Chapter 5

A New Methodology for Tuning PID-Type Fuzzy Logic Controllers Scaling Factors Using Genetic Algorithm of a Discrete-Time System

Wafa Gritli, Hajer Gharsallaoui and
Mohamed Benrejeb

Additional information is available at the end of the chapter

http://dx.doi.org/10.5772/intechopen.69604

Abstract

In this chapter, a proportional-integral derivative (PID)-type fuzzy logic controller (FLC) is proposed for a discrete-time system in order to track a desired trajectory generated using the flatness property. In order to improve the performance of the proposed controller, genetic algorithm (GA) based on minimizing the integral of the squared error (ISE) is used for tuning the input and output PID-type FLC scaling factors online. The considered controller is applied to an electronic throttle valve (ETV). GA tuning shows a better and robust performance compared to Simulink design optimization (SDO) algorithm in terms of tracking a desired trajectory with disturbances rejection.

Keywords: PID-type FLC, scaling factors, genetic algorithm, integral of the squared error, Simulink design optimization technique, flatness, electronic throttle valve

1. Introduction

Fuzzy logic control (FLC) has been widely used in many successful industrial applications. The first FLC algorithm was implemented by Mamdani in 1974. Unlike conventional control, which is based on mathematical model of a plant, an FLC usually embeds the intuition and experience of a human operator and may provide a nonlinear relationship induced by membership functions, rules and defuzzification. In that respect, FLC has been reported to be successfully used for a number of complex and nonlinear systems and are proved to be more robust and their performances are less sensitive to parametric variations than conventional controllers.

In the literature, various types such as proportional integral (PI), proportional derivative (PD) and proportional-integral derivative (PID) of FLCs have been proposed. For example, PI-type FLCs have been successfully implemented in many physical applications such the control of the temperature and pressure of a steam engine and control the steering and speed of an automobile. However, performance of PI-type FLCs for higher order systems and nonlinear systems may be poor due to the large overshoot and the excessive oscillation. PD-type FLCs are suitable for a limited class of systems and they are not recommendable in the presence of measurement noise and sudden load disturbances. Theoretically, PID-type FLCs provide a good performance. However, there are difficulties associated with the generation of an efficient rule base and the tuning of parameters.

In the proposed PID-type FLC, the design of parameters within two groups: structural parameters and tuning parameters. Basically, structural parameters include input/output (I/O) variables to fuzzy inference, fuzzy linguistic sets, membership functions, fuzzy rules, inference mechanism and defuzzification mechanism, which are usually determined during offline design. Tuning parameters include I/O scaling factors (SF) and parameters of membership functions (MF), which can be calculated during online adjustments of the controller in order to enhance the process performance [1].

The appropriate selection of input and output scaling factors is very important because they have significant effects on the dynamic of fuzzy controller. This leads researchers to explore the best method in searching optimum PID-type FLC parameters. Various strategies or methods have been used up to now. In Ref. [2], Qiao and Mizumoto proposed a peak observer mechanism-based method to adjust the PID-type FLC parameters. This self-tuning mechanism decreases the equivalent integral control component of the fuzzy controller gradually with the system response process time. Furthermore, Woo et al. [3] developed a method based on two empirical functions evolved with the system's error information. In Ref. [1], Guzelkaya et al. proposed a technique that adjusts the scaling factors, corresponding to the derivative and integral components, using a fuzzy inference mechanism. However, the major disadvantages of all these PID-type FLC tuning method are the difficult choice of their relative parameters and mechanisms. To overcome these difficulties, differential search algorithm (DSA) meta-heuristic technique is proposed for systematically tuning the scaling factors of the PID-type FLC in Ref. [4]. The fuzzy control design is formulated as a constrained optimization problem, which is efficiently solved based on an improved DSA. In this proposed technique, different optimization criteria such as integral square error (ISE) and maximum overshoot are considered in order to guarantee more robustness and performance control objectives.

In this chapter, a genetic algorithm (GA)-based heuristic optimization technique has been implemented to obtain better performance compared to the Simulink design optimization (SDO) technique. GA is based upon minimizing the error between the output system and the desired trajectory starting from a flat output variable generated using the flatness property. Various performance indices can be used. In this study, the integral of squared error (ISE) index has been used in order to minimize the error between the output and the desired flat trajectory. The methods are applied in a discrete-time framework to an electronic throttle valve as a case of study.

2. PID-type fuzzy logic controller description

In this study, we will deal with fuzzy PID-type controllers formed using one PD-type FLC with an integrator at the output.

The PID-type fuzzy logic controller structure is shown in **Figure 1** [5], where K_e and K_d ($K_e, K_d \in \mathbb{R}^+$) are the input scaling factors are α and β (α, $\beta \in \mathbb{R}^+$) the output scaling factors.

The inputs variables, well known as the error e_k between the desired trajectory y_k^d and the measure y_k, as well as the error variation Δe_k given by Eqs. (1) and (2) where T_e is the sampling period.

$$e_k = y_k^d - y_k \tag{1}$$

$$\Delta e_k = \frac{e_k - e_{k-1}}{T_e} \tag{2}$$

The output variable Δu_k of such a controller is the variation of the control signal u_k which can be defined as Eq. (3).

$$\Delta u_k = \frac{u_k - u_{k-1}}{T_e} \tag{3}$$

The output of the PID-type fuzzy is given by Eq. (4) [5]

$$\begin{aligned} u_k &= \alpha \Delta u_k + \beta \int \Delta u_k dt \\ &= \alpha(A + PK_e e_k + DK_d \Delta e_k) \\ &\quad + \beta \int (A + PK_e e_k + DK_d \Delta e_k) dt \\ &= \alpha A + \beta A t + (\alpha K_e P + \beta K_d D) e_k \\ &\quad + \beta K_e P \int e_k dt + \alpha K_d D \Delta e_k \end{aligned} \tag{4}$$

Thus, the equivalent control components of the PID-type FLC are such that

Proportional gain: $\alpha K_e P + \beta K_d D$

Integral gain: $\beta K_e P$

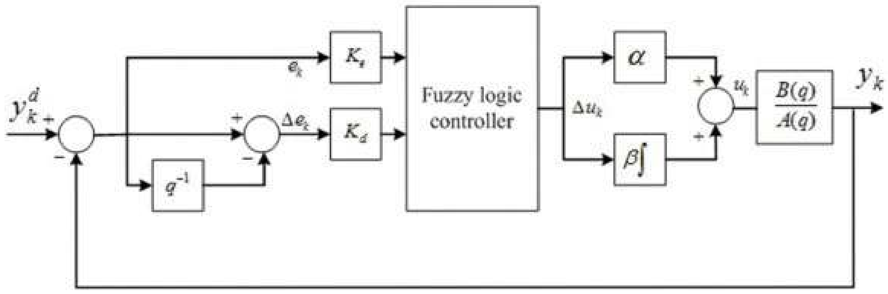

Figure 1. PID-type FLC.

$e_k \backslash \Delta e_k$	N	ZR	P
N	NL	N	ZR
ZR	N	ZR	P
P	ZR	P	PL

Table 1. Fuzzy rules-base.

Derivative gain: $\alpha K_d D$

where the terms P and D are given by Eqs. (5) and (6) [2].

$$P = \frac{\Delta u_{(i+1)j} - \Delta u_{ij}}{e_{i+1} - e_i} \tag{5}$$

$$D = \frac{\Delta u_{i(j+1)} - \Delta u_{ij}}{\Delta e_{j+1} - \Delta e_j} \tag{6}$$

The fuzzy controllers with a product-sum inference method, centroid defuzzification method and triangular uniformly distributed membership functions for the inputs and a crisp output proposed in Refs. [2, 6] are used in our case of study.

Table 1 gives the linguistic levels, assigned to the variables e_k, Δe_k and Δu_k, as follows: *NL*: negative large; *N*: negative; *ZR*: zero; *P*: positive; *PL*: positive large.

3. Scaling factors tuning using genetic algorithm

In this work, a new method is proposed for tuning the coefficients of PID-type FLCs. This method adjusts the input scaling factor corresponding to the derivative coefficient and the output scaling factor corresponding to the integral coefficient of the PID-type FLC using genetic algorithm, as shown in **Figure 2**. The integral of squared error (ISE) index has been used in order to minimize the error between the output and the desired flat trajectory.

3.1. Genetic algorithm

Genetic algorithm was first proposed by Holland [7]. It is a heuristic optimization technique inspired by the mechanism of natural selection. It is used in order to solve highly complex problems. GA starts with an initial population containing a number of parameters, where each one is regarded as the genes of a chromosome and can be structured by a string of concatenated values. Each chromosome represents a solution of the problem and its performance is evaluated based on fitness function.

In the beginning, an initial chromosome population is randomly generated. The chromosomes are candidate solutions to the problem. Then, the fitness values of all chromosomes are evaluated by calculating the objective function. So, a group of the best chromosomes is selected based on the fitness of each individual. In this 'surviving' population, the genetic operators of

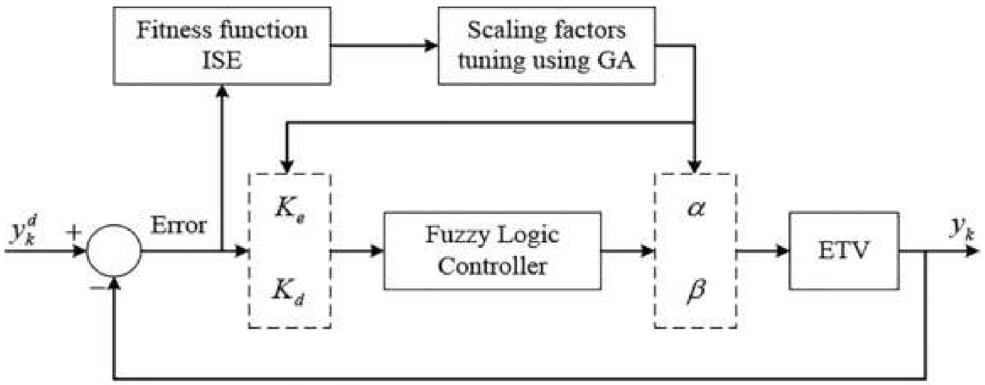

Figure 2. PID-type FLC scaling factors tuning.

crossover and mutation are applied in order to create the next population solution. The above steps are repeated until a specific termination criterion is found.

- **Reproduction**: Create a part of the new population by simply copying without changing the selected individuals from the present population. Also, new population has the possibility of selection by already developed solutions [8].
- **Crossover**: Create new individuals as offspring of two parents. It is a recombination operator that combines selected subparts called crossover points of two parent chromosomes. The individuals resulting in this way are the offspring [8].
- **Mutation**: Create a new individual for the new population by randomly mutating a selected individual. The modifications can consist of changing one or more values in the representation or adding/deleting parts of the representation [8].

To compute the fitness of each chromosome, the objective functions are used. Many authors use integral of time multiplied by absolute error (ITAE), mean of the squared error (MSE), integral of absolute error (IAE) and integral of the squared error (ISE) as performance index [9, 10].

In this chapter, the method of tuning PID-type FLC parameters using GA consists in finding the optimal I/O scaling factors, which minimize the defined objective function, chosen as the ISE in order to specify more performance in terms of tracking a desired trajectory.

If $y^d(t)$ is the desired trajectory and y is the output trajectory, then error *e(t)* is

$$e(t) = y^d(t) - y(t) \tag{7}$$

and the ISE can be defined by

$$ISE = \int_0^{\tau} e(t)^2\,dt \tag{8}$$

Fitness function is taken as inverse of error, i.e., performance index.

$$Fitness\ value = \frac{1}{Performance\ index} \tag{9}$$

3.2. Tuning procedure

The overall flowchart for optimization using GA is shown in **Figure 3**. Initially, a number of populations N have been generated for the scaling factors K_e, K_d, α and β. Each individual of these N sets in the current population is evaluated using the objective function ISE. Based on the values of the objective function, out of these N possible solutions, the good solutions are retained and the others are eliminated. A new population is formed by applying the genetic operators (reproduction, crossover and mutation) to these selected individuals. This process of

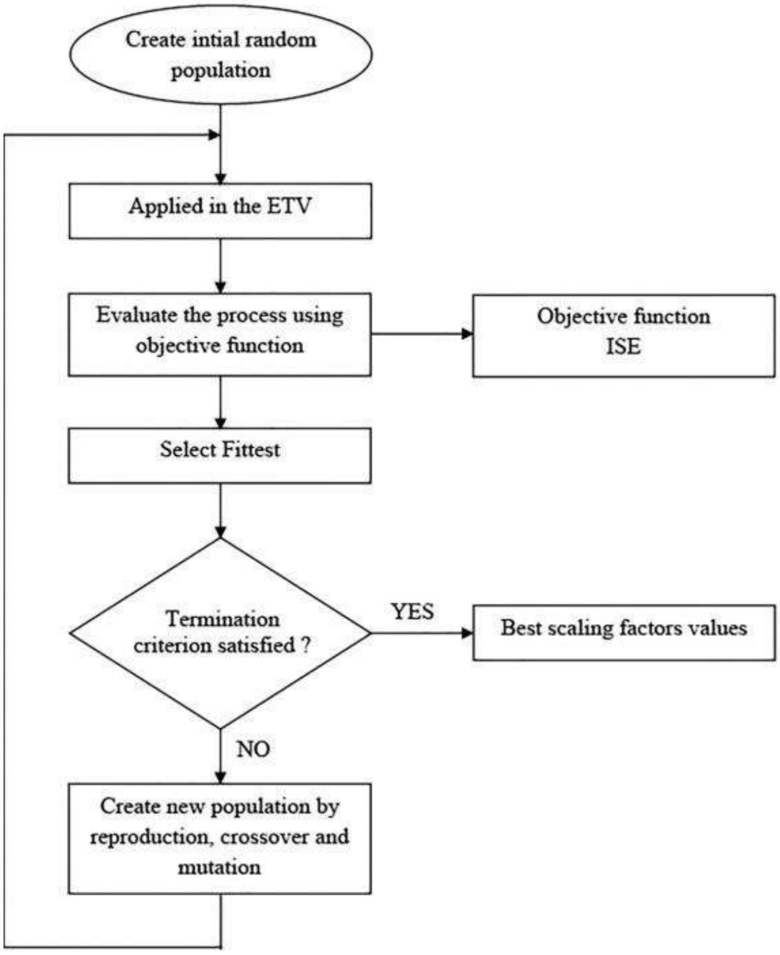
Create intial random population
Applied in the ETV
Evaluate the process using objective function
Objective function ISE
Select Fittest
Termination criterion satisfied ?
YES
Best scaling factors values
NO
Create new population by reproduction, crossover and mutation

Figure 3. Flowchart of the GA optimization algorithm.

production of a new generation and its evaluation is performed repetitively. The algorithm continues until the population converges to the stop criterion.

4. Flatness and trajectory planning

The flat property has been introduced in Ref. [11] for continuous-time nonlinear systems. It can be stated in a discrete-time version which leads for the design of a control which ensures a tracking of a desired trajectory. One major property of differential flatness is that the state and the input variables can be directly expressed, without integrating any differential equation, in terms of the flat output and a finite number of its derivatives. The flatness approach will be used in this chapter in a discrete-time framework.

The studied dynamic linear discrete system is described by Eq. (10).

$$A(q)y_k = B(q)u_k \tag{10}$$

where q is the forward operator, u_k and y_k are the input and the output, respectively, and $A(q)$ and $B(q)$ are polynomials defined by

$$A(q) = q^n + a_{n-1}q^{n-1} + \ldots + a_1 q + a_0 \tag{11}$$

$$B(q) = b_{n-1}q^{n-1} + \ldots + b_1 q + b_0 \tag{12}$$

where the parameters a_i and b_i are constants, $i=0,1,\ldots,n-1$. The partial state of such a dynamic system can be considered as a discrete flat output z_k which can be expressed as a function of input and output signals as following

$$A(q)z_k = u_k \tag{13}$$

$$B(q)z_k = y_k \tag{14}$$

Often, the real output signal y_k to be controlled is not a flat output. Then, it is necessary to plan a desired trajectory for the flat output [11] and to consider thereafter the relation (14).

The open loop control law can be determined by the following relations [12].

$$u^d(t) = f(z^d(t),\ldots,z^{d^{(r+1)}}(t)) \tag{15}$$

$$y^d(t) = g(z^d(t),\ldots,z^{d^{(r)}}(t)) \tag{16}$$

where f and g are the vectorial functions. Then, it is sufficient to find a desired continuous flat trajectory $t \mapsto z^d(t)$ that must to be differentiable at the $(r+1)$ order.

The polynomial interpolation technique is used in order to plan the desired flat trajectory $z^d(t)$. Let consider the state vector $Z^d(t) = (z^d(t) \;\; \dot{z}^d(t) \;\; \ldots \;\; z^{d^{(r+1)}}(t))^T$ containing the desired continuous flat output and its successive derivatives. The expression of $Z^d(t)$ can be given in Eq. (17); t_0 and t_f are the two moments known in advance.

$$Z^d(t) = M_1(t-t_0)c_1(t_0) + M_2(t-t_0)c_2(t_0,t_f) \tag{17}$$

where M_1 and M_2 are such as [12]

$$M_1 = \begin{pmatrix} 1 & t & \cdots & \frac{t^{n-1}}{(n-1)!} \\ 0 & 1 & \cdots & \frac{t^{n-2}}{(n-2)!} \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & 1 \end{pmatrix} \tag{18}$$

$$M_2 = \begin{pmatrix} \frac{t^n}{n!} & \frac{t^{n+1}}{(n+1)!} & \cdots & \frac{t^{2n-1}}{(2n-1)!} \\ \frac{t^{n-1}}{(n-1)!} & \frac{t^n}{n!} & \cdots & \frac{t^{(n-2)}}{(n-2)!} \\ \vdots & \ddots & \ddots & \vdots \\ t & \cdots & \frac{t^{n-1}}{(n-1)!} & \frac{t^n}{n!} \end{pmatrix} \tag{19}$$

and the vectors c_1 and c_2 defined by

$$c_1 = Z^d(t_0) \tag{20}$$

$$c_2 = M_2^{-1}(t_f - t_0)(Z^d(t_f) - M_1(t_f - t_0)Z^d(t_0)) \tag{21}$$

Then, the output desired trajectory y_k^d is defined. In the discrete-time framework, the real output y_k has asymptotically to track this such as Eq. (22).

$$y_k^d = B(q)z_k^d \tag{22}$$

In the following section of this chapter, the efficiency of the proposed methodology for tuning PID-type FLC scaling factors has been validated on a discrete-time system: an electronic throttle valve for a defined desired trajectory generated using the flatness property and compared to the Simulink design optimization (SDO) technique.

5. Case of study: electronic throttle valve (ETV)

Throttle valve is one of the most important devices in the engine management system. In conventional engine, the amount of airflow into the combustion system has been adjusted by the throttle valve, which is connected mechanically to an accelerator pedal [13]. The electronic throttle body (ETB) regulates air inflow into the car engine. Compared to the mechanical throttle, a well-controlled ETB can reduce fuel consumption.

5.1. System modelling

The case of the ETV is described in **Figure 4**.

The electrical part is modelled by Eq. (23)

$$u(t) = L\frac{d}{dt}i(t) + Ri(t) + k_v\omega_m(t) \tag{23}$$

where L is the inductance R is the resistance $u(t)$ and $i(t)$ are the voltage and the armature current, respectively, k_v is an electromotive force constant and $\omega_m(t)$ is the motor rotational speed.

The mechanical part of the throttle is modelled by a gear reducer characterized by its reduction ratio γ such as Eq. (24)

$$\gamma = \frac{C_g}{C_L} \tag{24}$$

where C_L is the load torque and C_g is the gear torque. The mechanical part is modelled according to Eq. (25), such that [14, 15].

$$J\frac{d}{dt}\omega_m(t) = C_e - C_f - C_r - C_a \tag{25}$$

and

$$\frac{d}{dt}\theta(t) = (180/\pi/\gamma)\omega_m(t) \tag{26}$$

where $\theta(t)$ is the throttle plate angle, J is the overall moment of inertia, $C_e = k_e i(t)$ is the electrical torque where K_e is a constant, C_f is the torque caused by mechanical friction, C_r is the spring resistive torque and C_a is the torque generated by the airflow. The electronic throttle valve involves two complex nonlinearities due to the nonlinear spring torque Cr and the friction

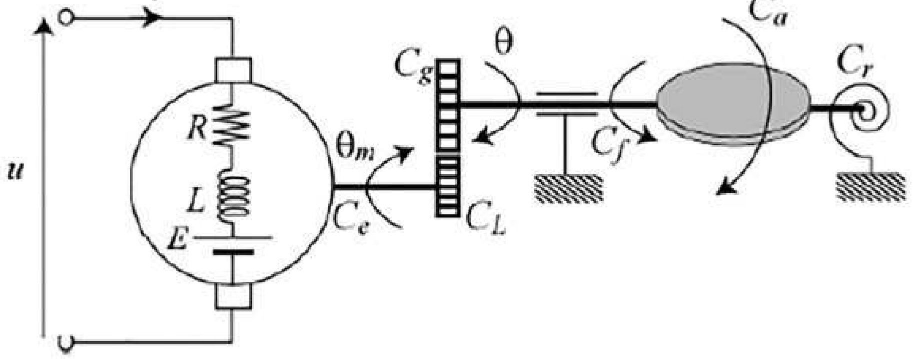

Figure 4. Electronic throttle valve system.

torque C_f. They are given by their static characteristics [16]. The static characteristic of the nonlinear spring torque C_r is defined by

$$C_r = \frac{k_r}{\gamma}(\theta - \theta_0) + D\,\mathrm{sgn}(\theta - \theta_0) \tag{27}$$

For $\theta_{\min} \le \theta \le \theta_{\max}$, k_r, is the spring constant, D is a constant, θ_0 is the default position and sgn(.) is the following signum function

$$\mathrm{sgn}(\theta - \theta_0) = \begin{cases} 1, \text{ if } \theta \ge \theta_0 \\ -1, \text{ else} \end{cases} \tag{28}$$

The friction torque function C_f of the angular velocity of the throttle plate can be expressed as

$$C_f = f_v\omega + f_c\,\mathrm{sgn}(\omega) \tag{29}$$

where f_v and f_c are two constants. By substituting in Eq. (25) the expressions of C_g, C_f and C_r and by neglecting the torque generated by the airflow C_a, the two nonlinearities $\mathrm{sgn}(\theta - \theta_0)$ and $\mathrm{sgn}(\omega)$ and the two constants $\frac{k_r}{\gamma}\theta_0$ and f_v the transfer function of the linear model becomes (30) [15].

$$H(q) = \frac{(180/\pi/\gamma)k_e}{JLq^3 + JRq^2 + (k_e k_v + Lk_s)q + Rk_s} \tag{30}$$

with $k_s = (180 / \pi / \gamma^2)k_r$ and q as the Laplace operators.

5.2. Simulation results

The identified parameters of $H(q)$ are given in **Table 2** at 25°C temperature [15].

The corresponding discrete-time transfer function is given by Eq. (31) for the sample time $T_e = 0.002$s.

$$H(q^{-1}) = \frac{0.007833q^{-1} + 0.01396q^{-2} + 0.0007724q^{-3}}{1 - 1.948q^{-1} + 0.954q^{-2} - 0.006152q^{-3}} \tag{31}$$

Parameters	Values
R(Ω)	2.8
L(H)	0.0011
ke (N.m/A)	0.0183
kv (v/rad/s)	0.0183
J (kg.m^2)	4×10^{-6}
γ	16.95

Table 2. Model's parameters.

The desired continuous time flat trajectory $z^d(t)$ can be computed according to the following polynomial form

$$z^d(k) = \begin{cases} \dfrac{cst1}{B(1)}, & \text{if } 0 \le k \le k_0 \\ Poly_1(k), & \text{if } k_0 \le k \le k_1 \\ \dfrac{cst2}{B(1)}, & \text{if } k_1 \le k \le k_2 \\ Poly_2(k), & \text{if } k_2 \le k \le k_3 \\ \dfrac{cst1}{B(1)}, & \text{if } k \ge k_3 \end{cases} \qquad (32)$$

where $cst1$ and $cst2$ are constants, $k_0 = 3s$, $k_1 = 6s$, $k_2 = 10s$ and $k_3 = 15s$ are the instants of transitions, $B(1)$ is the static gain between the flat output z_k and the output signal y_k for each operating mode and $Poly_1(k)$ and $Poly_2(k)$ are polynomials calculated using the technique of polynomial interpolation.

The desired trajectory is then given in **Figure 5**.

The obtained optimal I/O scaling factors (Ke, Kd, α, β) for Simulink design optimization technique and GA are summarized in **Table 3**.

The obtained results are given in **Figures 6–9**.

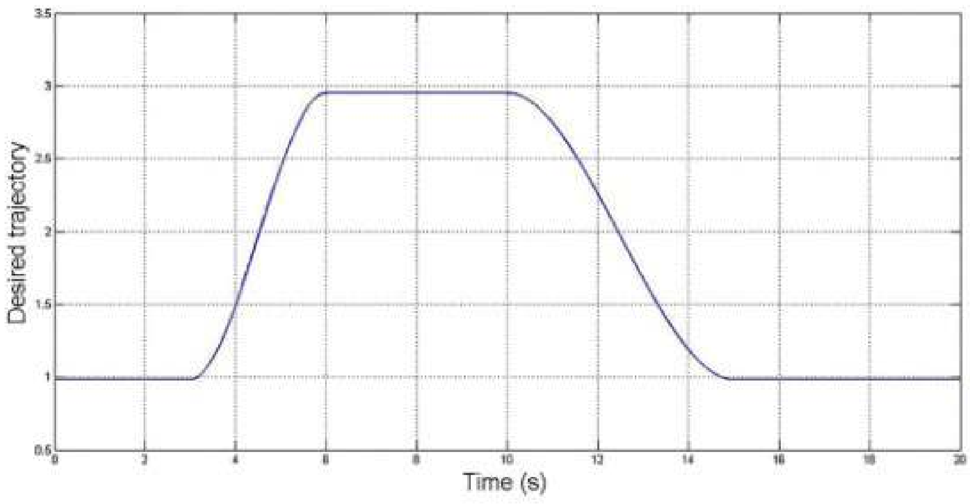

Figure 5. Desired trajectory.

SF\Method	SDO	GA
Ke	0.1144	0.0086
Kd	1.5997	0.4612
α	2.6239	0.1108
β	0.0001	0.1934

Table 3. PID-type fuzzy scaling factors values.

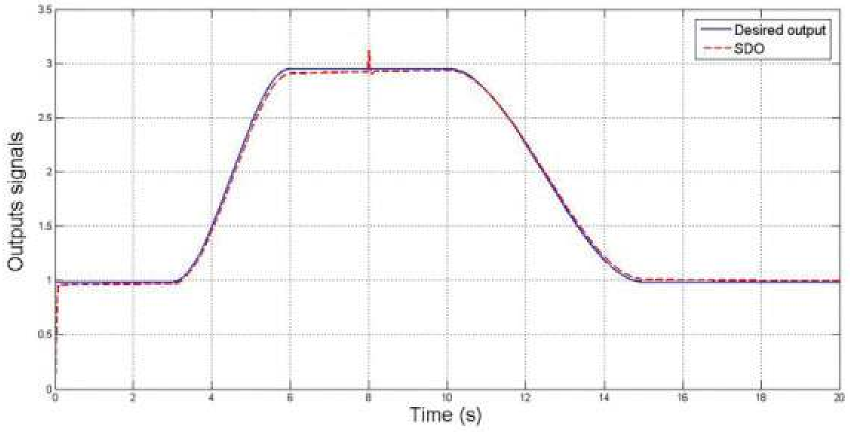

Figure 6. System outputs using Simulink design optimization.

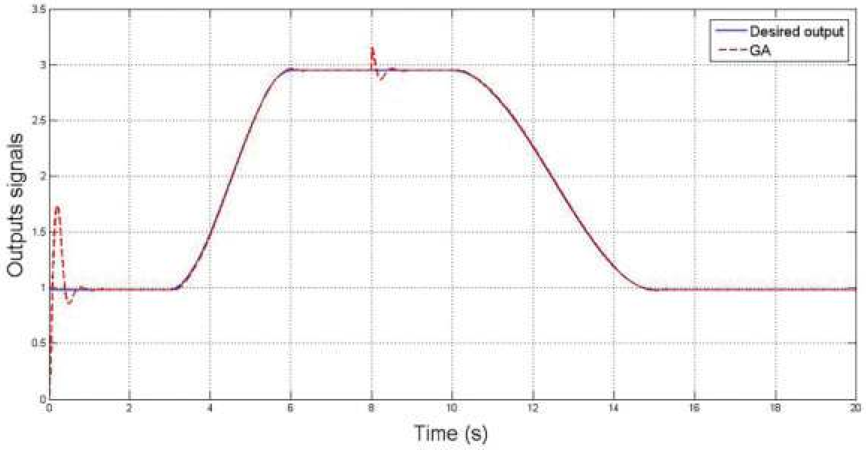

Figure 7. System outputs using genetic algorithm.

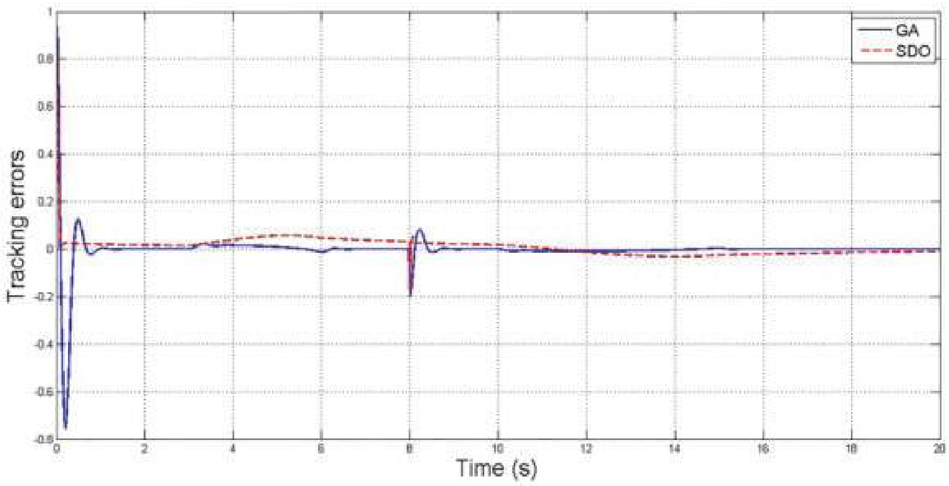

Figure 8. Tracking errors.

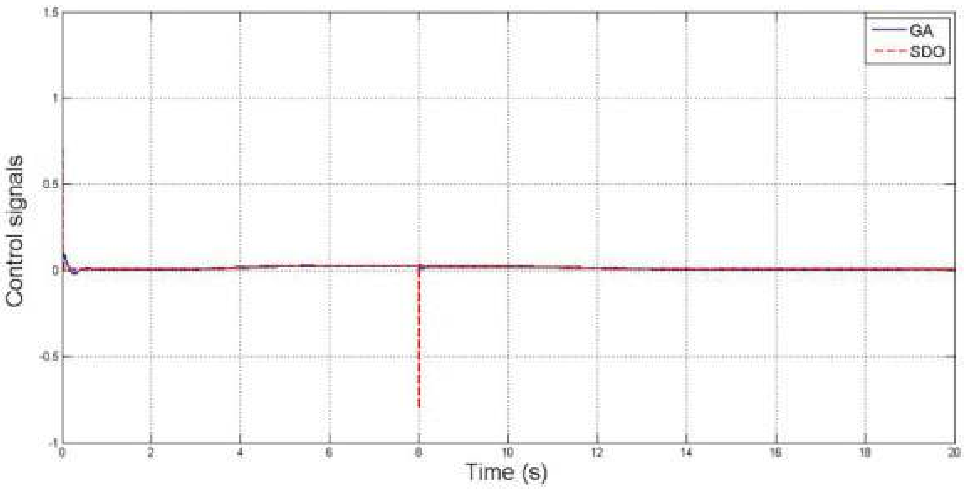

Figure 9. Control signals.

Figures 8 and **9** show the responses with Simulink design optimization technique and GA tuning using ISE criterion. Based on a comparative analysis, better results were there obtained with the GA tuning method.

All results, for obtained scaling factors values, are acceptable and show the effectiveness of the proposed GA tuning method in terms of the tracking desired trajectory with disturbances rejection in comparison with the SDO technique.

6. Conclusion

In this chapter, an optimization technique was introduced to tune the parameters of PID-type fuzzy logic controller (FLC). The idea is to use the genetic algorithm (GA)-based heuristic optimization technique in order to solve highly complex problems. In order to specify more robustness and performance of the proposed GA-tuned PID-type FLC, optimization criteria such as integral square error (ISE) is considered.

The proposed controller is applied to an electronic throttle valve (ETV) in the discrete-time framework in order to track a desired trajectory starting from a flat output generated using flatness property. The performance comparison with the Simulink design optimization (SDO) technique shows the efficiency of the proposed GA-tuned approach in terms of tracking a desired trajectory with disturbances rejection.

Author details

Wafa Gritli*, Hajer Gharsallaoui and Mohamed Benrejeb

*Address all correspondence to: wafa_gritli@yahoo.fr

National Engineering School of Tunis, Tunis, Tunisia

References

[1] Guzelkaya M, Eksin I, Yesil E. Self-tuning of PID-type fuzzy logic controller coefficients via relative rate observer. Engineering Applications of Artificial Intelligence. 2003;**16** (3):227–236

[2] Qiao WZ, Mizumoto M. PID type fuzzy controller and parameters adaptive method. Fuzzy Sets and Systems. 1996;**78**:23–35

[3] Woo ZW, Chung HY, Lin JJ. A PID type fuzzy controller with self-tuning scaling factors. Fuzzy Sets and Systems. 2000;**115**:321–326

[4] Toumi F, Bouallègue S, Haggège J, Siarry P. Differential search algorithm-based approach for PID-type fuzzy controller tuning. In: International Conference on Control, Engineering & Information Technology (CEIT'14); 2014. pp. 329–334

[5] Gritli W, Gharsallaoui H, Benrejeb M. PID-type fuzzy scaling factors tuning using genetic algorithm and Simulink design optimization for electronic throttle valve. In: International Conference on Control, Decision and Information Technologies (CoDIT'16); 6-8 April; Malta. 2016. pp. 216–221

[6] Galichet S, Foulloy L. Fuzzy controllers: Synthesis and equivalences. IEEE Transactions on Fuzzy Systems. 1995;**3**:140–148

[7] Holland JJ. Adaptation in Natural and Artificial Systems, University of Michigan Press; I975

[8] Kim JS, Kim JH, Park JM, Park SM, Choe WY, Heo H. Auto tuning PID controller based on improved genetic algorithm for reverse osmosis plant. International Journal of Computer, Electrical, Automation, Control and Information Engineering. 2008;**2**(11):3707–3712

[9] Gaing ZL. Particle swarm optimization approach for optimum design of PID controller in AVR system. IEEE Transactions on Energy Conversion. 2004;**19**:384–391

[10] Mahony TO, Downing CJ, Fatla K. Genetic algorithm for PID parameter optimization: minimizing error criteria. In: Process Control and Instrumentation; 26–28 July; 2000. pp. 148–153

[11] Fliess M, Levine J, Martin P, Rouchon P. On differentially flat non linear systems. In: Proc IFAC Symposium on Nonlinear Control Systems Design (NOLCOS), IFAC, Laxenburg, Austria; 1992, pp. 408–412

[12] Gharsallaoui H, Ayadi M, Benrejeb M, Borne P. Robust flatness-based multi-controllers approach. Studies in Informatics and Control. 2010;**19**(4):357–368

[13] Costin M, Schaller R, Maiorana M, Purcell J. An architecture for electronic throttle control systems. In SAE Technical Paper, SAE International; 2003.

[14] Lebbal Ml, Chafouk H, Hoblos G, Lefebvre D. Modelling and identification of non-linear systems by a multimodel approach: Application to a throttle valve. International Journal Information and Systems Science. 2007;**3**:67–87

[15] Yang C. Model-based analysis and tuning of electronic throttle controllers. In: Visteon Corporation, SAE Paper; March 8-11; Detroit, Michigan. 2004.

[16] Aidi I, Ayadi M, Benrejeb M, Borne P. Flatness-based control of throttle valve using neural observer. International Journal of Research and Surveys. 2012;**12**:333–344