We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

6,900
Open access books available

186,000
International authors and editors

200M
Downloads

Our authors are among the

154
Countries delivered to

TOP 1%
most cited scientists

12.2%
Contributors from top 500 universities

WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com

Chapter

Design of the Second-Order Controller by Time-Domain Objective Functions Using Cuckoo Search

Huey-Yang Horng

Abstract

The proportional-integral-derivative (PID) controllers are widely used in many industrial control applications. However, the lead-lag controllers are more practical. Traditionally, time-domain or frequency-domain methods have been used to design a lead-lag controller in order to meet the design specifications. This chapter will focus on the design of controller by optimizing the time-domain objective function. The proposed objective function includes the first peak time, maximum peak time, rise time, maximum overshoot, maximum undershoot, setting time, and steady-state error. In the study, cuckoo search algorithm is adopted to search the optimal controller parameters. Cuckoo search is a recently developed meta-heuristic optimization method, which is a population-based algorithm inspired by the behavior of some cuckoo species in combination with the Lévy flight behavior. A numerical example is simulated to illustrate the use of the proposed method.

Keywords: PID controller, lead-lag controller, controller design, cuckoo search

1. Introduction

In many industrial control applications, proportional-integral-derivative (PID) controllers are probably the most commonly used controllers. Several methods were proposed in the past for tuning the PID controller parameters [1–5]. It is noted the lead-lag controllers provide a more practical alternative. Tan used the Kharitonov and the Hermite-Biehler theorems to compute the parameters of lead-lag controller in [6]. Kuo et al. designed the lead-lag compensator based on vector margin and steady-state error of the step response [7]. Horng used cuckoo search to design lead-lag controllers [8]. It is easy to see that the lead-lag controller is just a special case of the general second-order controllers. Therefore, this paper extends the lead-lag controllers to general second-order controllers. The goal of this paper is to propose a simple second-order controller design procedure using the cuckoo search.

The control design specifications may usually be divided into time-domain and frequency-domain specifications. Time-domain specifications (*TDS*) include the lower and/or upper bounds of the quantities of the time response such as the first peak time, maximum peak time, rise time, maximum overshoot, maximum undershoot, setting time, and steady-state error. Frequency-domain specifications are

IntechOpen

usually given in terms of the resonant peak, phase margin, resonant frequency, and bandwidth. In this study, we only consider the time-domain specifications.

The main contribution of this study is that we provide a simple controller design procedure for simple second-order controllers. For the problem formulation, some of the time-domain specifications (e.g., desired peak time and maximum overshoot) can be used to fully define a desired simple second-order reference model. With this reference model, we may reasonably specify lower and/or upper bounds for other time-domain specifications. Finally, we define the deviation ratios (i.e., percentage errors) and total deviation ratio. The total deviation ratio, which is the weighted sum of the deviation ratios, will be used as the objective function for the problem. Zero objective value means that all specifications are satisfied. However, there is no guarantee of the zero objective value in the most general cases. Some specifications might actually be violated. The design goal is to choose the best controller parameters so that the objective value is as close to zero as possible.

To search for optimal controller parameters, some optimizer must be employed to solve the aforementioned optimization problem. Since there is usually no analytic formula for the objective function, evolutionary computation algorithms are well suited in this situation to solve this optimization problem. In this study, the meta-heuristic cuckoo search algorithm is adopted to search the optimal controller parameters.

Cuckoo search algorithm is one of the latest meta-heuristic techniques, developed by Yang and Deb [9, 10]. Nowadays, implementations of cuckoo search has proved to be effective in engineering optimization problems [11–13], for example, optimal power system stabilizers [14], load frequency controller design [15], optimal power system [16], synthesis of analog controller [17], etc. Cuckoo search achieved better results than available methods in most cases which appeared in the literature.

The novelty of this study is that the whole controller design problem can be formulated as an optimization problem by considering most important time-domain performance indices as a whole. Moreover, the meta-heuristic cuckoo search algorithm (or any other powerful evolutionary computation algorithms like genetic algorithm or particle swarm optimization) can be adopted to search for the best controller parameters. A numerical example will be provided to illustrate the design process. To show the wide applicability of the proposed method, four different plants and three different time-domain specifications will be used in the illustrative example.

2. Time-domain analysis of control systems

The time response of a control system is typically divided into two parts: the transient response and the steady-state response. The steady-state response is just the part of the total response which remains after the transient has died out. Hence, the steady-state response can still vary in a fixed pattern, such as a ramp function or a parabola function that increases with time.

The underdamped second-order system is a familiar model for physical problems. The detailed understanding of the underdamped response is necessary for both analysis and design. Let us begin by describing the step response for the second-order system. The transfer function of an underdamped second-order system is given by

$$C(s) = \frac{\omega_n^2}{s^2 + 2\xi\omega_n s + \omega_n^2}, \quad 0 < \xi < 1. \tag{1}$$

A typical unit-step response is shown in **Figure 1**. In this figure, y_{ss}, y_M, and y_m denote the steady-state value, maximum response value, and the response value where the maximum undershoot occurs, respectively. Moreover, T_r, T_p, and T_s are the rise time, peak time, and settling time, respectively.

We now describe seven time-domain specifications (*TDS*) used in objective function in more detail:

1. Rise time T_r. It is the time required for the step response to rise from 10 to 90% of its final value.

2. First peak time T_f. It is the time to reach the first peak.

3. Maximum peak time T_p. It is the time to reach the maximum peak.

4. Maximum overshoot *MOS*. This is defined as

$$MOS = \begin{cases} \dfrac{y_M - y_{ss}}{y_{ss}}, & \text{if } y_M > y_{ss}, \\ 0, & \text{if } y_M \leq y_{ss}. \end{cases} \tag{2}$$

5. Maximum undershoot *MUS*. It is defined as

$$y_m = \min(y(t)), \quad t \geq T_f,$$

$$MUS = \begin{cases} \dfrac{y_{ss} - y_m}{y_{ss}}, & \text{if } y_m < y_{ss}, \\ 0, & \text{if } y_m \geq y_{ss}. \end{cases} \tag{3}$$

6. Settling time T_s. This is the time required for the step response to decrease and stay within a specified ±2% of the final value.

7. Steady-state error E_{ss}. It is the difference between the desired and actual responses.

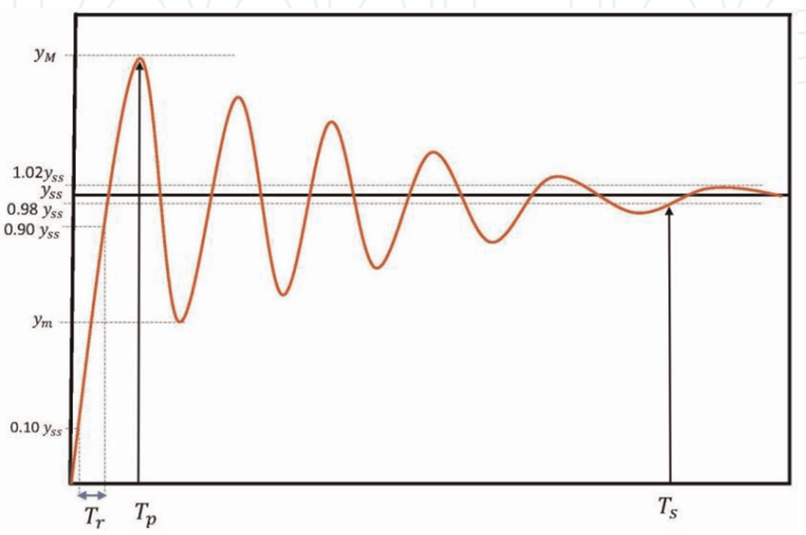

Figure 1.
Unit-step response for underdamped second-order systems.

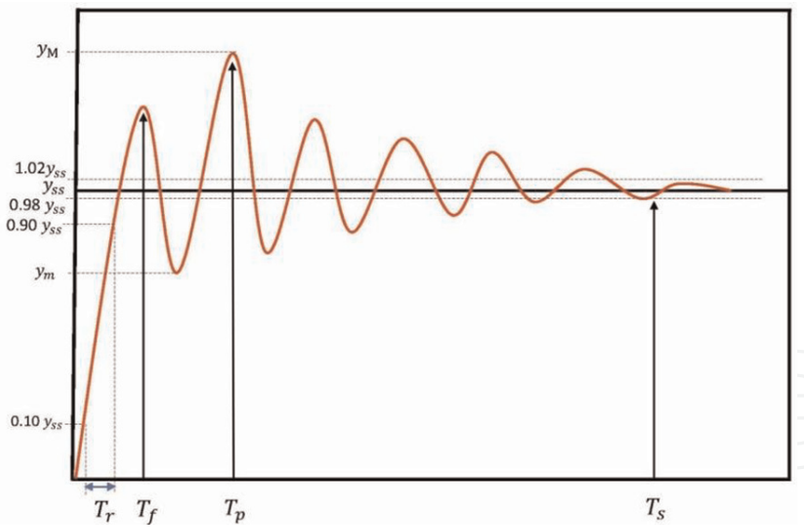

Figure 2.
Unit-step response for some underdamped higher-order systems.

Notice that for second-order systems, the first peak time is always the maximum peak time. However, for underdamped higher-order systems, they may not be the same, as illustrated in **Figure 2**. In general, the response of an underdamped high-order system is similar to that of an underdamped second-order system.

Notice also that the following relationships hold for second-order systems, which will be used in the illustrative example:

$$\xi = \frac{|\ln(MOS)|}{\sqrt{\pi^2 + \ln^2(MOS)}} \tag{4}$$

$$\omega_n = \frac{\pi}{T_p\sqrt{1-\xi^2}} \tag{5}$$

In the design process, a second-order system with satisfactory performances is designated as the reference standard. All the desired specifications with lower and upper bounds are tabulated. Note that the lower bounds of maximum overshot, maximum undershot, setting time, and steady error are set to zero.

3. Proposed controller

The transfer function of general second-order controller is written as

$$G_c(s) = K\left(\frac{cs^2 + ds + 1}{as^2 + bs + 1}\right), \tag{6}$$

where $K > 0,\ \ a,\ \ b,\ \ c, d \in \Re$. It is easy to see that the phase lead-lag controller is just a special case of general second-order controller. The overall control system in this study is shown in **Figure 3**, where $G_p(s)$ is the transfer function of the plant. Besides, $r(t),\ y(t)$, and $e(t)$ denote the reference input, output, and error signal.

For the feedback control system shown in **Figure 3**, the overall response is determined by the parameters of the controller. To establish the proposed time-domain objective function, we first define a function called the deviation ratio (DR), where TDS stands for the time-domain specifications described in Section 2:

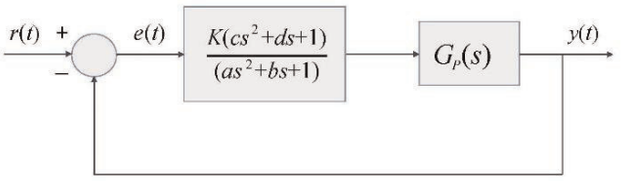

Figure 3.
Unity feedback system with second-order controller.

$$DR(TDS(K,a,b,c,d)) = \begin{cases} \dfrac{TDS(K,a,b,c,d) - TDS\_ub}{TDS\_ub}, & \text{if } TDS(K,a,b,c,d) > TDS\_ub, \\ \dfrac{TDS\_lb - TDS(K,a,b,c,d)}{TDS\_lb}, & \text{if } TDS(K,a,b,c,d) < TDS\_lb, \\ 0, & \text{otherwise}, \end{cases} \tag{7}$$

where $TDS(K,a,b,c,d)$ is any one of the time-domain specifications defined in the last section. In practical designs, some tolerances in time-domain specifications are allowed. Therefore, each of the time-domain specifications has a corresponding lower limit (TDS_lb) and an upper bound (TDS_ub). $DR(TDS(K,a,b,c,d))$ is a measure of how the actual quantity is close to the desired interval specified by lower limit and upper limit. For example, if the value of $DR(T_r(K,a,b,c,d))$ is zero, where the relevant quantity is the rise time, then this means that the rise time lies in the desired interval. That is, the specification is fully satisfied.

Next, we define the objective function $TDR(K,a,b,c,d)$, called the total deviation ratio (TDR), used in this study as follows:

$$\begin{aligned} TDR(K,a,b,c,d) = [&w_1 \cdot (DR(T_r(K,a,b,c,d)))^2 + w_2 \cdot \left(DR\left(T_f(K,a,b,c,d)\right)\right)^2 \\ &+ w_3 \cdot \left(DR\left(T_p(K,a,b,c,d)\right)\right)^2 \\ &+ w_4 \cdot (DR(MOS(K,a,b,c,d)))^2 + w_5 \cdot (DR(MUS(K,a,b,c,d)))^2 \\ &+ w_6 \cdot (DR(T_s(K,a,b,c,d)))^2 + w_7 \cdot (DR(E_{ss}(K,a,b,c,d)))^2] / \sum_{i=1}^{7} w_i \end{aligned} \tag{8}$$

In Eq. (8), ω_i, $i = 1, 2, \cdots, 7$ represents weights that reflect the relative importance of the corresponding terms.

Eqs. (6)–(8) are improved versions of those in Ref. [8]. Once we defined the deviation ratio and total deviation ratio, the problem of the design controller becomes the minimization of $TDR(K,a,b,c,d)$ for all possible parameters. Now, we can use various optimization methods to implement the controller design. In the paper, the cuckoo search algorithm is used to minimize the objective function. Further, if the $TDR(K,a,b,c,d)$ is zero, all specifications are within the range of the design specifications.

4. Cuckoo search algorithm

In this section, the cuckoo search algorithm is briefly introduced. This algorithm was proposed by Yang and Deb in [9, 10]. Cuckoo search represents an optimized

meta-heuristic algorithm that is biologically inspired by the way cuckoo looks for a nest where they could lay eggs in combination with the Lévy flight behavior of some birds and fruit flies.

Now we briefly describe some breeding behaviors of cuckoos. As pointed out in [9], some cuckoo species often lay the eggs in the nests of other host birds, especially those that just spawned eggs. Some host birds can engage direct conflict with the intruding cuckoos. If a host bird discovers the eggs are not their own, they may either throw these alien eggs away or simply abandon its nest and build a new nest elsewhere [9]. Some cuckoo species are often very specialized in the mimicry in color and pattern of the eggs of a few chosen host species, which reduces the probability of their eggs being abandoned and thus increases their reproductively.

In the optimization algorithm, each nest represents a potential solution. The process of cuckoo search algorithm is simplified by three rules [10]:

1. Each cuckoo lays an egg in a randomly selected nest.

2. The best nests will carry over to the next generation.

3. The number of available host nest is fixed, and there is a positive probability that the egg laid by a cuckoo is discovered by the host bird.

There are many variants of the cuckoo search algorithms. In the following, we describe a commonly used version. This algorithm uses a combination of a local random walk and the global random walk, controlled by a switching parameter p_a. This allows for proper balance between exploration and exploitation of the solution space. The local and global random walks for generating the new solution of for cuckoo i can be written as, respectively,

$$x_i^{t+1} = x_i^t + \alpha s \otimes H(p_a - \varepsilon) \otimes \left(x_j^t - x_k^t\right) \tag{9}$$

$$x_i^{t+1} = x_i^t + \alpha L(s, \lambda) \tag{10}$$

$$L(s, \lambda) = \frac{\lambda \Gamma(\lambda) \sin(\pi\lambda/2)}{\pi} \cdot \frac{1}{s^{1+\lambda}}, \quad s \geq s_0 \geq 0 \tag{11}$$

In Eqs. (9)–(11), the notations are explained as follows:

x_i^{t+1}	next position of cuckoo i
x_i^t	current position of cuckoo i
α	step size scaling factor
s	step size
$\otimes$	entry-wise multiplication of two vectors
H	Heaviside function
p_a	switching parameter used to switch local and global walks
ε	a random number drawn from a uniform distribution
x_j^t, x_k^t	current positions of cuckoos j and k selected by random permutation
$L(s, \lambda)$	Lévy distribution used to define the step size of random walk

5. Design procedure

In the design procedure, a second-order system with satisfactory performances is designated as the reference standard. All of the seven desired specifications with lower bounds and upper bounds are tabulated. After that, we set solution vector as

(K, a, b, c, d) and use cuckoo search algorithm to find the minimum value of $TDR(K, a, b, c, d)$ in Eq. (8). To save computation time, the initial populations of 25 host nests are selected using Routh-Hurwitz criterion (for linear time-invariant plants) so that the closed-loop systems are stable.

In the following design procedure, we set maximum generation to be 1000 and $p_a = 0.25$. The optimization process is hierarchical. First, run cuckoo search 25 times to get minimum values. Then the 25 minimum values become elite group. Next, run cuckoo search one more time, and use the group as initial host nests. The final result is the parameters for the second-order controller.

6. Illustrative example

In the section, a numerical example is provided to illustrate the design procedure. In the following simulations, four different plants are used for comparison, which are described in **Table 1**.

As an instance, the step response of the uncompensated system of Plant 1 is shown in **Figure 4**. The peak time is estimated to be 2.4 s, the maximum overshoot is 26.5398%, and the steady-state error E_{ss} is 0.5. Assume we are not satisfied with these time-domain performance indices. Consequently, compensation must be designed for better performances.

Experiment 1. Keep in mind that our reference model is a simple second-order system defined in Eq. (1). Assume that the desired peak time T_p is set to be 1 s and the maximum tolerable overshoot is 0.03. Now we can use Eqs. (4) and (5) to calculate the corresponding damping ratio ξ and natural frequency ω_n, respectively. This determines the desired reference model in Eq. (1). Based on the resulting reference model, we can calculate its seven performance indices depicted in Section 2 in order to establish reasonable bounds for these performance indices. We may put 2% tolerance on rise time T_r, first peak time T_f, and maximum peak time T_p. Only upper bounds are specified for the remaining four performance indices. The full specifications for Experiment 1 are listed in the second column of **Table 2**.

We set all the weights to be 1, i.e., $\omega_i = 1, i = 1, \ 2, \ \ldots, \ 7$. Suppose the maximum steady-state error is limited to [0.001, 0.022], and then we can set the range of K to [7.4091, 166.5000]; the ranges of a, b, c, and d are all set to $[-100, 100]$. The specified ranges of controller parameters are listed in **Table 3**.

The resulting control parameters are shown in **Table 4**.

The time-domain performances of the resulting closed-loop systems are shown in **Table 5**. Notice that the final objective values for four plants shown in the final row of **Table 5** are all zeros, which means that all seven design specifications are all met using our controller design methods.

The time responses due to unit-step input of the resulting four closed-loop systems are shown in **Figure 5**. These responses look quite nice.

	Transfer function	System type	Closed-loop stability
Plant 1	$G_p(s) = \frac{120}{s^2+12s+20}$	0	Stable
Plant 2	$G_p(s) = \frac{100s+1}{s(10s+1)(s+1)}$	1	Stable
Plant 3	$G_p(s) = \frac{100}{(s+1)(s+2)(s+4)}$	0	Unstable
Plant 4	$G_p(s) = \frac{1}{s(s+1)}$	1	Stable

Table 1.
Description of the four plants.

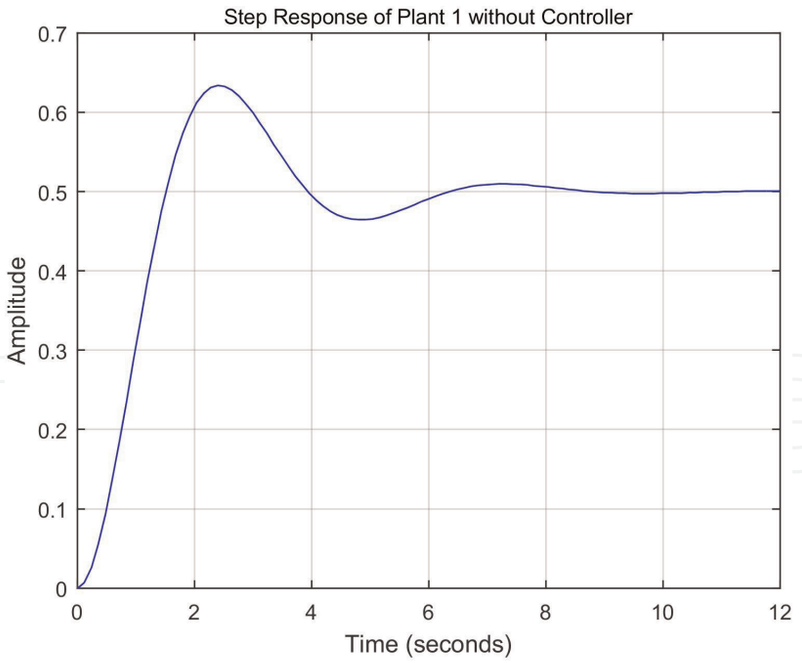

Figure 4.
Step response of the Plant 1 without controller.

	Experiment 1	**Experiment 2**	**Experiment 3**
T_r	[0.4730, 0.4923]	[0.7095, 0.7385]	[0.9461, 0.9847]
T_f	[0.9788, 1.0188]	[1.4682, 1.5281]	[1.9576, 2.0375]
T_p	[0.9788, 1.0188]	[1.4682, 1.5281]	[1.9576, 2.0375]
MOS	≤ 0.03	≤ 0.03	≤ 0.03
MUS	≤ 0.02	≤ 0.02	≤ 0.02
T_s	≤ 1.0724	≤ 1.6086	≤ 2.1448
E_{ss}	≤ 0.02	≤ 0.02	≤ 0.02

Table 2.
Design specifications.

	Plant 1	**Plant 2**	**Plant 3**	**Plant 4**
K	[7.4091,166.5000]	[45.4545, 1000]	[7.4091, 166.5000]	[45.4545, 1000]
a	[−100,100]	[−100,100]	[−100,100]	[−100,100]
b	[−100,100]	[−100,100]	[−100,100]	[−100,100]
c	[−100,100]	[−100,100]	[−100,100]	[−100,100]
d	[−100,100]	[−100,100]	[−100,100]	[−100,100]

Table 3.
Search ranges of controller parameters.

Experiment 2. In this experiment, assume that the first peak time and the maximum peak time are both set to 1.5 s. Following the procedure in Experiment 1, the full design specifications for Experiment 2 are also listed in **Table 2**. The specified ranges of controller parameters are listed in **Table 3**. The resulting control parameters are shown in **Table 6**.

	Plant 1	**Plant 2**	**Plant 3**	**Plant 4**
K	8.4963	995.7351	14.6760	934.2808
a	100.0000	1.6668	1.7609	−0.1364
b	30.4229	8.8714	60.2882	−0.8138
c	3.1336	98.4428	0.6624	−86.8253
d	7.3230	48.7921	1.2683	−85.5579

Table 4.
Resulting controller parameters in Experiment 1.

	Plant 1	**Plant 2**	**Plant 3**	**Plant 4**
T_r	0.4804	0.4734	0.4923	0.4735
T_f	1.0131	1.0131	0.9804	1.0131
T_p	1.0131	1.0131	0.9804	1.0131
MOS	0.0000	0.0294	0.0297	0.0281
MUS	0.0179	0.0004	0.0093	0.0004
T_s	0.7717	1.0295	0.9968	1.0295
E_{ss}	0.0192	0.0010	0.0054	0.0011
TDR	0	0	0	0

Table 5.
Time-domain performance indices of the resulting systems in Experiment 1.

The time-domain performances of the resulting closed-loop systems are shown in **Table 7**. Notice again that the final objective values for four plants are all zeros, which means that all seven design specifications are all met using our controller design methods.

The time responses due to unit-step input of the resulting four closed-loop systems are shown in **Figure 6**.

Experiment 3. In this experiment, assume that the first peak time and the maximum peak time are both set to 2.0 s. Following the procedure in Experiment 1, the full design specifications for Experiment 3 are also shown in **Table 2**. The specified ranges of controller parameters are listed in **Table 3**. The resulting control parameters are shown in **Table 8**.

The time-domain performances of the resulting closed-loop systems are shown in **Table 9**. Notice that the final objective values for four plants are zeros, which means that all seven design specifications are met using our controller design methods.

The time responses due to unit-step input of the resulting four closed-loop systems are shown in **Figure 7**.

In the illustrative example above, four different plants and three different time-domain specifications were used. As illustrated in the preceding example, before searching the best controller parameters, we first use some of the time-domain specifications (e.g., desired peak time and maximum overshoot) to fully specify the desired simple second-order reference model. Then we may use this reference model to define reasonable bounds for other time-domain specifications. Finally, we define the deviation ratios (i.e., percentage errors) and total deviation ratio as the objective function. If the final value of the objective function is zero, then all seven specifications are satisfied. But, in the most general cases, there is no

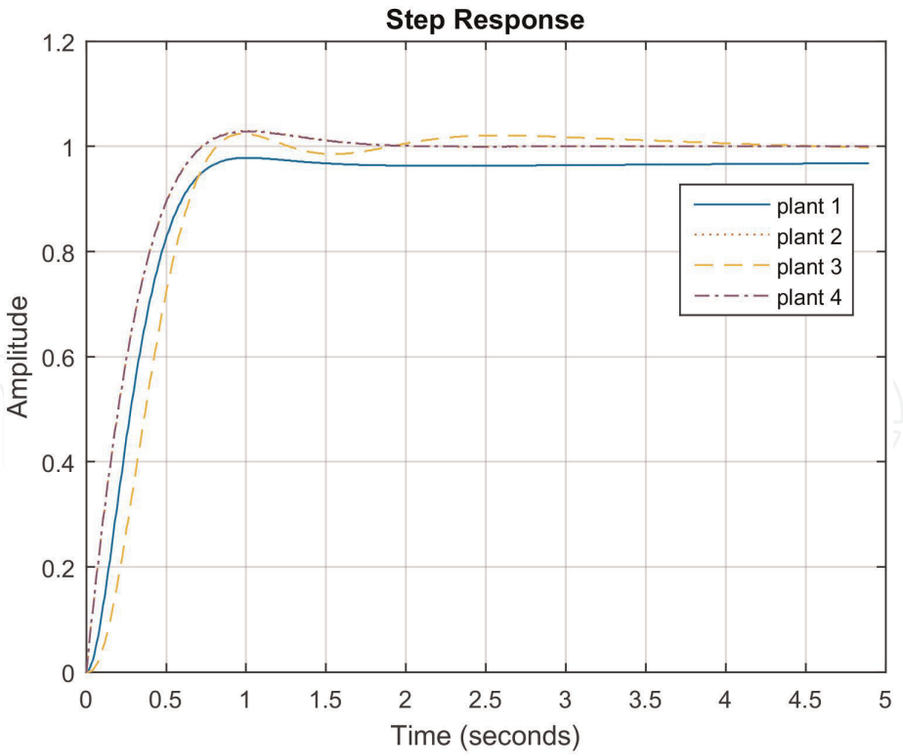

Figure 5.
Step responses of four resulting systems in Experiment 1.

	Plant 1	Plant 2	Plant 3	Plant 4
K	8.4963	995.7351	14.6760	934.2808
a	100.0000	1.6668	1.7609	−0.1364
b	30.4229	8.8714	60.2882	−0.8138
c	3.1336	98.4428	0.6624	−86.8253
d	7.3230	48.7921	1.2683	−85.5579

Table 6.
Resulting controller parameters in Experiment 2.

	Plant 1	Plant 2	Plant 3	Plant 4
T_r	0.7133	0.7105	0.7384	0.7102
T_f	1.5197	1.5197	1.4952	1.5197
T_p	1.5197	1.5197	1.4952	1.5197
MOS	0.0103	0.0282	0.0299	0.0281
MUS	0.0128	0.0004	0.0139	0.0004
T_s	1.0929	1.5442	1.5197	1.5442
E_{ss}	0.0198	0.0010	0.0129	0.0010
TDR	0	0	0	0

Table 7.
Time-domain performance indices of the resulting systems in Experiment 2.

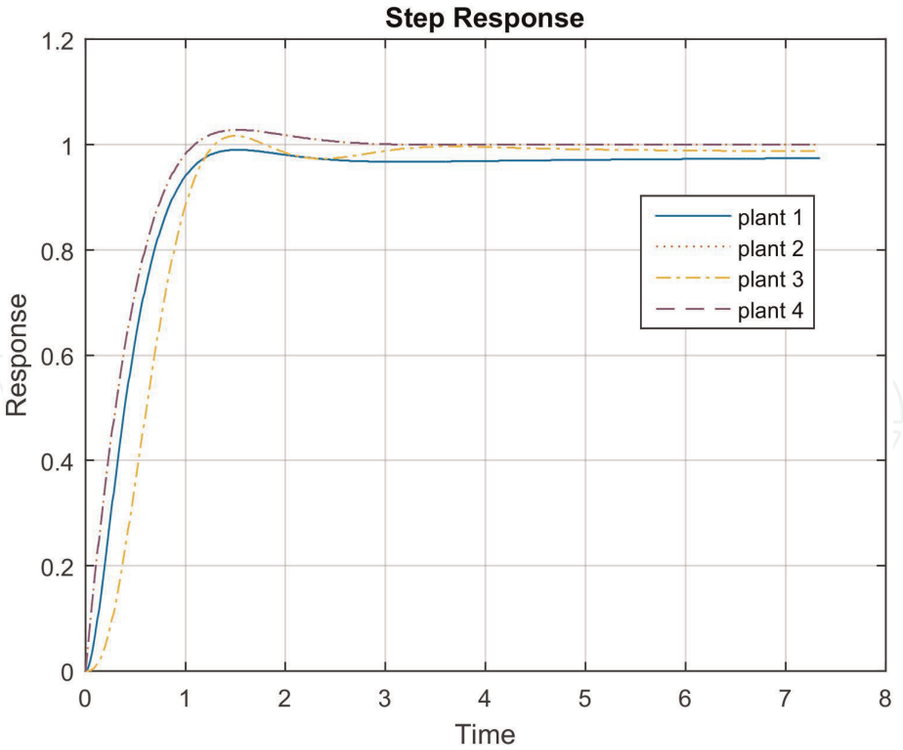

Figure 6.
Step responses of four resulting systems in Experiment 2.

	Plant 1	Plant 2	Plant 3	Plant 4
K	8.4963	995.7351	14.6760	934.2808
a	100.0000	1.6668	1.7609	−0.1364
b	30.4229	8.8714	60.2882	−0.8138
c	3.1336	98.4428	0.6624	−86.8253
d	7.3230	48.7921	1.2683	−85.5579

Table 8.
Resulting controller parameters in Experiment 3.

	Plant 1	Plant 2	Plant 3	Plant 4
T_r	0.9656	0.9468	0.9794	0.9467
T_f	1.9935	2.0262	1.9609	2.0262
T_p	1.9935	2.0262	1.9609	2.0262
MOS	0.0024	0.0292	0.0220	0.0284
MUS	0.0154	0.0004	0.0138	0.0004
T_s	1.4927	2.0589	1.9935	2.0589
E_{ss}	0.0198	0.0010	0.0199	0.0011
TDR	0	0	0	0

Table 9.
Time-domain performance indices of the resulting systems in Experiment 3.

guarantee that the final value of the objective function will always be zero. In that case, some specifications might be violated. The design goal is to choose the best controller parameters so that the objective value is as close to zero as possible. In the

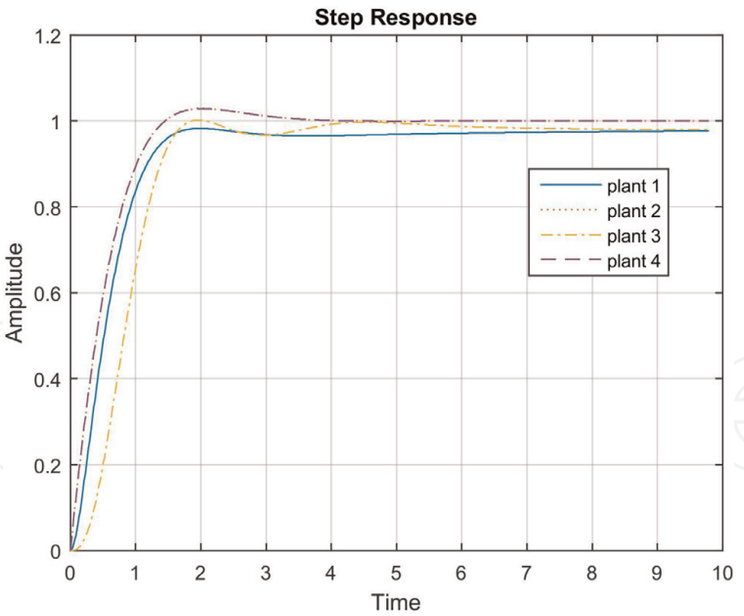

Figure 7.
Step responses of four resulting systems in Experiment 3.

simulations above, it is fortunate that the final objective values of all simulations are zeros and all seven time-domain specifications are met. This also shows that the cuckoo search is an excellent optimizer for searching best controller parameters.

7. Conclusion

In this study, a design procedure of second-order controller for various plants has been proposed. The final controller was obtained by minimizing a time-domain cost function which is weighted sum of important time-domain performance indices including the rise time, first peak time, maximum peak time, maximum overshoot, maximum undershoot, setting time, and steady-state error. In our approach, the desired design specifications were built via a good second-order reference model. Cuckoo search algorithm was adopted to search the optimal controller parameters. Detailed simulations with different design specifications for four plants were provided to illustrate the use of the proposed design method. From the simulation results, the resulting performances of the closed-loop systems are quite good by using the proposed method, which justifies the usefulness of our method. We wish to point out that the methodology proposed in this study can easily be modified to handle the time delay or nonlinear plants. This constitutes an interesting future research topic.

IntechOpen

IntechOpen

Author details

Huey-Yang Horng
Department of Electronic Engineering, I-Shou University, Kaohsiung City, Taiwan, R.O.C

*Address all correspondence to: hyhorng@isu.edu.tw

IntechOpen

© 2019 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the Creative Commons Attribution License (http://creativecommons.org/licenses/by/3.0), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited. CC BY

References

[1] Panda S, Sahu BK, Mohanty PK. Design and performance analysis of PID controller for an automatic voltage regulator system using simplified particle swarm optimization. Journal of the Franklin Institute. 2012;**349**(8): 2609-2625

[2] Perng JW, Chen GY, Hsieh SC. Optimal PID controller design based on PSO-RBFNN for wind turbine systems. Energies. 2014;**7**(1):191-209

[3] Kesarkar AA, Selvaganesan N. Tuning of optimal fractional-order PID controller using an artificial bee colony algorithm. Systems Science & Control Engineering. 2015;**3**:99-105

[4] Ghosal S, Darbar R, Neogi B, Das A, Tibarewala DN. Application of swarm intelligence computation techniques in PID controller tuning: A review. In: Proceedings of the InConINDIA 2012, AISC 132; 2012. pp. 195-208

[5] Saroja K, Stefka Sharon R, Meena S, Chitra K. Multi-loop PID controller design for distillation column using firefly algorithm. International Journal of Engineering and Technology (IJET). 2017;**9**(2):1404-1410

[6] Tan N. Computation of stabilizing lag/lead controller parameters. Computers and Electrical Engineering. 2003;**29**:835-849

[7] Kuo YS, Lin JY, Tang JC, Hsieh JG. Lead-lag compensator design based on vector margin and steady-state error of the step response via particle swarm optimization. In: International Conference on Fuzzy Theory and Its Applications (iFuzzy); 2016. pp. 96-101

[8] Horng HY. Design of lead-lag controller via time-domain objective function by using cuckoo search. In: Lecture Notes in Electrical Engineering (ICITES2013); Vol. 293; 2014. pp. 1083-1091

[9] Yang XS, Deb S. Cuckoo search via Lévy flights. In: World Congress on Nature & Biologically Inspired Computing, December 2009; India. USA: IEEE Publications; 2009. pp. 210-214

[10] Yang XS, Deb S. Engineering optimisation by cuckoo search. International Journal of Mathematical Modelling and Numerical Optimisation. 2010;**1**(4):330-343

[11] Fister I Jr, Fister D, Fister I. Comprehensive review of cuckoo search variants and hybrids. International Journal of Mathematical Modelling and Numerical Optimisation. 2013;**4**(4): 387-409

[12] Yang XS, Deb S. Cuckoo search: Recent advances and applications. Neural Computing and Applications. 2014;**24**(1):169-174

[13] Shehab M, Khader AT, Al-Betar MA. A survey on applications and variants of the cuckoo search algorithm. Applied Soft Computing. 2017;**61**: 1041-1059

[14] Abd Elazim SM, Ali ES. Optimal power system stabilizers design via cuckoo search algorithm. International Journal of Electrical Power & Energy Systems. 2016;**75**:99-107

[15] Abdelaziz AY, Ali ES. Load frequency controller design via artificial cuckoo search algorithm. Electric Power Components & Systems. 2016;**44**(1): 90-98

[16] Anh L, Tam NM, Nghia LT, Anh QH. Optimal power system stabilizer parameters tuned by cuckoo search algorithm. International Journal

of Engineering Research & Technology (IJERT). 2017;**6**(11):24-28

[17] Wongkaew B, Puangdownreong D. Application of cuckoo search to synthesize analog controllersi. nternational journal of circuits Systems and Signal Processing. 2019;**13**:79-84

IntechOpen

IntechOpen