ICIC Express Letters
Volume **10**, Number **11**, November **2016**

ICIC International ©2016 ISSN 1881-803X
pp. **2765–2771**

CONTROLLER DESIGN USING TIME-DOMAIN OBJECTIVE FUNCTIONS THROUGH FLOWER POLLINATION ALGORITHM

Huey-Yang Horng

Department of Electronic Engineering
I-Shou University
No. 1, Sec. 1, Syuecheng Rd., Dashu District, Kaohsiung City 84001, Taiwan
hyhorng@isu.edu.tw

Received April 2016; accepted July 2016

Abstract. *A large proportion of industrial systems can be represented by linear time-invariant transfer functions. The proportional-integral-derivative (PID) controller is one of the most commonly used controllers in industrial enterprises. The lead-lag (or lag-lead) controller is a more practical alternative to the PID controller. This paper focuses on the optimization of a time-domain objective function for a generalized controller similar to a lead-lag controller. The proposed objective function includes eight time-domain specifications, including delay time, rise time, and peak time. In this research, the flower pollination algorithm (FPA) is applied to determine the optimal solutions. FPA is a recently developed metaheuristic optimization method. The algorithm emulates biological flower pollination. If a plant can be modeled as a linear time-invariant transfer function, then the proposed method can be used to design a corresponding controller, which may meet specifications exactly or approximately.*

Keywords: Flower pollination algorithm, Lead-lag controller, Meta-heuristic optimization

1. **Introduction.** Most industrial plant systems can be represented by linear time-invariant transfer functions. Proportional-integral-derivative (PID) controllers are probably the most commonly used controllers in industrial applications. Several methods have been proposed for tuning PID controller parameters [1-4].

Lead-lag controllers provide a more practical alternative to PID controllers. The design of lead-lag controllers has been studied in the past [5,6]. Ou and Lin proposed a method based on a genetic algorithm and particle swarm optimization to design a PID controller [7]. Horng used cuckoo search to design a lead-lag controller [8]. Yang's flower pollination algorithm is one of the latest metaheuristic techniques [9]. A multiobjective variant of the flower pollination algorithm was also published [10]. Cuevas et al. compared five evolutionary computation techniques: swarm optimization, artificial bee colony optimization, electromagnetism-like optimization, cuckoo search, and flower pollination [11]. In that research, the flower pollination algorithm showed the same performance as cuckoo search, and greater performance than the other techniques. A flower pollination algorithm is more straightforward and easier to program than a cuckoo search. Therefore, a flower pollination algorithm was used in this study.

A generalized version of a lead-lag controller is proposed and discussed in Section 2. Time-domain objective functions are presented in Section 3. In Section 4, a flower pollination algorithm and its design procedure are discussed. Two illustrative examples are presented in Section 5, and the conclusions are presented in Section 6.

2. Generalized Lead-lag-like Controller. The transfer function of a lead-lag-like controller is written as

$$G_c(s) = K\left(\frac{T_1 s + 1}{\alpha T_1 s + 1}\right) \cdot \left(\frac{T_2 s + 1}{\beta T_2 s + 1}\right), \tag{1}$$

where $K > 0$, $\alpha > 0$, $T_1 > 0$, $\beta > 0$, and $T_2 > 0$. The proposed controller could be a lead-lag, two-stage phase-lead, or two-stage phase-lag controller, depending on the parameters α and β. This general-purpose lead-lag-like controller is more useful than a lead-lag (lag-lead) controller.

3. Time-domain Objective Functions. The unit-step response is the response of a control system when the input is a unit-step function. Let $y(t)$ be the unit-step response and y_{ss} the steady state. Some well known unit-step responses are delay time T_d, rise time T_r, percentage maximum overshoot $\%OS$, setting time T_s, and steady-state error E_{ss} [1]. For a more general system, the following specifications are defined:
1). First peak time, T_f, is the time required to reach the first peak.
2). Maximum peak time, T_m, is the time required to reach the maximum peak.
3). Percentage maximum undershoot, $\%US$, is defined as

$$y_{us} = \min(y(t)),\ t \geq T_f, \quad \%US = \begin{cases} \dfrac{(y_{ss} - y_{us})}{y_{ss}}, & \text{if } y_{us} < y_{ss}, \\ 0, & \text{if } y_{us} \geq y_{ss}. \end{cases} \tag{2}$$

For a second-order system, the first peak time is equal to the maximum peak time. However, for a general system, these two quantities are not always equal. To establish the proposed time-domain objective function, the deviation ratio (DR) is first defined as follows.

$$\begin{aligned} \mathrm{DR}(TDS) &= f(K, \alpha, T_1, \beta, T_2 | TDS) \\ &= \begin{cases} 0, & \text{if } lb \leq f(K, \alpha, T_1, \beta, T_2 | TDS) \leq ub \\ \dfrac{f(K, \alpha, T_1, \beta, T_2 | TDS) - ub}{ub}, & \text{if } f(K, \alpha, T_1, \beta, T_2 | TDS) > ub \\ \dfrac{lb - f(K, \alpha, T_1, \beta, T_2 | TDS)}{lb}, & \text{if } f(K, \alpha, T_1, \beta, T_2 | TDS) < lb \end{cases} \end{aligned} \tag{3}$$

where each TDS is some time-domain specification, which might be one of delay time, rise time, percentage maximum overshoot, or some other specification. The controller parameters are K, α, T_1, β, and T_2, as given by Equation (1). The objective function includes eight specifications, namely, delay time, rise time, first peak time, maximum peak time, percent maximum overshoot, percent maximum undershoot, setting time, and steady-state error. DR, given by Equation (3), is a measure of how close a value is to a desired interval. In practical designs, some tolerances in time-domain specifications are allowed. Each TDS has a lower bound (lb) and an upper bound (ub). If some DR is zero, the corresponding specification in the desired interval is fully satisfied.

Next, the following objective function is proposed:

$$\begin{aligned} TDOF = &\left[w_1 \mathrm{DR}^2(T_d) + w_2 \mathrm{DR}^2(T_r) + w_3 \mathrm{DR}^2(\%OS) + w_4 \mathrm{DR}^2(T_s) \right. \\ &\left. + w_5 \mathrm{DR}^2(E_{ss}) + w_6 \mathrm{DR}^2(T_f) + w_7 \mathrm{DR}^2(T_m) + w_8 \mathrm{DR}^2(\%US)\right] / TW \end{aligned} \tag{4}$$

where $TW = \sum_{i=1}^{8} w_i$. In Equation (4), w_i represents weights that reflect the relative priorities of the corresponding terms. Consequently, the controller design problem is essentially the minimization of $TDOF$ for all possible parameters. Equations (1) to (4) are improved versions of those in reference [8].

4. **Flower Pollination Algorithm and Design Procedure.** Yang [9] emulated biological flower pollination to construct an optimization algorithm based upon the following rules.

1). The global pollination processes are biotic pollination and cross-pollination through which pollens are transported pollinators that conduct random walks of the Lévy flight type.
2). Local pollination is regarded as abiotic and is a form of self-pollination.
3). Reproduction probability depends on flower constancy, which is related to the resemblance between the two flowers involved in pollination.
4). The algorithm determines local and global pollination according to a probability represented as $p \in [0, 1]$. Local pollination occurs if the algorithm generates a random number less than or equal to p; this simulates the processes of pollination that results from physical proximity and wind.

Taking inspiration from these idealized characteristics, we can construct a flower-based algorithm which can be called the flower pollination algorithm (FPA). This algorithm has two key steps, namely, global pollination and local pollination. In the global pollination step, flower pollen gametes are transported by pollinators such as insects; thus, pollen can be moved over long distances because insects often cover considerably long distances. This is the best fit between pollination and reproduction, and we represent it as g^*. The flower constancy can be denoted as

$$\mathbf{x}_i^{t+1} = \mathbf{x}_i^t + L\left(\mathbf{x}_i^t - g^*\right) \tag{5}$$

where $\mathbf{x}_i^t$ is the solution vector $\mathbf{x}_i$ for a pollen i at iteration t, and g^* is the best fit solution among all solutions at the current iteration or generation. Lévy flight is used to represent the movement of insects; hence, $L > 0$ from the Lévy distribution

$$L \sim \frac{\lambda\Gamma(\lambda)\sin(\pi\lambda/2)}{\pi} \cdot \frac{1}{s^{1+\lambda}}, \quad (s \geq s_0 > 0) \tag{6}$$

In Equation (6), $\Gamma(\lambda)$ is a standard gamma function, and the distribution is valid for large steps ($s > 0$). Then, to develop local pollination, both Rule 2 and Rule 3 can be represented as

$$\mathbf{x}_i^{t+1} = \mathbf{x}_i^t + \varepsilon\left(\mathbf{x}_j^t - \mathbf{x}_k^t\right) \tag{7}$$

where $\mathbf{x}_j^t$ and $\mathbf{x}_k^t$ represent pollen from different flowers of the same plant species. This way, flower constancy in a limited neighborhood is mimicked. Mathematically, if $\mathbf{x}_j^t$ and $\mathbf{x}_k^t$ are from the same species or chosen from the same population, Equation (7) is equivalent to a local random walk if we draw ε from a uniform distribution in $[0, 1]$. By Rule 4, we can use the switch probability p to switch from global pollination to intensive local pollination.

In the design procedure, a second-order system with satisfactory operation is designated as the reference standard. All the eight desired specifications, lower bounds, and upper bounds are tabulated. Subsequently, a solution vector set $\mathbf{x}_i = (K_i, \alpha_i, T_{1i}, \beta_i, T_{2i})$ and flower pollination are used to determine the minimum value of $TDOF$ from Equation (4). To save computing time, the initial values are preselected. The initial values are chosen such that the closed-loop system stability is maintained. The stability is determined using the Routh-Hurwitz criterion. The maximal number of iterations is 750 and $p = 0.8$. The optimization process is hierarchical. First, FPA is run 30 times to obtain 30 distinct minimal values. These 30 minimal values constitute the elite group in later calculations. Next, FPA is run one more time, the elite group values are used as initial values in this iteration. The final results obtained are the parameters for the design of the controller.

5. **Illustrative Examples.** Two numerical examples of unity feedback systems are provided below.

Example 5.1. *We consider a unity feedback system with the following forward transfer function:*

$$G_p(s) = \frac{180}{s^2 + 18s + 80}.$$

The forward-path transfer function is type 0, and the closed-loop system is stable. The step response of the uncompensated system is shown in Figure 1. The peak time is 2.76 s, the percentage overshoot is 20.50%, and the steady-state error E_{ss} is 0.6923.

The response of the uncompensated system is not satisfactory. Hence, a controller is used to improve the transient response. We choose a well-behaved second-order system as the reference standard. The desired specifications are listed in Table 1. We set each weight w_i to be 1. After the design procedure has been completed, the obtained parameters are $K = 38.9011$, $\alpha = 20.7979$, $T_1 = 0.1421$, $\beta = 0.1516$, and $T_2 = 0.0939$. Since all the specifications are in the desired range, all the deviation ratios are zeros, as listed in Table 1. The designed controller fully meets all the desired specifications. The compensated step response is shown in Figure 2.

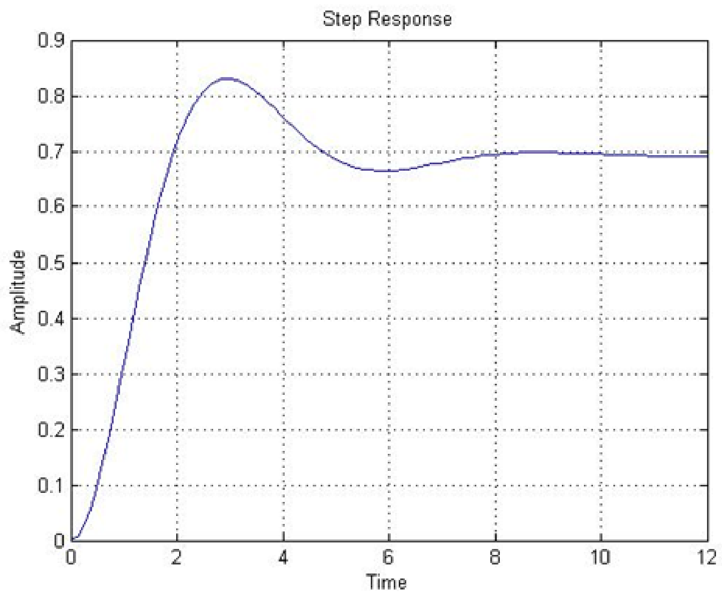

Figure 1. Uncompensated step response for Example 5.1

Table 1. Desired specifications and designed values for Example 5.1

Specification	Desired interval	w_i	Designed value	DR (TDS)
T_p	$[0.0988, 0.1008]$	1	0.0990	0.0000
T_m	$[0.0988, 0.1008]$	1	0.0990	0.0000
T_r	$[0.0477, 0.0487]$	1	0.0486	0.0000
T_d	$[0.0307, 0.0314]$	1	0.0309	0.0000
$\%OS$	≤ 0.03	1	0.0217	0.0000
$\%US$	≤ 0.02	1	0.0063	0.0000
T_s	≤ 0.1035	1	0.1006	0.0000
E_{ss}	≤ 0.22	1	0.0113	0.0000

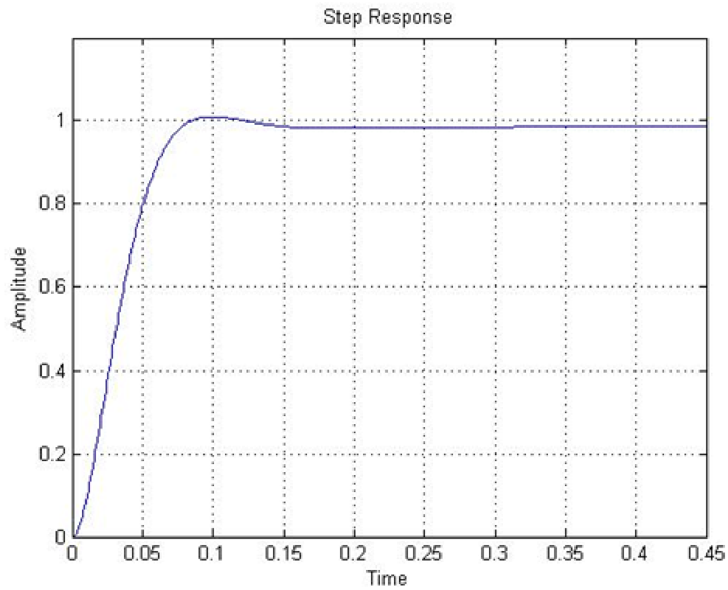

FIGURE 2. Compensated step response for Example 5.1

Example 5.2. *We consider a unity feedback system with the following forward transfer function:*

$$G_p(s) = \frac{800}{(s+2)(s+4)(s+8)}.$$

The closed-loop system is unstable. The step response of the uncompensated system is shown in Figure 3. Let us follow the same procedure as in Example 5.1. The desired specifications are listed in Table 2. After the design procedure is finished, the obtained parameters are $K = 7.6370$, $\alpha = 84.1030$, $T_1 = 0.5561$, $\beta = 0.0100$, and $T_2 = 0.1395$. All the desired specifications are satisfied. The compensated step response is shown in Figure 4.

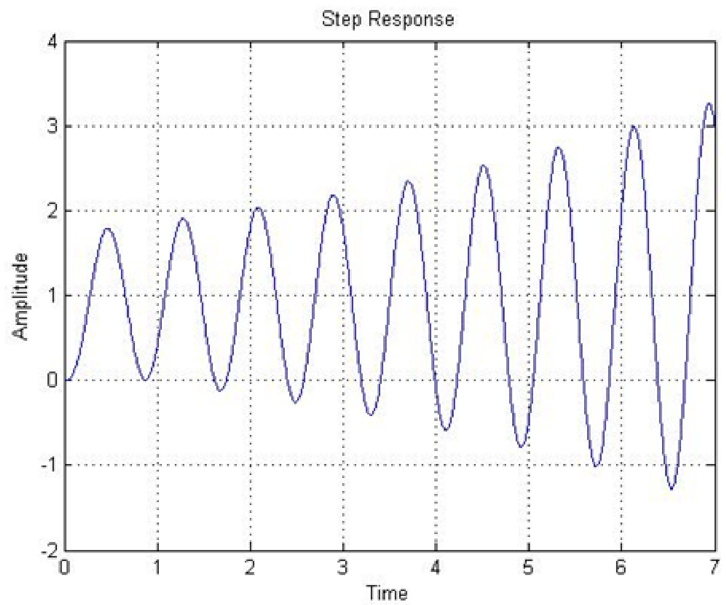

FIGURE 3. Uncompensated step response for Example 5.2

TABLE 2. Desired specifications and designed values for Example 5.2

Specification	Desired interval	w_i	Designed value	DR (TDS)
T_p	$[1.4850, 1.5132]$	1	1.4857	0.0000
T_m	$[1.4850, 1.5132]$	1	1.4857	0.0000
T_r	$[0.7168, 0.7313]$	1	0.7260	0.0000
T_d	$[0.4611, 0.4704]$	1	0.4632	0.0000
$\%OS$	≤ 0.03	1	0.0206	0.0000
$\%US$	≤ 0.02	1	0.0033	0.0000
T_s	≤ 1.5150	1	1.5104	0.0000
E_{ss}	≤ 0.011	1	0.0104	0.0000

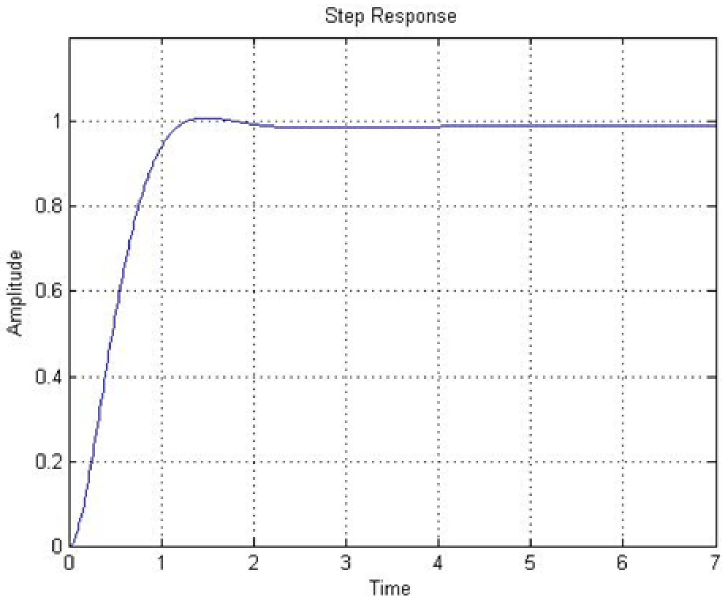

FIGURE 4. Compensated step response for Example 5.2

6. **Conclusions.** Numerous industrial plant systems can be represented by linear time-invariant transfer functions. In this paper, a design procedure using a flower pollination algorithm is proposed; the designed controllers meet specifications exactly or approximately. The proposed time-domain objective function is expressed in terms of eight specifications, including delay time, rise time, first peak time, maximum peak time, percent maximum overshoot, percent maximum undershoot, setting time, and steady-state error. Therefore, the controller design problem is basically the minimization of $TDOF$ for all possible parameters. Computer simulations validate the usefulness of the proposed method. In future, the proposed method may be modified to design controllers for time-delay systems.

REFERENCES

[1] F. Golnaraghi and B. C. Kuo, *Automatic Control Systems*, 9th Edition, John Wiley & Sons, Inc., 2011.

[2] Y. Okada, Y. Yamakawa, T. Yamazaki and S. Kurosu, Tuning method of PID controller for desired damping coefficient, *SICE Annual Conference*, Kagawa University, Japan, pp.795-799, 2007.

[3] S. Panda, B. K. Sahu and P. K. Mohanty, Design and performance analysis of PID controller for an automatic voltage regulator system using simplified particle swarm optimization, *Journal of the Franklin Institute*, vol.349, no.8, pp.2609-2625, 2012.

[4] R. P. Sree, M. N. Srinivas and M. Chidambaram, A simple method of tuning PID controllers for stable and unstable FOPTD systems, *Computers and Chemical Engineering*, vol.28, pp.2201-2218, 2004.

[5] J. Yang, C. F. Chen, C. S. Chen and Y. S. Xu, An approach to automatic tuning of phase-lead and phase-lag compensators, *Proc. of the 30th IEEE Conference on Decision and Control*, Brighton, England, vol.3, pp.2944-2945, 1991.
[6] N. Tan, Computation of stabilizing lag/lead controller parameters, *Computers and Electrical Engineering*, vol.29, pp.835-849, 2003.
[7] C. Ou and W. Lin, Comparison between PSO and GA for parameters optimization of PID controller, *International Conference on Mechatronics and Automation*, Luoyang, China, pp.2471-2475, 2006.
[8] H. Y. Horng, Design of lead-lag controller via time-domain objective function by using cuckoo search, *Lecture Notes in Electrical Engineering*, vol.293, pp.1083-1091, 2014.
[9] X. S. Yang, Flower pollination algorithm for global optimization, *Unconventional Computation and Natural Computation*, Springer, pp.240-249, 2012.
[10] X. S. Yang, M. Karamanoglu and X. He, Multi-objective flower algorithm for optimization, *International Conference on Computational Science*, pp.861-868, 2013.
[11] E. Cuevas, J. Gálvez, S. Hinojosa, O. Avalos, D. Zaldívar and M. Pérez-Cisneros, A comparison of evolutionary computation techniques for IIR model identification, *Journal of Applied Mathematics*, vol.2014, 2014.