

Tuning of proportional integral derivative controller based on firefly algorithm

D. Kumanan & B. Nagaraj

To cite this article: D. Kumanan & B. Nagaraj (2013) Tuning of proportional integral derivative controller based on firefly algorithm, Systems Science & Control Engineering: An Open Access Journal, 1:1, 52-56, DOI: [10.1080/21642583.2013.770375](https://doi.org/10.1080/21642583.2013.770375)

To link to this article: <https://doi.org/10.1080/21642583.2013.770375>



© 2013 The Author(s). Published by Taylor & Francis.



Published online: 18 Oct 2013.



Submit your article to this journal [↗](#)



Article views: 3823



View related articles [↗](#)



Citing articles: 5 View citing articles [↗](#)

Tuning of proportional integral derivative controller based on firefly algorithm

D. Kumanan* and B. Nagaraj

Department of Electronics and Instrumentation Engineering, Karunya University, Coimbatore, Tamil Nadu, India

(Received 6 December 2012; final version received 10 January 2013)

In this paper, a novel meta-heuristics algorithm, namely the firefly algorithm (FA), is applied to the proportional integral derivative (PID) controller parameter tuning for flow process. The controller is used to control flow rate and to maintain the desired set point. Simulation results indicate that the applied FA is effective and efficient. Good closed-loop system performance is achieved on the basis of the considered PID controllers tuning procedures. Moreover, the observed results are compared with the ones obtained by the Ziegler–Nichols method. The comparison of both meta-heuristics shows a superior performance for the FA PID controller tuning of the considered system than the Ziegler–Nichols tuned controller.

Keywords: meta-heuristics, firefly algorithm, flow process, PID controller, parameter tuning

1. Introduction

Despite the appearance of many complicated control theories and techniques, more than 95% of control loops still use proportional integral derivative (PID) controllers (Åström & Hägglund, 1995). This is mainly because PID controllers have structure simplicity and meaning of the corresponding three parameters, which can be easily understood by process operators. Moreover, PID controllers have the advantage of good stability and high reliability.

A typical structure of a PID controller involves three separate elements: the proportional, integral and derivative values. The proportional value determines the reaction to the current error, the integral value determines the reaction based on the sum of recent errors, and the derivative value determines the reaction based on the rate at which the error has been changing.

Usually, the PID controller is poorly tuned due to highly changing dynamics of most processes caused by the non-linear changes in the overall system. The conventional optimization methods of tuning are based on Ziegler–Nichols (ZN) tuning or cohen-coon settings (Table 1). As an alternative, meta-heuristics could be applied. The usage of nature inspired metaheuristic algorithm has been applied widely in most of the fields of process control (dos Santos Coelho & Mariani, 2012; Nagaraj & Vijayakumar, 2011, 2012; Roeva & Slavov, 2012; Solihin, Tack, & Kean, 2011). The main feature of this concept is the ability of self-learning and self-predicting some desired outputs which optimize PID parameters.

The objective of the research is to develop a soft computing-based PID tuning methodology for optimizing

control of flow rate. This research proposes the development of a tuning technique that would be best suitable for optimizing the control of processes operating in single-input and single-output process control loop. The proposed method has been proved to be the best by comparing the control performance of the system with the soft computing method to that of the system tuned using conventional method of Ziegler–Nichols.

In this scheme, the firefly algorithm (FA) is used to select proper tuning parameters, which is achieved using the FA to minimize the integral square error (ISE). This integrated approach improves the system performance, cost-effectiveness, efficiency, dynamism and reliability of the designed controller.

2. System identification and controller design

The classical tuning procedure such as the Ziegler–Nichols method is employed to find out the values of k_p , k_i and k_d . Although the classical methods are unable to provide the best solutions, they give initial values needed to start the soft computing algorithms (Nagaraj & Vijayakumar, 2012). Due to the high potential of metaheuristic techniques in finding the optimal solutions, the best values of k_p , k_i and k_d are obtained. The Ziegler–Nichols tuning method is used to evaluate the PID gains for the system. The simulations are carried out using Intel® Core™i5-2450 CPU @ 2.50 GHz, 4 GB Memory (RAM), Windows 7 (64 bit) operating system.

The system identification problem is to estimate a model of a system based on observed input–output data.

*Corresponding author. Email: kuman89@ymail.com

Table 1. Results from PID controller tuning.

Tuning methods	PID parameters		
	K_p	K_i	K_d
ZN	0.1390	0.2479	0.0198
FA	6.6488	4.5113	0.6488

Several ways to describe a system and to estimate such descriptions exist (Ang, Chong, & Li, 2005). This case study concerns data collected from step test using LabVIEW, and the data are used to identify the system transfer function as shown in Figure 1(a)–1(c)

$$G_p(s) = \frac{3.21}{1.91s + 1}. \quad (1)$$

After deriving the transfer function model, the controller has to be designed for maintaining the system to the optimal set point. This can be achieved by properly selecting the tuning parameters k_p , k_i and k_d for a PID controller. The purpose of this paper is to investigate an optimal controller design using the FA. The initial values of PID gain are calculated using the conventional ZN tuning method. These calculated values of controller parameters can be used as initial values for FA. Being a hybrid approach, the optimum value of gain is obtained using the heuristic algorithm. The advantages of using heuristic techniques for PID are optimizing the design criteria such as gain margin and phase margin. Closed-loop band width when the system is subjected to step and load change. Heuristic techniques such as FA have proved their excellence in giving better results by improving the steady-state characteristics and performance indices.

3. Firefly algorithm

The flashing light of fireflies is an amazing sight in the summer sky in the tropical and temperate regions. There are about 2000 firefly species, and most fireflies produce short and rhythmic flashes. The pattern of flashes is often unique for a particular species. The flashing light is produced by a process of bioluminescence, and the true functions of such signaling systems are still debating. However, two fundamental functions of such flashes are to attract mating partners (communication), and to attract potential prey. In addition, flashing may also serve as a protective warning mechanism. The rhythmic flash, the rate of flashing and the amount of time form part of the signal system that brings both sexes together. Females respond to a male's unique pattern of flashing in the same species, while in some species such as photuris, female fireflies can mimic the mating flashing pattern of other species so as to lure and eat the male fireflies who may mistake the flashes as a potential suitable mate.

We know that the light intensity at a particular distance r from the light source obeys the inverse square law. That

is to say, the light intensity I decreases, as the distance r increases in terms of $I \propto 1/r^2$. Furthermore, the air absorbs light which becomes weaker and weaker as the distance increases. These two combined factors make most fireflies visible only to a limited distance, usually several hundred meters at night, which is usually good enough for fireflies to communicate (Yang, 2009). The flashing light can be formulated in such a way that it is associated with the objective function to be optimized, which makes it possible to formulate new optimization algorithms. In the rest of this paper, we will first outline the basic formulation of the FA and then discuss the implementation as well as its analysis in detail.

Now we can idealize some of the flashing characteristics of fireflies so as to develop firefly-inspired algorithms (Yang, 2010a). For simplicity in describing our new FA, we now use the following three idealized rules:

- (1) All fireflies are unisex so that one firefly will be attracted to other fireflies regardless of their sex.
- (2) Attractiveness is proportional to their brightness; thus, for any two flashing fireflies, the less bright one will move toward the brighter one. The attractiveness is proportional to the brightness and they both decrease as their distance increases. If there is no brighter one than a particular firefly, it will move randomly.
- (3) The brightness of a firefly is affected or determined by the landscape of the objective function.

For a maximization problem, the brightness can simply be proportional to the value of the objective function (Sayadi, Ramezani, & Ghaffari-Nasab, 2010; Yang, 2010b). In this paper, the objective function defined to minimize the ISE based on changing the values of PID parameters k_p , k_i and k_d . Based on these three rules, the basic steps of the FA can be summarized as the pseudo-code shown.

Begin

Define

light absorption coefficient γ

initial attractiveness β_0

randomization parameter α

Objective function $f(X)$, $X = (X_1, \dots, X_d)^T$

Generate initial population of fireflies X_i ($i = 1, 2, \dots, n$)

Light intensity I_i at x_i is determined by $f(X_i)$

while ($t < \text{MaxGeneration}$)

for $i = 1 : n$ all n fireflies

for $j = 1 : i$ all n fireflies

if ($I_j > I_i$), Move firefly i towards j in d -dimension;

end if

Attractiveness varies with distance r via $\exp[-\gamma r^2]$

Evaluate new solutions and update light intensity

end for j

end for i

Rank the fireflies and find the current best

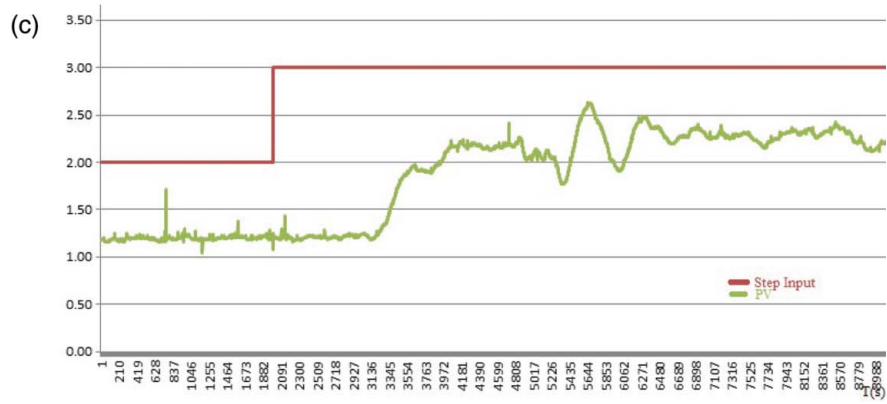
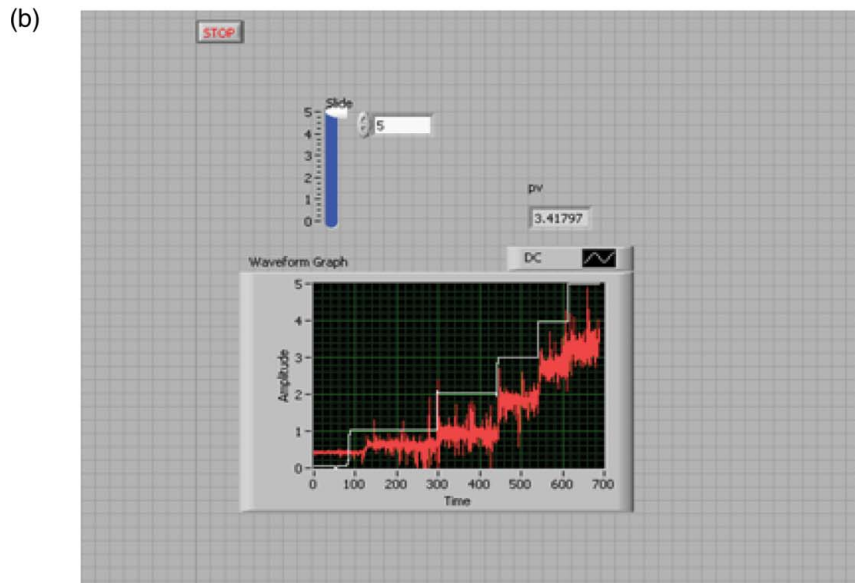
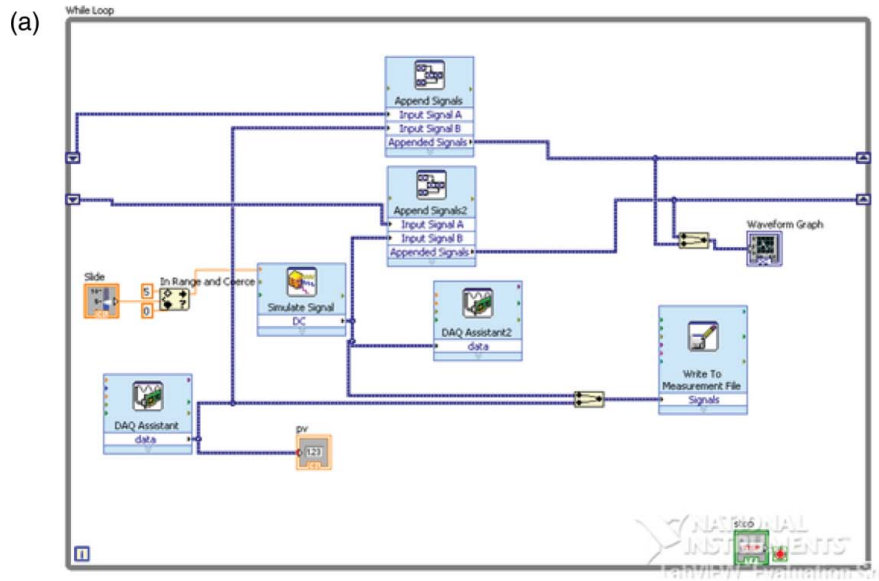


Figure 1. (a) LabVIEW block diagram for system identification. (b) LabVIEW front panel diagram for system identification. (c) Open-loop response.

end while

Postprocess results and visualization

end begin

In this algorithm, each firefly has a location $X = (x_1, x_2, x_3 \dots x_d)^T$ in a d -dimensional space and a light intensity $I(x)$ or attractiveness $\beta(x)$, which are proportional to the objective function $f(x)$. Attractiveness $\beta(x)$ and light intensity $I(x)$ are relative and these should be judged by the other fireflies. Thus, it will vary with the distance r_{ij} between firefly i and firefly j . Hence, attractiveness β of a firefly can be defined by

$$\beta = \beta_0 e^{-\gamma r^2}, \quad (2)$$

where r is the distance between any two firefly i and j at x_i and x_j , respectively, and is the Cartesian distance, β_0 is attractiveness at $r = 0$ and is the light absorption coefficient in the environment. The initial solution is generated based on

$$x_j = \text{rand}(\text{Ub} - \text{Lb}) + \text{Lb} \quad (3)$$

also each firefly i can move toward another more attractive (brighter) firefly j by

$$X_i^{t+1} = X_i^t + \beta \exp[-\gamma r_{ij}^2] + \alpha_t \left(\text{rand} - \frac{1}{2} \right), \quad (4)$$

where α is a significance factor of the randomization parameter and rand with uniform distribution $U(0, 1)$ is a random number obtained from the uniform distribution and is a random generator. The distance $r_{i,j}$ between any two fireflies i and j at x_i and x_j , respectively, is defined as the Cartesian

Table 2. Dynamic performance specifications.

Tuning method	Dynamic performance specifications			Performance index
	$T_r(s)$	$T_s(s)$	$M_p(\%)$	ISE
ZN	0.7538	6.38	37.5	11.2108
FA	0.395	0.61	0.9	0.2508

distance,

$$r_{i,j} = \|X_i - X_j\| = \sqrt{\sum_{k=1}^d (X_{i,k} - X_{j,k})^2}, \quad (5)$$

where $x_{i,k}$ is the k th component of the spatial coordinate X_i of the i th firefly.

4. Results and discussion

The newly formed PID controller is placed in a unity feedback loop with the system transfer function. This will result in a reduction in the compilation time of the program. The system transfer function is defined in another file and imported as a global variable. The controlled system is then given a step input and the error is assessed using an error performance criterion such as ISE. The equation to find out the ISE is given below in Equation (6)

$$\text{ISE} = \int_0^T e^2(t) dt, \quad (6)$$

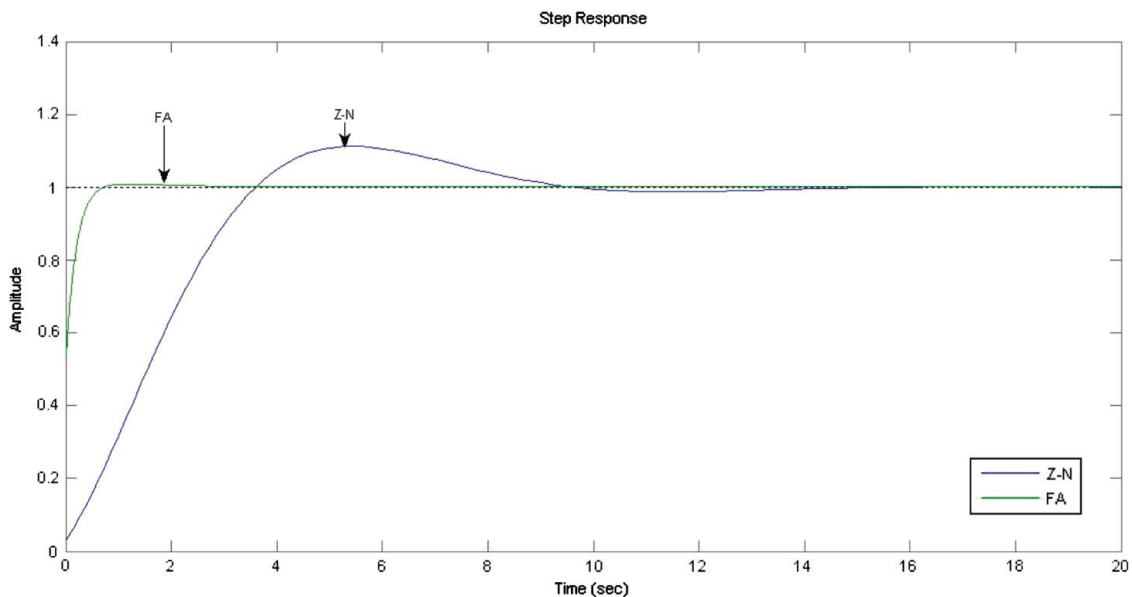


Figure 2. Tuned response of the system.

where $e(t)$ is the error calculated and t is time in seconds.

The parameters of the FA are tuned based on several pre-tests according to the problem considered here. After tuning procedures, the main FA parameters are set to the optimal settings $\beta_0 = 0.2$, $\gamma = 1.0$, $\alpha = 0.2$ and the number of fireflies = 20, number of iterations = 100. Because of the characteristics of the applied algorithms, a series of 100 runs are performed and the best results are presented. The numerical value of controller parameters and performance indices are presented in Table 2 and the graphical results of control system performance for flow process are presented in Figure 2.

5. Conclusion

The ZN and FA tuning methods have been implemented on flow control loop and a comparison of control performance using these methods has been completed. For the Z-N controller, set point tracking performance is characterized by the lack of smooth transition and by the more oscillations it has. Also, it takes much time to reach the set point. The soft computing-based controller tracks the set point faster and maintains a steady state. Also, the ISE is found to be very minimal compared with the Z-N. It was found for all control loops, the performance of the soft computing-based controller was much superior to the Z-N control. Soft computing techniques are often criticized for two reasons: Algorithms are computationally heavy and convergence to the optimal solution cannot be guaranteed. PID controller tuning is a small-scale problem and thus computational complexity is not really an issue here. It took only a couple of seconds to solve the problem. Compared with the conventionally tuned system, the FA tuned system has good steady-state response and performance indices.

References

- Ang, K. H., Chong, G., & Li, Y. (2005, July). PID control system analysis, design, and technology. *IEEE Transactions on Control Systems Technology*, 13(4), 559–576.
- Åström, K. J., & Hägglund, T. (1995). *PID controllers: Theory, design and tuning*. 2nd ed. Durham, NC: Instrument Society of America, Research Triangle Park.
- dos Santos Coelho, L., & Mariani, V. C. (2012, October). Firefly algorithm approach based on chaotic Tinkerbell map applied to multivariable PID controller tuning. *Computers and Mathematics with Applications*, 64(8), 2371–2382.
- Nagaraj, B., & Vijayakumar, P. (2011, February). A comparative study of PID controller tuning using GA, EP, PSO and ACO. *Journal of Automation, Mobile Robotics & Intelligent Systems*, 5(2), 42–48.
- Nagaraj, B., & Vijayakumar, P. (2012, April–June). Evolutionary computation based controller Tuning a comparative approach. *Indian Pulp and Paper Technical Association Journal*, 24(2), 85–90.
- Roeva, O., & Slavov, T. (2012, September). *Firefly algorithm tuning of PID controller for glucose concentration control during E. coli fed-batch cultivation process*. Proceedings of the Federated Conference on Computer Science and Information Systems, Wroclaw, Poland.
- Sayadi, M. K., Ramezani, R., & Ghaffari-Nasab, N. (2010). A discrete firefly meta-heuristic with local search for makespan minimization in permutation flow shop scheduling problems. *International Journal of Industrial Engineering Computations*, 1, 1–10.
- Solihin, M. I., Tack, L. F., & Kean, M. L. (2011). *Tuning of PID controller using particle swarm optimization*. Proceeding of the International Conference on Advanced Science, Engineering and Information Technology 2011, Putrajaya, Malaysia, pp. 458–461.
- Yang, X.-S. (2009). *Firefly algorithms for multimodal optimization*. Stochastic Algorithms: Foundations and Applications, SAGA 2009, Vol. 5792, Sapporo, Japan, pp. 169–178.
- Yang, X. S. (2010a). *Nature-inspired metaheuristic algorithms*. Frome: Luniver Press.
- Yang, X. S. (2010b). *Engineering optimization: An introduction with metaheuristic applications*. Hoboken, NJ: John Wiley & Sons.