Gain Tuning of PID Controllers with the Dynamic Encoding Algorithm for Searches (DEAS) Based on the Constrained Optimization Technique

Jong-Wook Kim and Sang Woo Kim

Electrical and Computer Engineering Division, POSTECH, Pohang, Kyungbuk, Korea
(Tel: +82-54-279-5018; Fax: +82-54-279-2903; E-mail:{kjwooks,swkim}@postech.ac.kr)

Abstract: This paper proposes a design method of PID controllers in the framework of a constrained optimization problem. Owing to the popularity for the controller's simplicity and robustness, a great deal of literature concerning PID control design has been published, which can be classified into frequency-based and time-based approaches. However, both approaches have to be considered together for a designed PID control to work well with a guaranteed closed-loop stability. For this purpose, a penalty function is formulated to satisfy both frequency- and time-domain specifications, and is minimized by a recent nonlinear optimization algorithm to attain optimal PID control gains. The proposed method is compared with Wang's and Ho's methods on a suite of example systems. Simulation results show that the PID control tuned by the proposed method improves time-domain performance without deteriorating closed-loop stability.

Keywords: PID control, gain tuning, Dynamic Encoding Algorithm for Searches, penalty function

1. Introduction

The 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 and time-based methods. The frequency-based tuning methods encompass most of the conventional methods, e.g. the Ziegler-Nichols rule, the Cohen-Coon method, the internal mode control, Wang's and Ho's tuning rules, and so on. These methods generally guarantee the stability of the controlled systems, but often design poorly tuned PID controllers in terms of time-domain specifications. On the other hand, the time-based tuning methods which resort to the tools of soft computing, such as the fuzzy inference system, the neural network, and the genetic algorithm, are recently being researched in addition to the conventional criteria such as IAE and ITAE. However, despite their excellent control performance on a time domain, the time-based methods often lack the guarantee of the system's stability. Therefore, both the frequency-based and time-based approaches have to be considered together for determining the gains of stable and well-performing PID controllers. This paper presents a constrained optimization technique in which frequency-domain related factors (i.e. phase and gain margins) serve as a constraint, and the three gains enabling the output response to well agree with a desired response are sought by a nonlinear optimization algorithm.The desired output response retains all the required characteristics in terms of a rise time, an overshoot, and a settling time. This reduces the difficulties of determining the weights between the time-domain factors in constituting a cost function to be minimized.

The dynamic encoding algorithm for searches (DEAS) is a recently developed nonlinear optimization method verified on several benchmark functions [1] and on the parameter identification of a simple RLC circuit [2] and an induction motor [3]. DEAS is a type of a greedy search on a discrete structure, and thus consumes an exponential computation amount as the number of parameters increases. However, owing to the fact that the PID controllers require only three gains to be optimized, DEAS is suitable to search high quality parameters within a short CPU time. The output responses of the proposed tuning method are compared with those of Wang's and Ho's methods [4][5] via computer simulations on a suite of example systems. Simulation results show that the PID control tuned by the proposed method improves time-domain performance without deteriorating closed-loop stability.

The paper is organized as follows: Section 2 describes an overall principle of the proposed tuning method. Section 3 provides a brief explanation of DEAS. Section 4 presents simulation results for three example systems. Section 5 concludes the work and discusses future work.

2. Problem Formulation

The proposed design rule is formulated as solving the following constrained optimization problem as

$$\text{Minimize} \quad \int_0^\infty |y(t) - y_d(t)| dt \tag{1}$$

subject to

$$GM \geq g_0 \quad \text{and} \quad PM \geq p_0 \tag{2}$$

where $y(t)$ and $y_d(t)$ denote controlled and desired step responses, respectively, and GM and PM represent gain and phase margins, respectively. The terms g_0 and p_0 are gain and phase specifications which guarantee system stability.

Penalty function methods transform the constrained optimization problem into alternative formulations such that numerical solutions are sought by solving an unconstrained optimization problem [6]. The penalty function formulations for the inequality constrained problems can be divided into two categories: interior and exterior methods. In the interior methods, the penalty term is not defined for infeasible solutions. Thus, it requires a feasible starting point for the search toward the optimum point. The exterior penalty

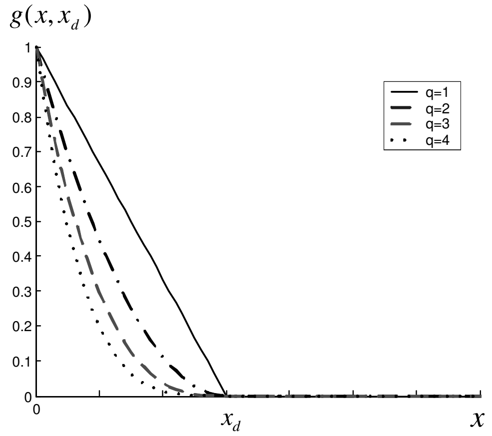

Fig. 1. Normalized constraint function

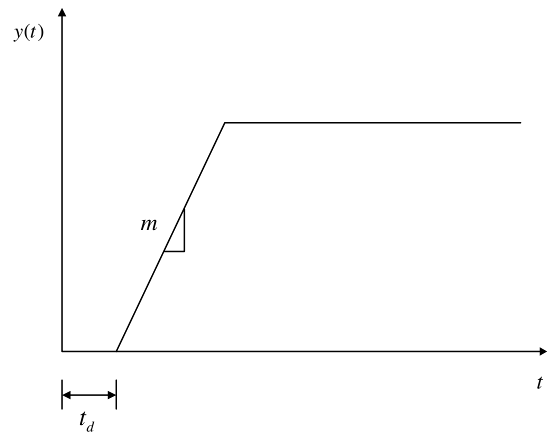

Fig. 2. Desired step response

method, however, is free from such a requirement, and is more suitable for the PID control design. Using the exterior penalty method, the above constrained optimization problem is converted into solving the following penalty function

$$\phi = \int_0^\infty |y(t) - y_d(t)|dt + r\{g(GM, g_0)^q + g(PM, p_0)^q\} \quad (3)$$

where r is a positive penalty parameter, the exponent q is a nonnegative constant, and the constraint function is defined as

$$g(x, x_d) = \max(\frac{x_d - x}{x_d}, 0). \quad (4)$$

Fig. 1 plots the profiles of the constraint function $g(x, x_d)^q$ under the change of the exponent q. Since x is normalized by the desired value x_d within $[0, x_d]$, the term $(x_d - x)/x_d$ in (4) ranges from 0 to 1. As q increases, the values of $((x_d - x)/x_d)^q$ gradually decrease, which leads to weakening the penalty for violating the constraint. Hence, q is set to be 1, and the other parameter r is adjusted for acceptable solutions.

The desired step response $y_d(t)$ in (3) is designed to meet the expected time-domain specifications such as a rise time, a settling time, and maximum overshoot. Since the time-domain specifications are usually used as the final measure of system performance, a control designer will feel familiar with, for example, the requirement that the maximum overshoot should be less than 1 percent and a settling time less than 2 sec. However, owing to the fact that control parameters usually interact with each other and influence design specifications in conflicting ways, the design process is in general complicated despite many useful guidelines as in [7]. This difficulty has led to the application of recent optimization techniques, such as the genetic algorithm, to the PID control design with the following performance indices

$$\begin{aligned} ISE &= \int_0^T |r(t) - y(t)|^2 dt \\ IAE &= \int_0^T |r(t) - y(t)| dt \quad (5) \\ ITAE &= \int_0^T t|r(t) - y(t)| dt \end{aligned}$$

where $r(t)$ being the reference input and $y(t)$ the output of the system [8]. However, the performance indices are obscure in terms of the time-domain specifications. That is, though one wants the rise time of a closed-loop step response to be 1 sec., there's no relevant factor in (5). Therefore, a need for a desired step response in which design values of the time-domain specifications can be reflected is risen in applying the optimization methods. The desired step response can be attained by the transient response of a prototype second-order system and the relation between the time- and frequency-domain factors [7]. However, this is another complicated task unwanted for the designer. For those difficulties, this paper presents a simple desired response as

$$y_d(t) = \begin{cases} 0, & t < t_d \\ m(t - t_d), & t_d \leq t < t_d + \frac{1}{m} \\ 1, & \text{otherwise} \end{cases} \quad (6)$$

where $m = 0.8/t_r^d$ is the rising ratio derived from the definition of rise time. In (6), the only adjusting factor for design specifications is the desired rise time t_r^d, while the time delay t_d is system dependent. This makes sense since the maximum overshoot and settling time in general have to be reduced as much as possible as shown in Fig. 2 which is the profile of (6). It is to be noted that the unit step input $r(t)$ in (5) is a special form of $y_d(t)$ with $t_d = 0, t_r^d = 0$. In this paper, the penalty function (3) is minimized by DEAS.

3. DEAS

The basic search operators of DEAS are bisectional search (BSS) and unidirectional search (UDS). BSS is based on the real-valued relation of binary strings [2] which is adopted for nonlinear local optimization. Adding 0 or 1 to a current binary string is interpreted as probing search space with smaller or larger values from the current minimum. Thus in the case the cost of a zero-added string is smaller than that of a one-added string, one can obtain the better approximation of a local minimum by adding 0 to the current string in the sequel, and vice versa. The name 'bisectional search' comes from this dichotomous search strategy.

However, BSS alone has the serious drawback of a regional limitation. If the length of an initial string is k, the ratio

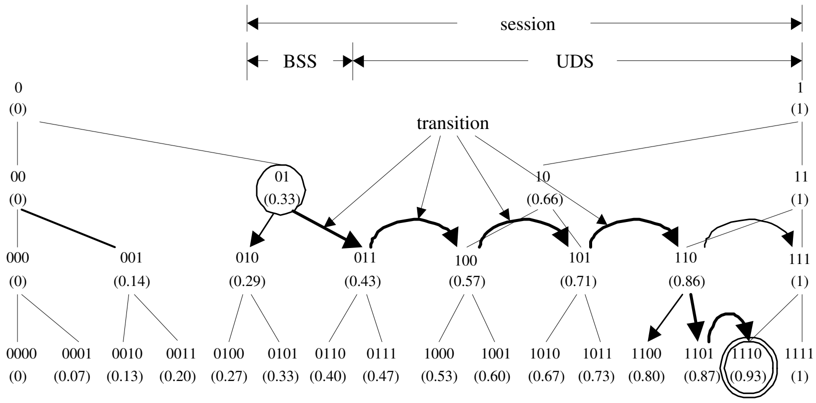

Fig. 3. Local search aspect of DEAS in a one-dimensional problem along with related terms

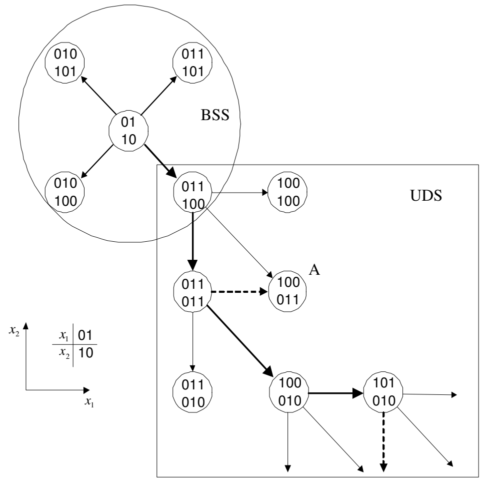

Fig. 4. Search aspect of DEAS in a two-dimensional problem (solid: basic direction, thick and solid: selected direction, dash-dotted: redundant direction)

of reachable search space by BSS is $1/2^k$ whose maximum is 0.5. The branch and bound method, which employs a search strategy similar with BSS but differs in that 0 and 1 represent divided search space, is incapable to overcome this problem. However, the simple operations of increment addition (INC) and decrement subtraction (DEC) for a binary string can readily remove the barrier between any binary trees. In UDS, these operations are carried out until a better optimum is located in a guided direction obtained by BSS. As an illustration of BSS and UDS, assume that a one-dimensional optimization problem whose cost function is unimodal and has a local and global minimum at 1110b (about 0.93), is addressed. In case 01b (about 0.33) is selected as an initial string, the series of optimal strings will proceed as follows:

$$01 \overset{BSS}{\rightarrow} 011 \overset{UDS(1)}{\rightarrow} 100 \overset{UDS(2)}{\rightarrow} 101 \overset{UDS(3)}{\rightarrow} 110 \overset{BSS}{\rightarrow} 1101 \overset{UDS(1)}{\rightarrow} 1110. \quad (7)$$

In (7), only two sessions (one session = one BSS + multiple UDS) of BSS and UDS and 9 function evaluations, where additional evaluations are consumed in each BSS, are required to locate the local minimum as shown in Fig. 3. If a local minimum has to be encoded with a longer binary string, more cost evaluation will be required.

In handling multi-dimensional problems, the binary strings of DEAS is extended to binary matrices. Fig. 4 illustrates a session in a two-dimensional problem whose local minimum exists in a lower right part. For an optimal direction to be discovered, BSS evaluates the cost function four, or 2^2, times by adding 0 or 1 to a given matrix, and $[1\ 0]^T$ is determined as an optimal direction vector (DV) of this session, which is handed over to UDS for further exploration. Then UDS continues to search in three, or $2^2 - 1$, fixed directions per transition by extension vectors (EV) while better solutions are sought. For notational convenience, a binary digit 0 of each EV means that no extension occurs in a given direction, and vice versa. As shown in Fig. 4, an optimal EV for the first transition in UDS is determined as $[0\ 1]^T$, since the current best solution is obtained by only moving along the direction of x_2.

It is to be noted that, for the second transition, the matrix A is evaluated again, which also recurs for the fourth transition. This phenomenon is related with the previous optimal EVs depicted with thick solid lines in Fig. 4. Since every extension coordinate is shifted along the direction of an optimal EV, the neighborhood matrix with no valid extension for the optimal EV will revisit already sought points. Therefore, the revisit has to be prevented in advance. Fortunately, it is quite simple to implement the redundancy check in UDS resorting the masking technique

1	0	0	Previous optimal EV
X	X	X	Current EV
X	0	0	Masked result

If all the bits represented as 'X' in the masked result are simultaneously zero, corresponding EV is regarded as a redundant search direction with the protection for cost evaluation. For example, revisiting EVs for the previous optimal EV of $[1\ 0\ 0]^T$ are $[0\ 0\ 1]^T$, $[0\ 1\ 0]^T$, and $[0\ 1\ 1]^T$ except $[0\ 0\ 0]^T$ which is basically excluded in UDS.

For more efficient search, the current version of DEAS includes subroutines for a limit check and a history check. They are described in detail in [1] and in the dedicated website (www.deasgroup.net). The readers can take the codes

```
Randomly generate an initial binary matrix.
Check if the initial matrix has already been visited (HISTORY CHECK).
For m = initLen : 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 ≤ 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 in 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 initializtion.
    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 ≤ 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 condtion.
    Check if the current best matrix has already been visited
    (HISTORY CHECK).
end for
```

Fig. 5. Algorithm of DEAS

written in C/C^{++} and Matlab by sending an E-mail to the author.

Fig. 5 shows a brief pseudocode of DEAS, where *initLen* and *finLen* represent predefined initial and final row lengths, respectively. The row length is directly associated with the resolution of the corresponding decoded real value, thus the final row length of 15 yields the solutions of sufficient resolution.

For a global search strategy, the current version employs the multistart algorithm, where each random point is taken as a starting point for local optimization and the process is iterated until a termination condition is met. Many local searches of the multistart algorithm may lead to the same local minima [9]. However, by the aid of the history check, DEAS iterates the procedure of Fig. 5 without revisit and locates the global or near-global minimum whose cost function is lowest among all the local minima found. This global strategy with DEAS were tested for several benchmark functions and verified its effectiveness [1].

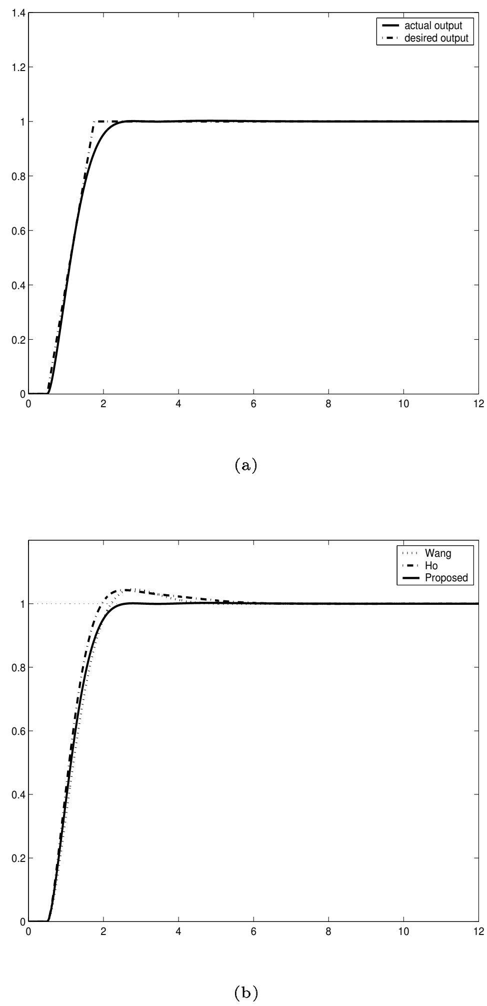

Fig. 6. Step response of the process $G(s) = \frac{1}{(s+1)(s+5)^2}e^{-0.5s}$ (a) desired response against actual response (b) performance comparison with Wang's and Ho's methods

4. Simulation Examples

This section presents some examples for the proposed design technique. Comparisons will be made with Wang's and Ho's gain and phase margin methods (GPM) [4][5]. The gain and phase specifications are set at 3 and 60°.

The first example system is a nonoscillatory high-order process

$$G(s) = \frac{1}{(s+1)(s+5)^2}e^{-0.5s}. \qquad (8)$$

Wang's method requires a second-order with time-delay model which approximates the real system by matching them at two nonzero frequency points. Then, the pole-zero cancellation technique is employed for the design process. However, the proposed tuning method requires only adjusting the penalty parameter r in (3) and the desired rise time t_r^d or m in (6). Since $r \to \infty$ enables the unconstrained minima to converge to a feasible region, a large number is recommended for r. The desired rise time t_r^d relevant to a controlled plant

Table 1. Simulation results under the variation of the penalty parameter r
(OS: overshoot in percentage, ST: settling time in seconds, RT: rising time in seconds)

	PID gains			Performance				
r	K_p	K_i	K_d	GM	PM	OS	ST	RT
0.1	29.8456	21.6987	8.7466	2.8592	63.8725	0.84	2.1	1.02
1.0	29.7845	21.6620	8.7161	2.8652	63.8915	0.812	2.1	1.02
10.0	29.6365	21.5888	8.6581	2.8797	63.9305	0.739	2.12	1.03
100.0	28.4555	20.9540	8.2491	3.00	64.4085	0.257	2.33	1.1

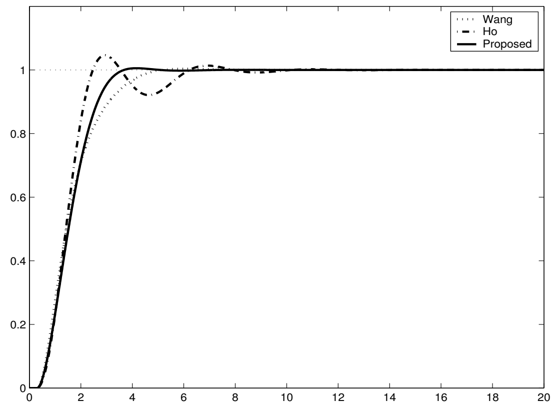

Fig. 7. Step responses of the process $G(s) = \frac{1}{(s^2+2s+3)(s+3)}e^{-0.3s}$

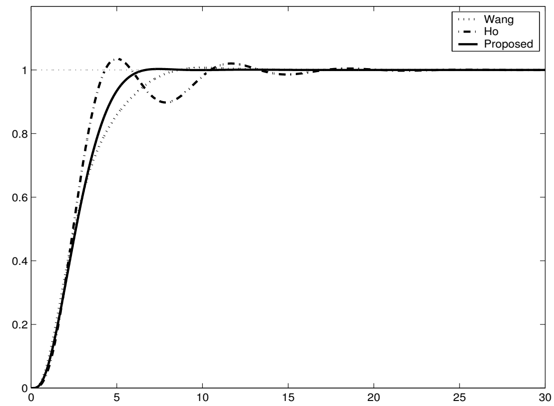

Fig. 8. Step responses of the process $G(s) = \frac{1}{(s^2+s+1)(s+2)^2}e^{-0.1s}$

is an approximated guideline for controlled output. Therefore, unless the desired rise time is set to be far from the actual values, DEAS will find the gains that commands the closed-loop step response to follow the given profile. The rule of thumb for (8) is $t_r^d = 1$. Table 1 tabulates optimal PID control gains and the corresponding specifications for given penalty parameters. Note that, as r increases, the gain margin gradually reaches the specification value of 3, while the rising time departs from the desired value of 1.0. Moreover, the overshoot is largely reduced for increasing r with sufficient phase margin. Therefore, the control gains for $r = 100$ are acceptable for their improved time-domain performance with guaranteed stability. Fig. 6 shows that the PID controller attained by the proposed method for $r = 100$ coincides the desired response elegantly and outperforms the other controllers designed by Wang's and Ho's methods.

The second example system is a high-order and moderately oscillatory process given by

$$G(s) = \frac{1}{(s^2+2s+3)(s+3)}e^{-0.3s}. \tag{9}$$

The only changes for this system is that $t_d = 0.3$, $t_r^d = 2$. Fig. 7 shows the step responses of the controllers. The proposed method results in an improved response with the smallest overshoot and the shortest settling time.

The third example system is another high-order and heavily oscillatory process with multiple lag given by

$$G(s) = \frac{1}{(s^2+s+1)(s+2)^2}e^{-0.1s}. \tag{10}$$

For this system, $t_d = 0.1$, $t_r^d = 4$. Fig. 8 shows that the closed-loop performance of the proposed controller is better than those of the other methods with no oscillation.

5. Conclusions

This paper presents a compromising tuning rule between time- and frequency- domain requirements. By the help of the robust and fast optimization algorithm such as DEAS, the attained PID control gains make actual output responses match well with desired output responses that can be modified in an easy way, and guarantee the closed-loop stability by maintaining specified gain and phase margins.

The future work is focused on adding an uncertainty constraint to the penalty function in order to reinforce the robustness of PID controllers.

References

[1] J. -W. Kim and S. W. Kim, "A numerical method for global optimization: dynamic encoding algorithm for searches," submitted to *IEE Proc. -Control Theory Appl.*

[2] J. -W. Kim and S. W. Kim, "A novel parametric identification method using dynamic encoding algorithm for searches (DEAS)," *International Conference on Control, Automation and Systems*, pp. 406–411, Muju, Korea, 2002.

[3] J. -W. Kim and S. W. Kim, "Parameter identification of induction motors using dynamic encoding algorithm for

searches (DEAS)," submitted to *IEEE Trans. Energy Conversion*.

[4] 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.

[5] 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.

[6] S. S. Rao, *Engineering Optimization*, John Wiley & Sons, Inc., 1996.

[7] B. C. Kuo, *Automatic Control Systems*, Prentice-Hall, Inc., 1995.

[8] P. Cominos and N. Munro, "PID controllers: recent tuning methods and design to specification," *IEE Proc. -Control Theory Appl.*, Vol. 149, No. 1, pp. 46–53, 2002.

[9] A. Törn and A. Žilinskas, *Global Optimization*, Springer-Verlag & Berlin, 1989.