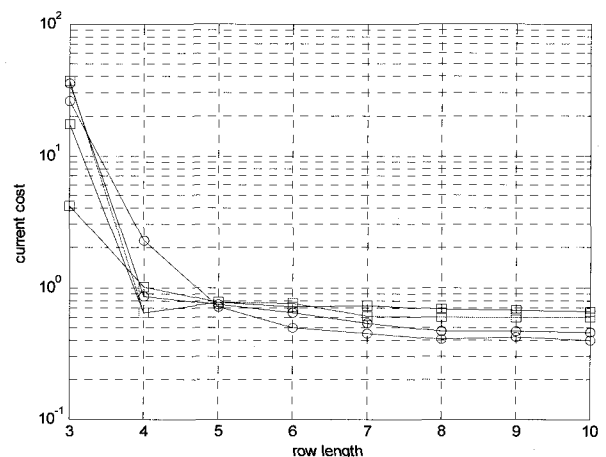


Fig. 5. Minimization of design error J_3 using eDEAS.

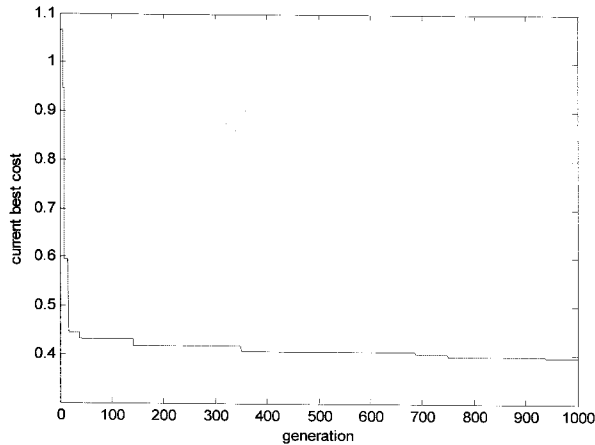
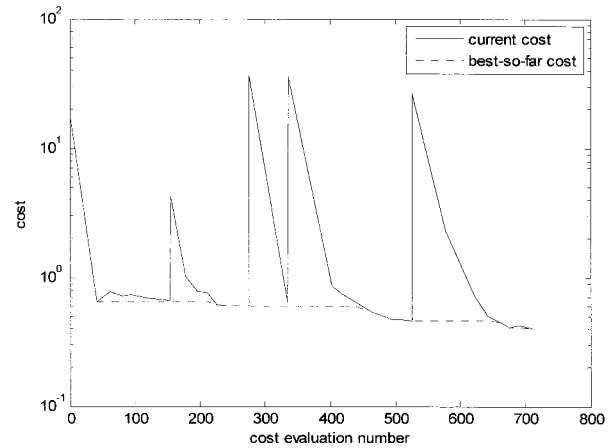


Fig. 6. Minimization of design error J_3 using RCGA.

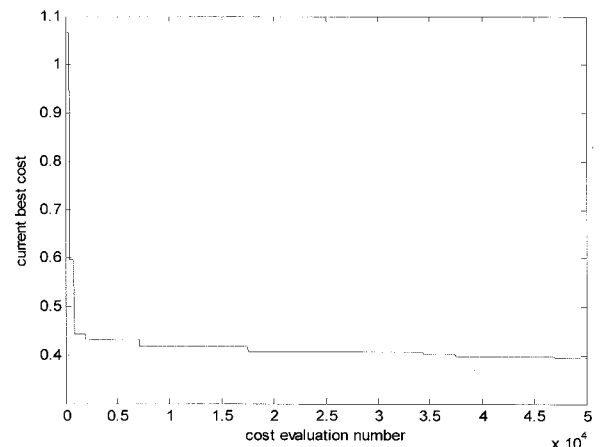
For optimization performance comparison, the real-coded genetic algorithm (RCGA) is used for minimizing the cost function J_3 with weight factors of (23). As is well-known, RCGA is more appropriate for optimizing real values than the binary-coded GA. The configuration of RCGA is such that roulette-wheel selection, modified simple crossover, dynamic mutation, and the elitism strategy are used for evolution. Population size and maximum generation are set to 50 and 1000, respectively, as usual. Simulation is carried out with the m-file codes provided in [17]. Fig. 6 shows that RCGA seems to minimize rapidly with respect to generation. Considering 50 times of J_3 evaluation is required at each generation, however, comparison with respect to the number of J_3 evaluation is fairer.

Fig. 7 compares Figs. 5 and 6 in terms of cost evaluation numbers that are proportional to computation time. The peaks that appear in Fig. 7(a) are due to restart from random binary matrices whose row lengths are shortened from $finLen$ to $initLen$. The third and fourth peaks are very close since at the third restart revisit has happened and has been protected by HISTORY CHECK. After five times of restart, eDEAS concludes that the globally optimal PID gains whose cost is 0.4003 are attained after a total of 712 times of J_3 evaluation. Fig. 7(b) shows that RCGA also searches the optimal PID gains after a much longer running time. The ratio of running time is such that RCGA consumed 749 generations, i.e., 37450 (=749×50) times of cost evaluation for decreasing J_3 below 0.4003, which means that it is 52.6 (=37450/712) times faster for eDEAS.

Fig. 8 validates the predominance of the attained PID controller over the other controllers. Simulation is carried out on Matlab using margin function and SIMULINK. eDEAS is also coded with m-files. The stability and performance indices attained with the designed controller are



(a) Minimization of J_3 with eDEAS.



(b) Minimization of J_3 with RCGA.

Fig. 7. Comparison of optimization time in terms of the number of cost function evaluation.

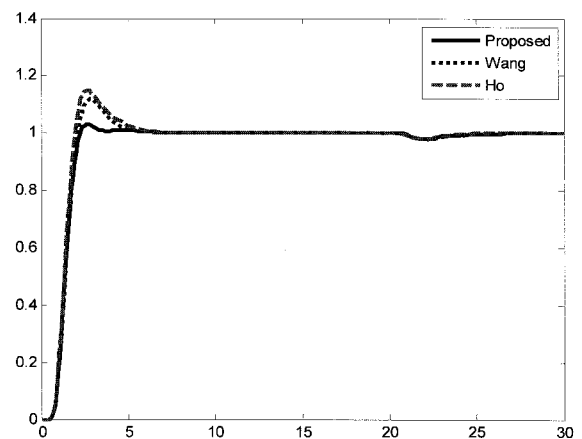


Fig. 8. Step response of the process $G_m(s) = \frac{1}{(s+1)(s+5)^2} e^{-0.5s}$. (Solid line: proposed, dotted line: Wang's, dashed line: Ho's).

- $K_p^* = 29.30, K_i^* = 18.94, K_d^* = 9.08$
- $gm = 2.89\text{dB}, pm = 70.39^\circ$

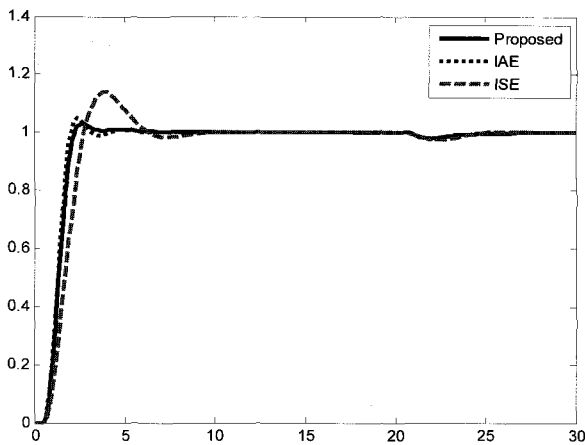


Fig. 9. Step response of the process $G_m(s)$ controlled by three PID controllers. (Solid line: proposed, dotted line: IAE, dashed line: ISE).

- $os = 0.03$, $rt = 1.07\text{sec}$, $st = 4.98\text{sec}$,

where the time-domain performance indices os , rt , and st are calculated using simulated output responses at every step of searching optimal PID gains. The gain and phase margins are automatically calculated by the margin function of Matlab owing to calculation complexity. In the case that Matlab is not available, the gain and phase margins can be computed by following the routines described in Section 3.1.

In the above result, the overshoot exactly matches with the desired value. The slight deviations of the other values arising from system characteristics are ignorable. In Fig. 8, overshoots are 0.12 for Wang's controller and 0.15 for Ho's controller. This superiority of the proposed design method that is simplest comes from the strong performance of eDEAS. The perturbation occurred at $t = 20$ is due to the load change exerted stepwise from 0 to -1 at that time.

Fig. 9 compares the controlled result of the proposed approach with those of PID controllers attained by the well-known IAE and ISE criteria. IAE denotes integral of absolute error as follows

$$\int_0^{\infty} |e(t)| dt,$$

and ISE represents integral of squared error such as

$$\int_0^{\infty} e^2(t) dt.$$

eDEAS minimized these criteria with the same search space and configuration. The attained gain margins for IAE and ISE are 2.62 and 1.95, and their overshoots are 0.05 and 0.18, respectively. The result of IAE that is often used for PID design [7] is slightly worse than that of eDEAS. Moreover, owing to its simplicity, a

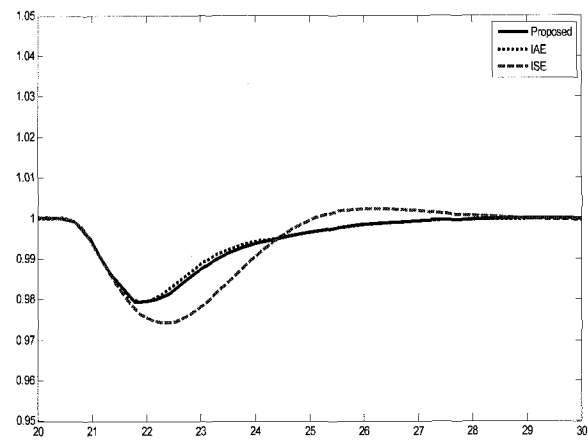


Fig. 10. Comparison of regulatory performance for three optimal PID gains. (Solid line: proposed, dotted line: IAE, dashed line: ISE).

more delicate tuning is impossible with IAE. Fig. 10 reveals regulatory performances of the three PID gains, which is an enlargement of the load perturbation part occurring at 20 seconds in Fig. 9. The PID controller designed with the proposed approach is as robust as that of IAE by virtue of stability guarantee.

To test design flexibility, weight factors are changed solely for single objectives and simultaneously for double objectives. Table 1 shows changes in weight factors and corresponding optimal PID gains and performance/stability indices, all of which were attained by eDEAS. As for the result of single objective design, all the corresponding indices except the phase margin precisely match with the desired ones. The deviation in the phase margin is due to the fact that the actual margin cannot be lowered below 65.24° with maintaining the other indices to desired values. Since 65.24° is acceptable for phase margin, the PID controller is not problematic.

As for the double objective design, the sampled two cases well validate the control design performance of the proposed approach. In the first case, the resultant gain margin and overshoot become exactly desired values by adjusting $\alpha_1 = 10$ and $\beta_1 = 10$ and optimizing with eDEAS, while in the second case $\beta_1 = 10$ and $\beta_2 = 10$ for overshoot and rise time.

What is interesting in Table 1 is that in every case overshoot exactly follows 0.03. This implies that if one aims to decrease only overshoot of step response, the desire may be met at the cost of deteriorating stability and the other performance indices. Moreover, if only performance indices are concerned as shown in the third, fourth, and the fifth rows of the simple objective part, gain margins decrease below 3.0. Therefore for balancing stability and performance

Table 1. Adjustment of weight factors and attained stability and performance indices using eDEAS. Desired indices are $gm^*=3\text{dB}$ for gain margin, $pm^*=60^\circ$ for phase margin, $os^*=0.03$ for overshoot, $rt^*=0.9\text{sec}$ for rise time, and $st^*=5\text{sec}$ for settling time.

	Stability weight factors		Performance weight factors			Optimal gains			Stability and performance indices attained by eDEAS				
	α_1	α_2	β_1	β_2	β_3	K_p^*	K_i^*	K_d^*	gm	pm	os	rt	st
Single objective	10	1	1	1	1	28.33	18.71	8.54	3.00	69.96	0.03	1.10	4.99
	1	10	1	1	1	20.16	15.64	2.19	3.44	65.24	0.03	1.47	4.35
	1	1	10	1	1	19.33	15.63	3.38	3.98	65.99	0.03	1.56	5.46
	1	1	1	10	1	35.10	20.44	14.08	2.24	74.42	0.03	0.90	5.27
	1	1	1	1	10	31.50	19.30	10.43	2.66	71.75	0.03	1.00	4.99
Double objective	10	1	10	1	1	28.42	18.01	7.44	3.00	69.98	0.03	1.10	4.33
	1	1	10	10	1	35.16	20.38	14.28	2.22	74.83	0.03	0.90	5.20

double objective optimization such as $\alpha_1=10$, $\alpha_2=1$, $\beta_1=10$, $\beta_2=1$, $\beta_3=1$ are recommended.

The overall simulation results show that eDEAS robustly finds optimal PID gains that satisfy the desired stability and performance indices of the closed-loop system.

4. CONCLUSIONS

This paper provides a new design method of PID control that makes a closed-loop system retain desired performance and sufficient stability. What is new for this approach is that stability and performance indices can be arbitrarily chosen within reasonable range, and eDEAS finds an optimal controller that fulfils the given indices. This method is straightforward and easy to apply in that no model reduction and no derivative is required. The present approach has five degrees of freedom, i.e., two for stability and three for set-point tracking, in PID control design owing to fast optimization of eDEAS. Thus, finer performance tuning is possible for existing PID controllers without sacrificing closed-loop stability. The complete design process is straightforward, and thus general operators can easily tune the existing PID controller considering operation condition change. A number of design experiments carried out by changing the weight constants confirms the above statements.

In this paper, eDEAS is employed as a background optimization method that minimizes multiobjective cost function. A one-dimensional problem of root finding is also tested with eDEAS, and its solutions are shown to be exact. Based on this success, PID autotuning using eDEAS will be studied with improvement of optimization efficiency.

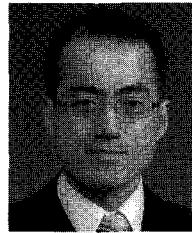
Future research concerning DEAS is oriented in developing more advanced series of DEAS with

applications to various research fields. The advanced DEAS series will adopt modular and randomized local search strategies. In addition, the properties of local and global optimality and convergence rate will be analyzed for them. Performance enhancement through the DEAS series will improve state-of-the-art technologies concerning intelligent robots, ubiquitous computing, and many biological and chemical optimization problems.

REFERENCES

- [1] J. G. Ziegler and N. B. Nichols, "Optimum settings for automatic controllers," *Trans. ASME*, vol. 64, pp. 759-768, 1942.
- [2] G. H. Cohen and G. A. Coon, "Theoretical consideration of retarded control," *Trans. ASME*, vol. 75, pp. 827-834, 1953.
- [3] K. J. Åström and T. Hägglund, *PID Controllers; Theory, Design, and Tuning*, International Society for Measurement and Control, 1995.
- [4] W. K. Ho, C. C. Hang, and L. S. Cao, "Tuning of PID controllers based on gain and phase margin specifications," *Automatica*, vol. 31, no. 3, pp. 497-502, 1995.
- [5] Q.-G. Wang, T.-H. Lee, H.-W. Fung, Q. Bi, and Y. Zhang, "PID tuning for improved performance," *IEEE Trans. Contr. Syst. Technol.*, vol. 7, no. 4, pp. 457-465, 1999.
- [6] D. E. Goldberg, *Genetic Algorithm in Search, Optimization and Machine Learning*, Addison Wesley, 1989.
- [7] F. G. Shinskey, *Process Control Systems, Application, Design, and Tuning*, McGraw-Hill, New York, 1988.
- [8] M. Zhuang and D. P. Atherton, "Automatic tuning of optimum PID controllers," *IEE Proc. Part D*, vol. 140, pp. 216-224, 1993.
- [9] C.-L. Lin, H.-Y. Jan, and N.-C. Shieh, "GA-

- based multiobjective PID control for a linear brushless DC motor," *IEEE/ASME Trans. on Mechatronics*, vol. 8, no. 1, pp. 56-65, March 2003.
- [10] J.-W. Kim and S. W. Kim, "Gain tuning of PID controllers with the dynamic encoding algorithm for searches (DEAS) based on the constrained optimization technique," *Proc. of International Conference on Control, Automation and Systems*, Gyeongju, Korea, pp. 871-876, October 2003.
- [11] W. K. Ho, C. C. Hang, and J. H. Zhou, "Performance and gain and phase margins of well-known PI tuning formulas," *IEEE Trans. Contr. Syst. Technol.*, vol. 3, no. 2, pp. 245-248, 1995.
- [12] W. K. Ho, O. P. Gan, E. B. Tay, and E. L. Ang, "Performance and gain and phase margins of well-known PID tuning formulas," *IEEE Trans. Contr. Syst. Technol.*, vol. 4, no. 4, pp. 473-477, 1996.
- [13] J.-W. Kim and S. W. Kim, "A numerical method for global optimization: Dynamic encoding algorithm for searches," *IEE Proc.-Control Theory and Appl.*, vol. 151, no. 5, pp. 661-668, Sept. 2004.
- [14] J.-W. Kim, N. G. Kim, and S. W. Kim, "On-load parameter identification of an induction motor using univariate dynamic encoding algorithm for searches," *Proc. of International Conference on Control, Automation and Systems*, Bangkok, Thailand, pp. 852-856, August 2004.
- [15] S.-C. Choi, N. G. Kim, J.-W. Kim, and S. W. Kim, "Improvement of dynamic encoding algorithm for searches (DEAS) using hopping unidirectional search (HUDS)," *Proc. of International Conference on Control, Automation, and Systems*, KINTEX, Korea, pp. 324-329, June 2005.
- [16] Y. S. Park, *Improvement of DEAS using history information*, Master Thesis, Pohang Univ. of Sci. and Tech., Feb. 2006.
- [17] G.-G. Jin, *Genetic Algorithms and Their Applications*, Gyo Woo Sa, 2000.



Jong-Wook Kim received the B.S., M.S., and Ph.D. degrees from the Electrical and Computer Engineering Division at Pohang University of Science and Technology (POSTECH), Pohang, Korea, in 1998, 2000, and 2004, respectively. Currently, he is an Assistant Professor in the Department of Electronics Engineering at Dong-A University, Busan, Korea. His current research interests are numerical optimization methods, robot control, intelligent control, diagnosis of electrical systems, and system identification.



Sang Woo Kim received the B.S., M.S., and Ph.D. degrees from the Department of Control and Instrumentation Engineering, Seoul National University, in 1983, 1985, and 1990, respectively. Currently, he is an Associate Professor in the Department of Electronics and Electrical Engineering at POSTECH, Pohang, Korea. He joined POSTECH in 1992 as an Assistant Professor and was a Visiting Fellow in the Department of Systems Engineering, Australian National University, Canberra, Australia, in 1993. His current research interests are in optimal control, optimization algorithms, intelligent control, wireless communication, and process automation.