Engineering Science and Technology, an International Journal 18 (2015) 194–206

HOSTED BY

ELSEVIER

Contents lists available at ScienceDirect

Engineering Science and Technology, an International Journal

journal homepage: http://www.elsevier.com/locate/jestch

Full length article

A novel optimal PID plus second order derivative controller for AVR system

Mouayad A. Sahib

Software Engineering Department, Salahaddin University, Kurdistan Regional, Erbil, Iraq

ARTICLE INFO

Article history:
Received 6 August 2014
Received in revised form
7 November 2014
Accepted 19 November 2014
Available online 6 January 2015

Keywords:
Optimal control
PID controller
Automatic voltage regulator
Particle swarm optimization
Fractional order PID

ABSTRACT

This paper proposes a novel controller for automatic voltage regulator (AVR) system. The controller is a four term control type consisting of proportional, integral, derivative, and second order derivative terms ($PIDD^2$). The four parameters of the proposed controller are optimized using particle swarm optimization (PSO) algorithm. The performance of the proposed $PIDD^2$ is compared with various PID controllers tuned by modern heuristic optimization algorithms. In addition, a comparison with the fractional order PID (FOPID) controller tuned by Chaotic Ant Swarm (CAS) algorithm is also performed. Furthermore, a frequency response, zero-pole map, and robustness analysis of the AVR system with $PIDD^2$ is performed. Practical implementation issues of the proposed controller are also addressed. Simulation results showed a superior response performance of the $PIDD^2$ controller in comparison to PID and FOPID controllers. Moreover, the proposed $PIDD^2$ can highly improve the system robustness with respect to model uncertainties.

1. Introduction

In power generation systems, automatic voltage regulator (AVR) is utilized to maintain the terminal voltage of a synchronous generator at a specified level. The AVR controls the consistency of the terminal voltage by varying the exciter voltage of the generator [1]. Due to the high inductance of the generator field windings and load variation, stable and fast response of the regulator is difficult to achieve. Therefore, it is important to improve the AVR performance and ensure stable and efficient response to transient changes in terminal voltage. Various control structures have been proposed for the AVR system, however, among these controllers the proportional plus integral plus derivative (PID) was the most preferable controller. The PID controller is distinguished by its robust performance over a wide range of operating conditions and simplicity of structure design [2]. The design of the PID controller involves the determination of three parameters which are the proportional, integral, and derivative gains. In recent years, many intelligent optimization algorithms were proposed to tune the PID gains of the AVR system. Such algorithms include Particle Swarm Optimization (PSO) [3], Genetic algorithm (GA) [3,4], Craziness based particle swarm optimization (CRPSO) [5], Reinforcement Learning Automata (RLA) [6], Artificial Bee Colony (ABC) [7], Differential Evolution Algorithm (DEA) [8], Many Optimizing Liaisons (MOL) [9], Local Unimodal Sampling (LUS) [10], and Chaotic Ant Swarm (CAS) [11]. CAS is a new search algorithm inspired by the biological behavior of ants in nature proposed by Li et al. [12]. However, it is a deterministic process different from the conventional ant algorithm [13]. It combines the chaotic behavior of individual ants with the intelligent optimization action of an ant colony and thus it integrates the advantages of chaotic search and the powerful ability of swarm collectiveness. Based on CAS algorithm, Li et al. developed a model which can be used to describe how an ant colony organizes itself to find the optimal path between a food source and the nest [14]. The CAS algorithm shows a great potential in solving difficult optimization problems encountered in various fields such as parameter identification of dynamic systems [13], fuzzy system identification [15], and parameters tuning of PID controller [11].

Recently, large and growing body of literature has investigated the concept of fractional calculus in many control applications to enhance the performance of PID controller [16–18]. Fractional order PID (FOPID) controller was first proposed by Podlubny in 1999 [19]. FOPID is a generalization of the PID in which the orders of derivatives and integrals are non-integer [20]. The application of FOPID controller was also employed to control AVR system [21–23]. Compared to conventional PID, FOPID can ensure good control

E-mail address: mouayad.sahib@gmail.com.
Peer review under responsibility of Karabuk University.

http://dx.doi.org/10.1016/j.jestch.2014.11.006

Table 1
Transfer functions of the AVR system components.

AVR component	Transfer function	Range of the gain K	Range of the time constant T (s)
Amplifier	$G_a = \frac{K_a}{T_a s+1}$	10–40	0.02–0.1
Exciter	$G_e = \frac{K_e}{T_e s+1}$	1–10	0.4–1.0
Generator	$G_g = \frac{K_g}{T_g s+1}$	0.7–1	1.0–2.0
Sensor	$H_s = \frac{K_s}{T_s s+1}$	0.9–1.1	0.001–0.06

performance and improve the system robustness with respect to model uncertainties [24]. However, due to fractional order in the differentiator and integrator, realization of FOPID is performed with high order discrete time controllers affecting the computational load and memory size of the control algorithm. Therefore, various approximation methods have been proposed to reduce the controller's order. However, the so-called long memory principle feature of the FOPID controllers will not be preserved after approximation. Another property that is lost after approximation is the optimality of controller [25].

The main contribution of this paper is to propose a novel four term structure PID plus second order derivative (PIDD^2) controller for AVR system. The four gains of the PIDD^2 are tuned by PSO algorithm. The performance of the proposed PIDD^2 is compared with some PID controllers tuned by recently published modern heuristic optimization algorithms such as MOL, GA, ABC, DEA, and LUS algorithms. In addition, a comparison with the FOPID controller tuned by CAS algorithm is also performed. The performance of the proposed PIDD^2 is further investigated using frequency response, zero-pole map, and robustness analysis.

The remaining part of this paper is organized as follows. In Section 2, the AVR system model is described. The AVR system with PID controller is analyzed in Section 3. The proposed PIDD^2 controller is presented in Section 4. The PSO algorithm is explained in section 5. In Section 6, the practical implementation issues of the PIDD^2 controller are addressed. Section 7 is devoted to computer simulation. Finally, Section 8 concludes the paper.

2. AVR system model

In synchronous generators, the AVR system is used to maintain the terminal voltage magnitude at a constant specified level. A simple AVR system consists of four main components, namely amplifier, exciter, generator, and sensor. Each component is modeled by a first order system defined by a gain and a time constant. Table 1, shows the four AVR main components transfer functions with their corresponding gain and time constants typical ranges [9].

The arrangement of the AVR system components is shown in Fig. 1. The terminal voltage $\Delta V_t(s)$ of the generator is continuously sensed by the sensor and compared with the desired reference voltage $\Delta V_{\text{ref}}(s)$. The difference between the reference and the sensed terminal voltages (error voltage $\Delta V_e(s)$) is amplified through the amplifier and used to excite the generator using the exciter. The

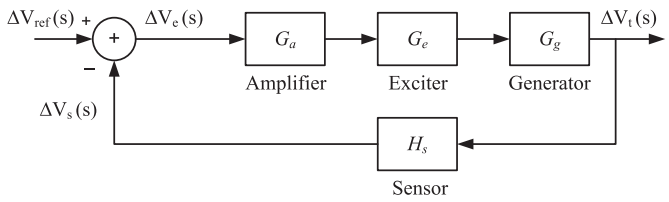

Fig. 1. AVR system block diagram.

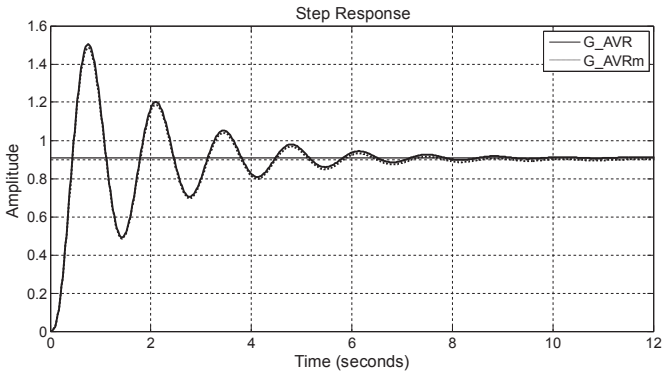

Fig. 2. Step response of the AVR system without PID controller.

AVR system parameters considered in this work are; $K_a = 10.0$, $T_a = 0.1$, $K_e = 1.0$, $T_e = 0.4$, K_g=1.0, $T_g = 1.0$, K_s=1.0, $T_s = 0.01$ [3,7,9,24,26]. With these parameter values the closed loop transfer function of the AVR system becomes:

$$\begin{aligned} G_{\text{AVR}} &= \frac{\Delta V_t(s)}{\Delta V_{\text{ref}}(s)} = \frac{0.1s+10}{0.0004s^4+0.045s^3+0.555s^2+1.51s+11} \\ &= \frac{250(s+100)}{(s+98.82)(s+12.63)(s^2+1.057s+22.04)} \\ &\cong \tilde{G}_{\text{AVR}} = \frac{250}{(s+12.63)(s^2+1.057s+22.04)} \end{aligned} \tag{1}$$

The transfer function of the AVR system (G_{AVR}) have one zero at $z = -100$, two real poles at $s_1 = -98.82$ and $s_2 = -12.63$, and two complex poles at $s_{3,4} = -0.53 \pm 4.66i$. The G_{AVR} can be approximated by canceling the zero at -100 with the pole at -98.82 to obtain $\tilde{G}_{\text{AVR}}$. The unit step responses of G_{AVR} and $\tilde{G}_{\text{AVR}}$ are shown in Fig. 2. It can be observed from Fig. 2 that the AVR system G_{AVR} and its approximation $\tilde{G}_{\text{AVR}}$ are almost similar and possess an underdamped response with steady state amplitude value of 0.909, peak amplitude of 1.5 ($M_p = 65.43\%$) at $t_p = 0.75$, $t_r = 0.42$ s, $t_s = 6.97$ s at which the response has settled to 98% of the steady state value.

3. Analysis of the AVR system with PID controller

The response of the AVR can be improved by utilizing a controller in the forward path capable of processing the voltage difference $\Delta V_e(s)$ and producing a manipulated actuating signal. Commonly, a PID controller is employed for this task due to its simple structure. The PID controller combines three control actions related to the error signal in proportional, deferential, and integral manners and its transfer function is given by:

$$C_{PID} = K_p + \frac{K_i}{s} + sK_d \tag{2}$$

where K_p, K_i, and K_d are the proportional, integral, and derivative gains, Fig. 3 shows a block diagram of the AVR system with PID controller. The general transfer function of the AVR system controlled by a PID controller is given by

$$G_{\text{AVR\_PID}} = \frac{C_{\text{PID}} G_a G_e G_g}{1 + C_{\text{PID}} G_a G_e G_g H_s} \tag{3}$$

Substituting the transfer functions of the AVR system components listed in Table 1 with their parameters and the transfer function of the PID controller given by Equation (2) in (3) yields,

$$G_{\text{AVR\_PID}} = \frac{0.1K_d s^3 + (0.1K_p + 10K_d)s^2 + (0.1K_i + 10K_p)s + 10K_i}{0.0004s^5 + 0.0454s^4 + 0.555s^3 + (1.51 + 10K_d)s^2 + (1 + 10K_p)s + 10K_i} \tag{4}$$

The effect of the PID gain parameters on the overall AVR system can be analyzed by plotting the closed loop zero-pole locus as a function of the PID gains. The zero-pole locus can be obtained when K_p, K_i, and K_d are varied within the closed ranges $1 \le K_p \le K_{p\_\max}$, $0 \le K_i \le K_{i\_\max}$, and $0 \le K_d \le K_{d\_\max}$ respectively. The initial state of the zero-pole locus can be easily obtained by setting $K_p = 1$, $K_i = 0$, and $K_d = 0$ in Equation (4) and as a result the transfer function of the AVR system reduces to that given by Equation (1) (without PID controller).

The characteristic of the transient response of the AVR system is closely related to the location of the closed-loop poles. From the design viewpoint, the adjustment of the PID gains may move the closed-loop poles to a desired location. Hence, with the use of the zero-pole locus method, it is possible to determine the values of the PID gains that will make the damping ratio of the dominant closed-loop poles as prescribed. However, a multi-gain root-locus is not an easy way to obtain and difficult to illustrate and plot on the complex plane. Alternatively, the problem of evaluating the optimum PID gains can be handled using an optimization problem in which an optimization algorithm is employed. The optimization algorithm, such as PSO, uses an objective or cost function to tune the PID gains. For example, Panda et al., proposed the simplified PSO algorithm to design a PID controller for the AVR system [9]. By investigating the zero-pole map of the overall transfer function (the AVR system with the designed PID), given by [9],

$$\begin{aligned} G_{\text{AVR}_{\text{PID}}} &= \frac{0.01772s^3 + 1.831s^2 + 5.899s + 4.189}{0.0004s^5 + 0.045s^4 + 0.555s^3 + 3.282s^2 + 6.857s + 4.189} \\ &= \frac{44.3(s + 100.03)(s + 2.26)(s + 1.05)}{(s + 100.49)(s + 2.11)(s + 1.06)(s^2 + 9.84s + 46.52)} \cong \tilde{G}_{\text{AVR}_{\text{PID}}} = \frac{46.52}{(s^2 + 9.84s + 46.52)}, \end{aligned} \tag{5}$$

one can observe that the objective of the PID controller is to compensate the effect of two poles in the AVR system at $s_1 = -2.11$ and $s_2 = -1.06$, thus the overall transfer function $G_{\text{AVR}_{\text{PID}}}$ can be approximated to $\tilde{G}_{\text{AVR}_{\text{PID}}}$. Fig. 4, shows the step responses of $G_{\text{AVR}_{\text{PID}}}$ and $\tilde{G}_{\text{AVR}_{\text{PID}}}$.

From Fig. 4, it is observable that the step response of the AVR system and its approximation has been improved when using an optimal PID controller. This is evident through an improved values of rise time $t_r = 0.343$, settling time $t_s = 0.516$ sec, maximum overshoot $M_p = 1.95\%$, and damping ratio $\zeta = 0.72$.

From the above analysis, it can be concluded that the PID controller attempts to compensate the effect of two poles of the AVR system. When the PID controller gain parameters are optimized, the overall transfer function is approximately reduced from fourth to a simple second order system. However, in a second-order system, the maximum overshoot and the rise time of the unit step response conflict with each other. Therefore, the improvement of the AVR system response achieved by the conventional PID controller is a compromise between maximum overshoot and rise time.

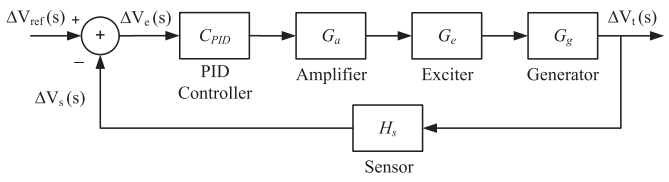

Fig. 3. AVR system with PID controller.

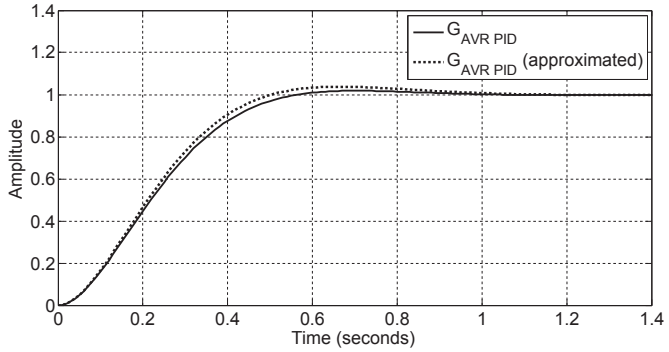

Fig. 4. Step response of the AVR system with PID controller.

4. PID plus second order derivative controller (PIDD^2)

The closed loop transfer function of the AVR system with optimized PID controller can be approximated by a standard form of a second-order system given by

$$\tilde{G}_{\text{AVR}_{\text{PID}}} = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \tag{6}$$

where ω_n is the undamped natural frequency. The proposed method is to modify the structure of the conventional PID controller such that it can reduce the overall transfer function to produce a modified form of Equation (6) in which an additional zero is added at $s = -\alpha$, such that,

$$G_z = \frac{\left(\frac{s}{\alpha} + 1\right)\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} = \frac{s}{\alpha}\tilde{G}_{\text{AVR}_{\text{PID}}} + \tilde{G}_{\text{AVR}_{\text{PID}}} \tag{7}$$

then the step response of the modified system become

$$Y_z = \frac{1}{s}G_z = \frac{1}{s}\left(\frac{s}{\alpha}\tilde{G}_{AVR_{PID}} + \tilde{G}_{AVR_{PID}}\right) = \frac{s}{\alpha}\left(\frac{1}{s}\tilde{G}_{AVR_{PID}}\right) + \left(\frac{1}{s}\tilde{G}_{AVR_{PID}}\right) \quad (8)$$

where $(1/s\tilde{G}_{AVR_{PID}})$ is the unit step response (Y) of the original approximated transfer function $(\tilde{G}_{AVR_{PID}})$, thus,

$$Y_z = \frac{s}{\alpha}Y + Y \quad \text{or} \quad y_z(t) = \frac{1}{\alpha}\dot{y}(t) + y(t) \quad (9)$$

This means that, the step response of the modified second order system with a zero at $s = -\alpha$ is given by the step response of the original system plus a scaled version of its derivative. As the zero moves further to the left side of the complex plane (α increases), the contribution of the derivative term $\dot{y}(t)$ decreases and the step response of the modified system starts to resemble the response of the original approximated system. Conversely, as the zero moves closer to the origin from the left side (α decreases), the contribution of the derivative term $\dot{y}(t)$ increases resulting in an increased overshoot, decreased rise and peak times (the step response becomes faster). Fig. 5, shows the effect of adding a zero to the approximated system defined by Equation (5) on the unit step response. The scaling factor $(1/\alpha)$ of the derivative term $\dot{y}(t)$ is varied from 0 to 0.3.

From Fig. 5, it can be observed that when the contribution of the derivative term increases ($1/\alpha$ increases) the response becomes faster (t_r decrease) and possess higher overshoot peak (M_p increase). These results have been also illustrated in Fig. 6 where the step response parameters (t_r, t_s, M_p, and t_p) are plotted against the same range of variation of the scaling parameter $(1/\alpha)$.

In Fig. 6, the rise time is recorded as the time in which the response takes to rise from 0 to 80% of the steady-state value. When $(1/\alpha)$ increases, t_r, t_s, and t_p decrease against an increase of M_p.

When adding a zero to a second order system with underdamped case ($\zeta < 1$), such as the approximated system defined by Equation (5), the modified system will possess a faster response versus an undesirable increase of M_p. Within the time interval $t_r \le t \le t_p$, the value of the original step response is $1 \le y(t) \le 1 + M_p$ and thus the value of its derivative is $\varepsilon \le (1/\alpha)\,\dot{y}(t) \le 0$, where ε is a positive real number. The value of ε depends proportionally on the scaling parameter $(1/\alpha)$. Therefore the value of the modified response is $1 + \varepsilon \le \{y_z(t) = y(t) + (1/\alpha)\,\dot{y}(t)\} \le 1 + M_p$, and thus M_{pz} will become greater than M_p, as well as $t_{pz} < t_p$, and $t_{rz} < t_r$ where M_{pz}, t_{pz}, and t_{rz} are the maxim overshoot, peak time, and rise time of the modified response $y_z(t)$.

In the critical damped case ($\zeta = 1$), where the poles are both located at $s = -\omega_n$, the unit step response is given by [27],

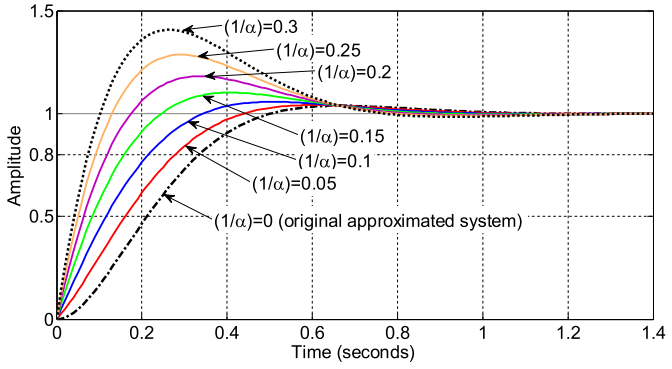

Fig. 5. Effect of adding a zero to $\tilde{G}_{AVR_{PID}}$ on the unit step response.

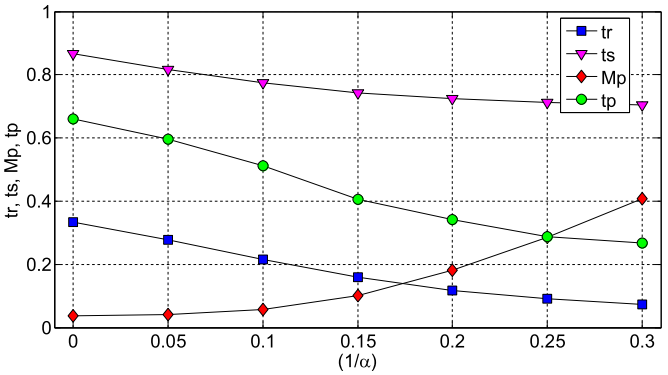

Fig. 6. Effect of adding a zero to $\tilde{G}_{AVR_{PID}}$ on t_r, t_s, M_p, and t_p.

$$y(t) = 1 - e^{-\omega_n t} - \omega_n t e^{-\omega_n t} \quad (10)$$

The peak time t_{pz} at which the maximum overshoot M_{pz} of the modified response $y_z(t)$ occurs, can be found by substituting Equation (10) in (9), taking the derivative of $y_z(t)$, and equating to zero yields,

$$\dot{y}_z(t) = \frac{1}{\alpha}\ddot{y}(t) + \dot{y}(t) = e^{-\omega_n t}\left[\omega_n^2 t\left(1 - \frac{\omega_n}{\alpha}\right) + \frac{\omega_n^2}{\alpha}\right] = 0 \quad (11)$$

Solving equation (11) for t to get,

$$t = t_{pz} = \infty \quad \text{or} \quad t = t_{pz} = \frac{1}{(\omega_n - \alpha)}, \quad \text{for } \alpha < \omega_n \quad (12)$$

From Equation (12) choosing a value of α less than ω_n ($\alpha < \omega_n$) will make the step response posses an overshoot given by,

$$M_{pz} = y_z(t_{pz}) = y_z\left(\frac{1}{\omega_n - \alpha}\right) \quad (13)$$

On the other hand, choosing $\alpha \ge \omega_n$ will positively eliminate the overshoot.

In the overdamped case ($\zeta > 1$), where the poles are both real located at $s_1 = -r_1$ and $s_2 = -r_2$, where $r_2 > r_1$, the unit step response is given by [27],

$$y(t) = 1 - \frac{r_2 e^{-r_1 t} - r_1 e^{-r_2 t}}{r_2 - r_1} \quad (14)$$

Similarly, the peak time t_{pz} at which the maximum overshoot M_{pz} of the modified response $y_z(t)$ occurs can be found as in the critical damped case to get,

$$\dot{y}_z(t) = \frac{1}{\alpha}\ddot{y}(t) + \dot{y}(t) = e^{-r_1 t}\left(1 - \frac{r_1}{\alpha}\right) - e^{-r_2 t}\left(1 - \frac{r_2}{\alpha}\right) = 0 \quad (15)$$

Solving Equation (15) for t to get,

$$t = t_{pz} = \frac{1}{(r_2 - r_1)}\ln\left(\frac{\alpha - r_2}{\alpha - r_1}\right) \quad (16)$$

From Equation (16) choosing a value of α such that $(\alpha < r_1 < r_2)$ will make the step response posses an overshoot. Otherwise, choosing $(r_1 < \alpha < r_2)$ or $(r_1 < r_2 < \alpha)$ will eliminate the overshoot.

From the previous analysis, in AVR system controller design, two objectives are considered. The first objective is to modify the PID

controller such that, when optimized, the overall transfer function of the AVR system can be reduced to have the form defined by Equation (7). The second objective is to direct the optimization algorithm used to tune the controller parameters to minimize M_{pz}, t_{pz}, and t_{rz} as well as the settling time t_{sz}. To achieve these objectives a four term control type structure is proposed consisting of proportional, integral, derivative, and second order derivative terms (PIDD^2) defined by,

$$C_{\mathrm{PIDD}^2} = K_p + \frac{K_i}{s} + K_d s + K_{d2} s^2. \tag{17}$$

The difference between the proposed PIDD^2 and the conventional PID controllers is the extra second order derivative term added in the PIDD^2 controller. This term is determined by the gain parameter K_{d2}. Substituting the transfer functions of the AVR system components listed in Table 1 with their parameters and the transfer function of the proposed PIDD^2 controller given by Equation (17) in (3) yields,

$$G_{\mathrm{AVR\_PIDD}^2} = \frac{0.1K_{d2}s^4 + (0.1K_d + 10K_{d2})s^3 + (10K_d + 0.1K_p)s^2 + (0.1K_i + 10K_p)s + 10K_i}{0.0004s^5 + 0.0454s^4 + (10K_{d2} + 0.555)s^3 + (10K_d + 1.51)s^2 + (10K_p + 1)s + 10K_i} \tag{18}$$

The proposed PIDD^2 controller is expected to compensate the effect of two AVR system poles, and hence reducing the overall transfer function to that defined by Equation (7). With the new proposed controller structure, the optimization algorithm employed for designing the PIDD^2 controller will attempt to tune four gain parameters.

5. Particle swarm optimization

The PSO algorithm is considered to be one of the most promising optimization techniques due to its simplicity, robustness, fast convergence, and ease of implementation [28]. Solving optimization problem with PSO is based on the concept of social interaction in which a population of individual solutions called particles is employed for the searching process [29]. The particles are grouped in a finite set called swarm and are updated iteratively. In each iteration, the particles exchange previously discovered information with neighbors and use these information to update their new position. The new positions of particles are calculated by adding their previous position to their corresponding updated velocity values. In PSO algorithm, updating the velocity for each particle is the most important step. The velocity is updated using the previous velocity (inertia), personal influence (cognitive), and social influence (social) components. The inertia component prompts the particle to move in the same previous direction and velocity. The cognitive component improves the new particle's position by comparing it with the best previous position found associated with this particle. The social component makes the particle follow the best neighbor's direction. The modified velocity and position of each particle are calculated according to the following equations [30]:

$$V_i^{k+1} = wV_i^{k+1} + c_1 r_{i1}^k \left(P_i^k - X_i^k\right) + c_2 r_{i2}^k \left(P_g^k - X_i^k\right) \tag{19}$$

$$X_i^{k+1} = X_i^k + V_i^{k+1} \tag{20}$$

where $i = 1, 2, \ldots, L$, and L is the number of population (swarm size); w is the inertia weight, c_1 and c_2 are two positive constants, called the cognitive and social parameters respectively; r_{i1} and r_{i2} are random numbers uniformly distributed within the range [0, 1]. Equation (19) above is used to find the new velocity for the ith particle, while Equation (20) is used to update the ith position by adding the new velocity obtained by Equation (19).

A simplified version of PSO (SSO) called "social only" suggested by Kennedy is implemented by eliminating the personal influence (cognitive) term in the velocity update equation [31]. This can be achieved by setting c_1 in Equation (19) to zero, thus it becomes:

$$V_i^{k+1} = wV_i^{k+1} + c_2 r_{i2}^k \left(P_g^k - X_i^k\right) \tag{21}$$

The simplified PSO is also called Many Optimizing Liaisons (MOL) to make it easy to distinguish from the original PSO [9]. MOL differs from PSO in that it eliminates the particle's best-known position thus making the algorithm simpler.

6. PIDD^2 implementation issues

Presently, almost all control strategies are implemented as digital algorithms in microprocessor-based equipment such as programmable logic controllers (PLCs) and digital signal processors (DSPs). To become applicable in such equipment, the PID control algorithm has to be discretized using discretization methods. These methods can be applied similarly to discretize the proposed PIDD^2 controller. The continuous time expression of the proposed PIDD^2 controller in ideal form is given by:

$$u(t) = K_p e(t) + K_i \int_0^t e(t)dt + K_d \frac{de(t)}{dt} + K_{d2} \frac{d^2 e(t)}{dt^2}. \tag{22}$$

Applying the trapezoidal approximation to discretize the integral term and the backward finite differences approximation to discretize the first and second derivative terms [32] in Equation (22) to get an approximated discrete transfer function of the PIDD^2 given by,

$$\frac{U(z)}{E(z)} = C_{\mathrm{PIDD}^2}(z) = K_p + \frac{K_i T_s}{2}\left[\frac{z+1}{z-1}\right] + K_d\left[\frac{z-1}{T_s z}\right] + K_{d2}\left[\frac{z-1}{T_s z}\right]^2 \tag{23}$$

where T_s is the sampling interval. The common practical implementation problems of the PID controller are the integral windup and derivative kick problems. Remedies for the integral windup problem used with PID implementation can also be applied for the PIDD^2 controller. However, due to the second derivative term of the proposed PIDD^2 controller, the derivative kick problem becomes a major concern in practical implementation. A drawback with the first order derivative term is that it will amplify the input signal with a gain directly related to its frequency (linear increasing magnitude Bode plot with 20 dB per decade). The effect of this drawback will be doubled with the second order derivative term and the gain become directly related to the square of its frequency

(linear increasing magnitude Bode plot with 40 dB per decade). The amplification effect is more evident when the error signal exhibit high frequency components caused by measurement noise, load disturbance, and/or set point changes. For example, when an abrupt (stepwise) change of the set-point value occurs, the first and second derivative actions will be very large and this results in an undesirable spike (first plus second derivative kick) in the control variable signal. As a result, the actuator unit will experience a rapidly changing command signal that could be detrimental to the operation of the unit. This problem can be solved by limiting the bandwidth of the first and second order derivative actions with a first and second order low-pass filters respectively. In this context, the PIDD^2 controller defined by Equation (17) can be modified to be

$$\tilde{C}_{\mathrm{PIDD}^2} = K_p + \frac{K_i}{s} + K_d\left[\frac{s}{1+\frac{sT_d}{N_1}}\right] + K_{d2}\left[\frac{s}{1+\frac{sT_{d2}}{N_2}}\right]^2, \tag{24}$$

where T_d and T_{d2} are the first and second derivative time constants respectively. The filters coefficients N_1 and N_2 can be adjusted to set the cutoff frequencies of the first and second order derivative filters respectively. When N_1 and N_2 approach infinite, Equation (24) reduces to the ideal form C_{PIDD^2}. The high-frequency gains of the modified first and second derivative terms are

```
// initialize 1st and 2nd derivatives limits
max_derv1 = 0
min_derv1 = 0
max_derv2 = 0
min_derv2 = 0

//-------------- inside sample loop -------------------------------
// calculate current 1st derivative using Backward difference
derv1 = (error - previous_error)/sampling_period;
if (derv1 > max_derv1)
        new_derv1 = max_derv1;                  // Upper limit
else
if (derv1 < min_derv1)
        new_derv1 = min_derv1;                  // Lower limit
else
        new_derv1 = derv1;                      // Within range

// new_derv1 is used for the 1st derivative PIDD2 action (D)

        previous_new_derv1 = new_derv1;         // for next sample
        previous_error = error                  // for next sample

// set new 1st derivative limits
if (derv1 > previous_derv1)                     // for next sample
        {max_derv1 = derv1;
         min_derv1 = previous_derv1;}
else
        {max_derv1 = previous_derv1;
         min_derv1 = derv1;}

        previous_derv1 = derv1;                 // for next sample

// calculate current 2nd derivative using Backward difference
derv2 = (new_derv1 - previous_ new_derv1)/sampling_period;
if (derv2 > max_derv2)
        new_derv2 = max_derv2;                  // Upper limit
else
if (derv2 < min_derv2)
        new_derv2 = min_derv2;                  // Lower limit
else
        new_derv2 = derv2;                      // Within range

// new_derv2 is used for the 2nd derivative PIDD2 action (D2)

// set new 2nd derivative limits
if (derv2 > previous_derv2)                     // for next sample
        {max_derv2 = derv2;
         min_derv2 = previous_derv2;}
else
        {max_derv2 = previous_derv2;
         min_derv2 = derv2;}

        previous_derv2 = derv2;                 // for next sample
//-------------------------------------------------------------------
```

Fig. 7. Pseudocode for NMF used to realize first and second order derivative actions.

$$\lim_{s\to\infty} K_d\left[\frac{s}{1+\frac{sT_d}{N_1}}\right] = K_d(N_1/T_d) \quad \text{and} \quad \lim_{s\to\infty} K_{d2}\left[\frac{s}{1+\frac{sT_{d2}}{N_2}}\right]^2 = K_{d2}(N_2/T_{d2})^2 \tag{25}$$

With the modified PIDD^2 controller defined by Equation (24), the optimization algorithm can also be modified to tune the filters coefficients N_1 and N_2 along with the four gain parameters. In this case, the optimization objective is to minimize t_r, t_s, M_p, and to minimize the maximum range of the controller output.

An alternative method for smoothing the first and second derivative actions is to use a nonlinear median filter (NMF) [33], which is widely applied in image processing. The NMF compares several data points around the current point and selects their median for the control action. As a result, the high frequency components (unwanted spikes) resulting from a step command, noise, or disturbance are removed completely. Fig. 7 illustrates the pseudocode of the NMF for the first and second derivative actions.

Unlike lowpass filters, which averages past values, NMF is capable of removing extraordinary derivative values resulting from sudden changes in the error signal. Fig. 8 shows an example of an error signal, $e(t)$, having high frequency components (abrupt changes and sharp edges). The first and second derivatives of $e(t)$ are computed using NMF.

The error signal example, shown in Fig. 8, has abrupt changes at time instants 0.5, 3.5, 4, and 4.9 s. When a backward difference method is used to approximate the first and second order derivatives, unwanted spikes will occur at these instants. However, with NMF the undesired spikes are completely removed and thus resulting in a nonaggressive control signal.

7. Simulation results and discussion

In this section, the proposed PIDD^2 controller is tested in controlling the AVR system G_{AVR} defined by Equation (1). The performance of the PIDD^2 controller is compared with conventional PID controllers tuned by recently published modern heuristic optimization techniques. The PIDD^2 is also compared with FOPID controller. In addition, transient response, zero-pole, frequency response, and robustness analysis are performed on the proposed PIDD^2 controller. The realization of the proposed PIDD^2 controller and its discrete implementation is also tested in SIMULINK®. The PSO algorithm is employed to tune the PIDD^2 gain parameters using the integral of time multiplied by absolute error (ITAE) performance criterion [34]. The simulation parameters of the PSO algorithm are listed in Table 2.

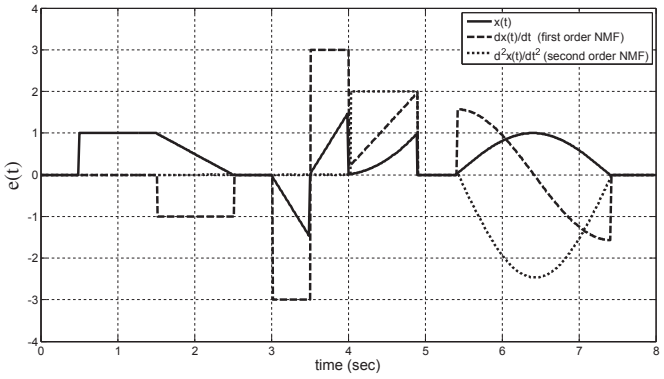

Fig. 8. An example illustrating computation of first and second order derivatives using NMF.

The searching range of the PIDD^2 gains and their corresponding velocity constraints are defined in Table 3.

To improve the search process of any optimization algorithm, it is necessary to bound the dimensions of the searching space. In PID controller tuning, defining the maximum limits of the gains is important for control system stability. From recent literature results, it has been found that optimum PID gain values used to control the AVR system G_{AVR} are within [0, 1.5], [0, 1], [0, 1] for K_p, K_i, and K_d respectively [3,9,10]. However, for the proposed PIDD^2 the search ranges of all gains are expanded to be [0.0001, 3]. The maximum and minimum velocity limits determine the resolution, or fitness, with which regions be searched between the present PIDD^2 gain value and the target value. If these limits are chosen high, the PIDD^2 gain values may move erratically, going beyond a good solution. On the contrary, if the limits are chosen too small the gains may not explore sufficiently beyond local solutions. An effective velocity limit value is chosen to be 20% of the corresponding maximum gain value [35].

For each particle (set of PIDD^2 gains), the closed-loop system stability is tested using the "isstable" Matlab function. If the function returns a logical true value, then the solution is feasible and its fitness value is considered. Otherwise, if the function returns a logical false, then the closed-loop system is unstable and hence the solution is infeasible. Infeasible solutions are excluded by penalizing them with very large fitness value.

7.1. Transient response analysis

The transient response of the proposed PIDD^2 controller tuned with PSO is analyzed by comparing the unit-step response with different PID controllers. The PID controllers were designed in recent literature using PSO [3], MOL [9], GA [3], ABC [7], DEA [7], and LUS [10] for the same AVR system. Fig. 9, shows a comparison of the AVR terminal voltage step response of the proposed PIDD^2 and different PID controllers. Each PID controller is associated with one of the aforementioned tuning algorithms and one objective function. The different objective functions used are the ITAE, integral of time multiplied by squared error (ITSE) [7], f function [3], and OF4 function [10], defined by,

Table 2
PSO searching parameters.

Parameter	Value
Number of iterations (N)	50
Number of trials (T)	10
Swarm size (L)	30
Acceleration constants ($c_1 = c_2$)	2
Inertia weight factor (w)	[0.9:0.014:0.2]

Table 3
Searching range of parameters.

Parameter	Min. value	Max. value
K_p	0.0001	3
K_i	0.0001	3
K_d	0.0001	3
K_{d2}	0.0001	3
v_{K_p}	−0.6	0.6
v_{K_i}	−0.6	0.6
v_{K_d}	−0.6	0.6
$v_{K_{d2}}$	−0.6	0.6

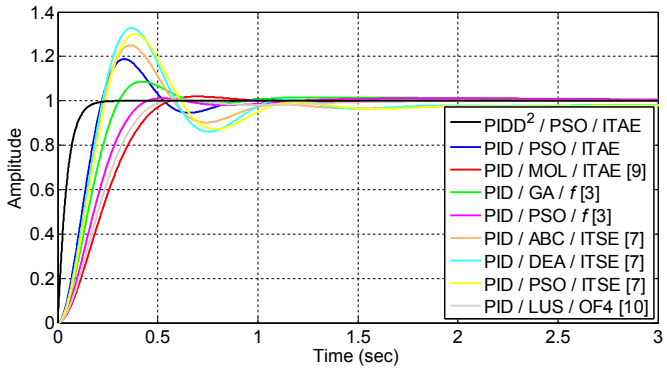

Fig. 9. Terminal voltage step response of the AVR system with different controllers.

$$\mathrm{ITAE}_{\min} = \int_0^{t_{ss}} t|e(t)|dt, \tag{26}$$

$$\mathrm{ITSE}_{\min} = \int_0^{t_{ss}} te^2(t)dt, \tag{27}$$

$$f_{\max} = \frac{1}{(1-e^{-\beta})(M_p + E_{ss}) + e^{-\beta}(t_s - t_r)}, \tag{28}$$

$$\mathrm{OF4}_{\min} = 0.8* \int_0^{t_{ss}} e^2(t)dt + 0.1*t_s + 0.1*M_p, \tag{29}$$

respectively. In Equations (26)–(29), t_{ss} is the time at which the response reaches steady state, β is a weighting factor, and E_{ss} is the steady state error.

From Fig. 9, it can be observed that the proposed PIDD^2 possess a superior step response behavior compared to other PID controllers. Table 4, lists the numerical results of the response comparison including; controller parameters, the time domain performance indices (M_p, t_r, t_s, and t_p), and the objective function values.

It is clear from Table 4, that the best response performance indices values, highlighted in bold, are those obtained with the proposed PIDD^2 controller ($M_p = 0$, $t_r = 0.0929$, $t_s = 0.1635$, and $t_p = 0.32$). Therefore, in comparison to all PID controllers, the PIDD^2 has the ability to achieve the fastest (minimum t_r and t_s), most accurate (minimum response oscillation), and most stable (minimum overshoot) unit step response.

The proposed PIDD^2 controller designed by PSO is compared with PID and FOPID controllers designed using CAS algorithm [12,13,24]. The CAS algorithm is implemented using the parameters listed in Table 5.

Table 5
CAS algorithm parameters [24].

Parameter	Value
Number of ants (K)	20
Positive constants (a, b)	(300, 2/3)
Organization factor of ant i (r_i)	0.04 + 0.1 × rand( )
Initial state of ant i ($y_i(0)$)	0.999
$\psi_d (d = 1, 2, \ldots, 5)$	$7.5/\omega_d$
Number of iterations	300

rand( ) is a uniformly distributed number in [0, 1].
ω_d is the interval of search of the d-th controller parameter.

Also, the PSO-PIDD^2 is compared with PSO-PID [3] and PSO-FOPID controllers [24]. The reciprocal of f defined in Equation (28) is considered as the objective function to tune the PSO-PID, PSO-FOPID, CAS-PID, and CAS-FOPID with two cases; $\beta = 1$ and $\beta = 1.5$. The terminal voltage step responses of the AVR system controlled by PSO-PIDD^2, PSO-PID, PSO-FOPID, CAS-PID and CAS-FOPID controllers are shown in Fig. 10 with $\beta = 1$ and $\beta = 1.5$.

As can be seen from Fig. 10, the response of the PIDD^2 is much better than the PID and FOPID controllers tuned with PSO and CAS algorithms in both cases (i.e. $\beta = 1$ and $\beta = 1.5$). This can be clearly observed from the time performance indices of all controllers listed in Table 6.

It is observed from Table 6 that the PSO-PIDD^2 has the best performance compared to PSO/CAS-PID and PSO/CAS-FOPID controllers. The terminal voltage step response of the AVR system controlled by the proposed PIDD^2 controller has the smallest values of M_p, E_{ss}, t_r, and t_s highlighted in bold.

The transfer function of the FOPID controllers defined by the parameters listed in Table 6, are then implemented with integer orders transfer function using Oustaloup recursive distribution of poles and zeroes approximation [36]. The integer orders transfer function obtained by Oustaloup approximation will have an order equal to 12 [24]. This fact adds another preference to the PIDD^2 related to implementation complexity. It is worth noting that, the proposed PIDD^2 controller can be extended to fractional order PIDD^2 (PI$^{\lambda}$D$^{\mu}$D$^{\mu_2}$) where λ, μ, and μ_2 are non-integer (fractional) orders of the integral, first and second order derivatives parts respectively. The complexity of this controller is evident due to the increase in the number of control parameters. There are seven different parameters (K_p, K_i, K_d, K_{d2}, λ, μ, and μ_2) that have to be tuned. The challenge of this work is to develop a realizable FOPIDD^2 controller that exhibits a robust performance with fewer parameters, yet achieving the same design requirements. The key point is to look for acceptable and realizable approximations of s^{λ}, s^{μ}, and $s^{\mu 2}$ which is recommended for future investigation.

Table 4
Controller parameters and response performance indices of different controllers.

Controller/algorithm/OF	Controller parameters				M_p%	t_r 0.1 → 0.9	t_s ±2%	t_p	Obj. value
	K_p	K_i	K_d	K_{d2}					
PIDD^2/PSO/ITAE	**2.7784**	**1.8521**	**0.9997**	**0.07394**	**0**	**0.0929**	**0.1635**	**0.3200**	**0.0018**
PID/PSO/ITAE	1.3541	0.9266	0.4378	-	18.805	0.1493	0.8146	0.3276	0.0329
PID/MOL/ITAE [9]	0.5857	0.4189	0.1772	–	1.9539	0.3433	0.5155	0.7036	0.0464
PID/GA/f [3]	0.8861	0.7984	0.3158	–	8.6532	0.2041	0.6058	0.4222	1.1982
PID/PSO/f [3]	0.6568	0.5393	0.2458	–	1.1652	0.2722	0.4111	1.9200	1.4480
PID/ABC/ITSE [7]	1.6524	0.4083	0.3654	–	25.035	0.1559	3.0939	0.3629	0.0177
PID/DEA/ITSE [7]	1.9499	0.4430	0.3427	–	32.830	0.1513	2.6494	0.3636	0.0220
PID/PSO/ITSE [7]	1.7774	0.3827	0.3184	–	30.048	0.1609	3.3994	0.3909	0.0238
PID/LUS/OF4 [10]	0.6190	0.4222	0.2058	–	0.5900	0.3123	0.4778	0.6008	0.1677

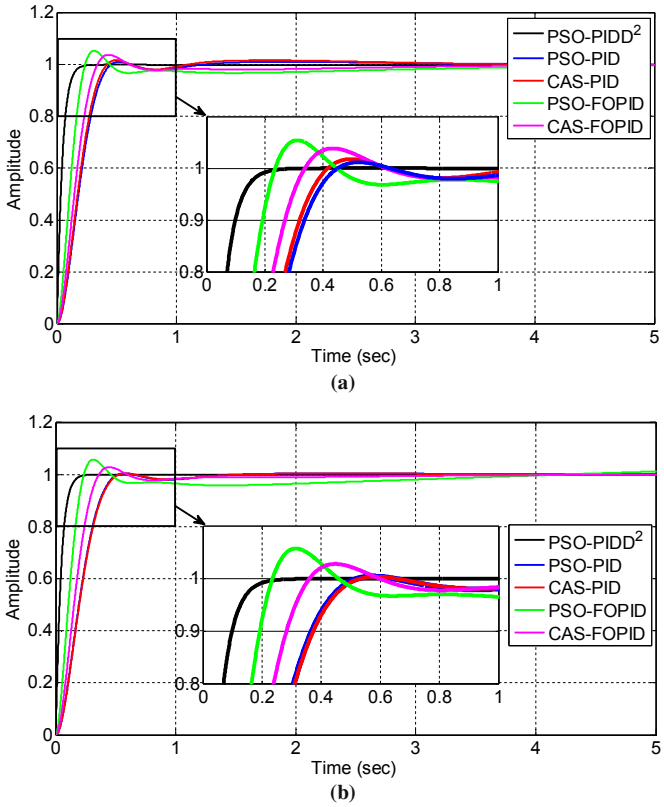

Fig. 10. Step response of AVR system controlled by PSO-PIDD^2, PSO-PID, CAS-PID, PSO-FOPID, and CAS-FOPID (a) $\beta = 1$ (b) $\beta = 1.5$.

7.2. Zero-pole analysis

The overall closed-loop transfer function of the AVR system with the proposed PIDD^2 controller is of 5th order given by

$$G_{\text{AVR\_PIDD}^2} = \frac{18.4855(s+100)(s+10.02)(s+2.501)(s+0.9994)}{(\text{s}+75.53)\ (\text{s}+24.43)\ (\text{s}+10.04)\ (\text{s}+2.502)\ (\text{s}+0.9994)} \tag{30}$$

The zero-pole map of the AVR system with the proposed PIDD^2 controller is shown in Fig. 11.

It can be observed that the system possess three zero-pole cancellation pairs located at -1, -2.5, and -10, two real dominant poles at $s_1 = -24.43$ and $s_1 = -75.53$, and one real zero at $z_1 = -\alpha = -100$. Due to the three zero-pole cancellation, the overall transfer function in Equation (30) can be approximated to be,

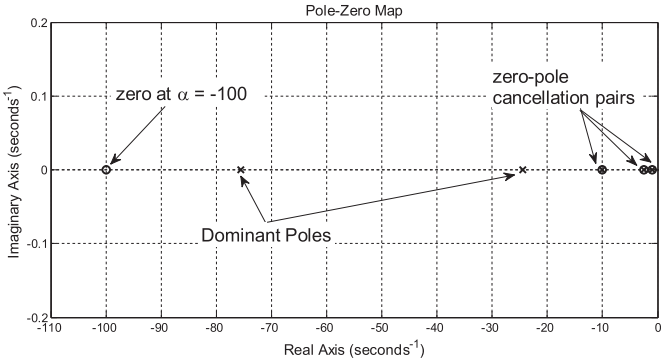

Fig. 11. Zero-pole map of the AVR system controlled by PIDD^2.

$$\tilde{G}_{\text{AVR\_PIDD}^2} = \frac{1845.2\left(\frac{s}{100}+1\right)}{(\text{s}+75.53)\ (\text{s}+24.43)} \tag{31}$$

Comparing Equation (31) with the overdamped case of Equation (7) yields, $\alpha = 100$, $r_1 = 24.43$, and $r_2 = 75.53$ with $r_1 < r_2 < \alpha$. In this case, the system response possess no overshoot and this can be ensured by substituting the values of α, r_1, and r_2 in Equation (16). The ration of $(\alpha - r_2)/(\alpha - r_1)$ inside the logarithm function is less than one, thus resulting in a negative time value which indicates no overshoot exists in the system's response.

7.3. Frequency response analysis

The frequency response of the AVR system with the proposed PIDD^2 controller is analyzed. The magnitude and phase plots of the AVR with PIDD^2 controller is shown in Fig. 12. The peak gain, phase margin, delay margin and bandwidth obtained from the system's frequency response are depicted in Table 7 and compared with different controllers.

As shown in Table 7, the PIDD^2 is the most stable system compared to other controllers. The AVR with PIDD^2 controller have minimum peak gain 0 dB at 0 Hz, maximum phase margin 180°, infinite delay margin (smallest time delay required to make the system unstable), and maximum bandwidth (fastest response). It is worth noting that, a wide bandwidth allows the system to follow arbitrary inputs accurately.

Table 6
Controller parameters and performance indices of PSO-PIDD^2, PSO-PID, CAS-PID, PSO-FOPID, and CAS-FOPID.

Algorithm-controller	Controller parameters						M_p%	E_{ss}	t_r 0.1 → 0.9	t_s ±2%
	K_p	K_i	K_d	K_{d2}	μ	λ				
PSO-PIDD^2	**2.778**	**1.852**	**0.999**	**0.074**	**-**	**-**	**0**	**1.06–08**	**0.0929**	**0.1635**
PSO-PID ($\beta = 1$)	0.6570	0.5389	0.2458	–	–	–	1.1601	1.63e-05	0.2721	0.4110
CAS-PID ($\beta = 1$)	0.6746	0.6009	0.2618	–	–	–	1.7686	1.04e-05	0.2574	0.3856
PSO-FOPID ($\beta = 1$)	1.6264	0.2956	0.3226	–	1.1980	1.3183	5.4124	0.009037	0.1567	2.6848
CAS-FOPID ($\beta = 1$)	1.0537	0.4418	0.2510	–	1.1122	1.0624	3.8524	0.001733	0.2191	0.5372
PSO-PID ($\beta = 1.5$)	0.6254	0.4577	0.2187	–	–	–	0.4394	4.68e-06	0.3003	0.4606
CAS-PID ($\beta = 1.5$)	0.6202	0.4531	0.2152	–	–	–	0.4026	5.30e-06	0.3045	0.4676
PSO-FOPID ($\beta = 1.5$)	1.6986	0.1797	0.3122	–	1.2081	1.8373	5.7732	0.043639	0.1579	33.518
CAS-FOPID ($\beta = 1.5$)	0.9315	0.4776	0.2536	–	1.0838	1.0275	2.8362	7.18e-04	0.2297	0.8949

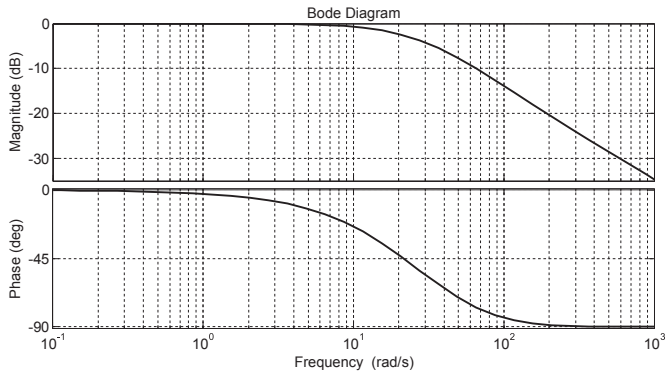

Fig. 12. Bode diagrams of the AVR controlled by PIDD^2.

Table 7
Bode analysis of different AVR controllers.

Controller/algorithm/OF	Peak gain dB	Phase margin (deg.)	Delay margin	Bandwidth
PIDD^2/PSO/ITAE	0 (0 Hz)	180	Inf.	23.5031
PID/PSO/ITAE	1.79 (1.32 Hz)	79.3	0.1207	13.9142
PID/MOL/ITAE [9]	0 (0 Hz)	180	Inf.	6.3391
PID/GA/f [3]	0.17 (0.17 Hz)	116.2	0.2926	10.6614
PID/PSO/f [3]	0.07 (0.11 Hz)	166.9	2.6147	8.3137
PID/ABC/ITSE [7]	2.87 (1.21 Hz)	69.4	0.1109	12.8798
PID/DEA/ITSE [7]	4.20 (1.23 Hz)	58.4	0.0916	12.8006
PID/PSO/ITSE [7]	3.76 (1.16 Hz)	62.2	0.1033	12.1825
PID/LUS/OF4 [10]	0 (0 Hz)	180	Inf.	7.1673
CAS-FOPID ($\beta = 1$) [24]	0.0053 (0.01 Hz)	178.5	23.324	9.9543
CAS-FOPID ($\beta = 1.5$) [24]	0.0003 (0.01 Hz)	179.2	39.151	9.6367

7.4. Robustness analysis

Robustness analysis is used to evaluate the controller ability to tolerate uncertainties exists in some system parameters. In this subsection, the PIDD^2 controller is tested against uncertainties of AVR system parameters. The uncertainties of the AVR model are specified in terms of variations in the amplifier, exciter generator, and sensor time constants (T_a, T_e, T_g, and T_s respectively) above and below their nominal values. The variation range of the time constants is chosen to be ±50% of their nominal values with a 25% step size. Figs. 13–16 show step responses of the PIDD^2 controlled AVR system with T_a, T_e, T_g, and T_s time constants variations about nominal responses respectively.

It can be realized from Figs. 13–16, that the deviations of response curves (±50% and ±25%) from the nominal response for the selected time constant parameters are within a small range. This can ensures the ability of the PIDD^2 to maintain stability and to perform properly despite such large variations. Tables 8 and 9 present a summary of the PIDD^2 robustness analysis results and list the total deviation ranges and maximum deviation percentage of the system respectively.

From Table 9, the average deviation of maximum overshoot, settling time, rise time and peak time are 3%, 138%, 41% and 188% respectively. The ranges of total deviation are acceptable and are within limit. Therefore, it can be concluded that the AVR system with the proposed PIDD^2 controller is robust and can still perform acceptable control behavior.

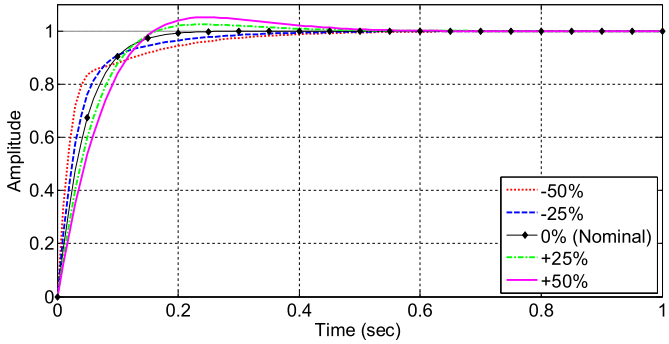

Fig. 13. Step response curves ranging from −50% to +50% for T_a.

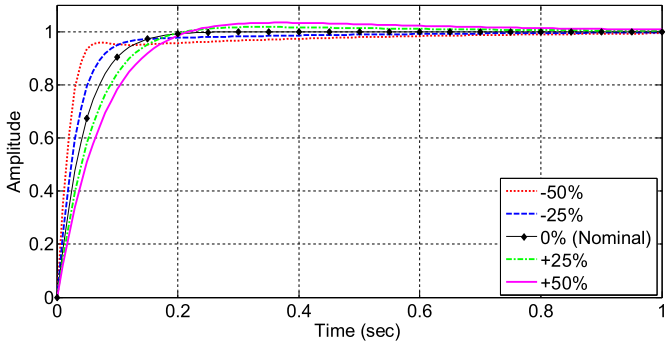

Fig. 14. Step response curves ranging from −50% to +50% for T_e.

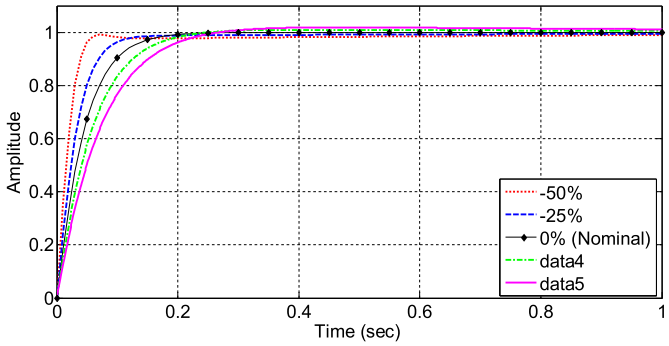

Fig. 15. Step response curves ranging from −50% to +50% for T_g.

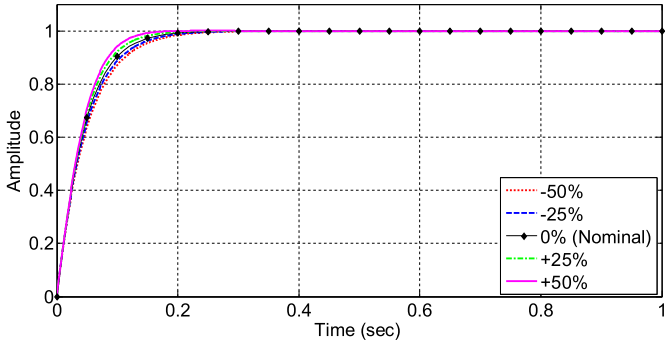

Fig. 16. Step response curves ranging from −50% to +50% for T_s.

7.5. Digital implementation

The realization of the proposed PIDD^2 controller and its discrete implementation is tested in SIMULINK® and compared with PID/MOL [9] and PID/GA [3] discrete controllers. The general Simulink model of the AVR control system is shown in Fig. 17.

Table 8
Robustness analysis results of the AVR system with the proposed PIDD^2 controller.

Parameter	Rate of change (%)	Peak value (pu)	t_s	t_r	t_p
T_a	−50%	0.9994	0.3352	0.1226	0.7836
	−25%	0.9967	0.2674	0.0897	0.4794
	+25%	1.0243	0.2964	0.1005	0.2394
	+50%	1.0514	0.4038	0.1084	0.2476
T_e	−50%	0.9985	0.5242	0.0395	1.6345
	−25%	0.9972	0.2615	0.0685	1.0805
	+25%	1.0173	0.1770	0.1139	0.3379
	+50%	1.0336	0.6386	0.1325	0.3698
T_g	−50%	0.9910	0.3119	0.0362	0.0717
	−25%	0.9940	0.1261	0.0648	0.8705
	+25%	1.0096	0.1952	0.1189	0.4176
	+50%	1.0186	0.2243	0.1430	0.4687
T_s	−50%	0.9997	0.1896	0.1067	0.3801
	−25%	0.9997	0.1774	0.1000	0.3532
	+25%	0.9996	0.1471	0.0857	0.2703
	+50%	1.0003	0.1285	0.0790	0.2441

Table 9
Total deviation ranges and maximum deviation percentage of the system.

Parameter	Total deviation range/max deviation percentage (%)			
	Peak value (pu) **0.9997**	t_s **0.1635**	t_r **0.0929**	t_p **0.3200**
T_a	0.0547/5%	0.1364/147%	0.0329/32%	0.5442/145%
T_e	0.0364/3%	0.4616/291%	0.0930/57%	1.2966/411%
T_g	0.0276/2%	0.1858/91%	0.1068/61%	0.7935/172%
T_s	0.0007/0%	0.0611/21%	0.0277/15%	0.1369/24%
Average	**0.0299/3%**	**0.2112/138%**	**0.0651/41%**	**0.6928/188%**

The controller subsystem, shown in Fig. 17, is implemented by a discrete PIDD^2 or PID controller having specifications defined in Table 10.

The sampling time (T_s) is chosen according to the rule of thumb suggested by Astrom and Wittenmark such that the product of T_s and the gain crossover frequency (ω_c in radians per second) of the loop gain ($C_{\text{PIDD}^2} G_a G_e G_g H_s$), is between 0.15 and 0.5 [32]. The gain crossover frequency of the AVR control system loop gain is $\omega_c = 18.2$ radian per second. Thus, an appropriate sampling time is between 0.008 and 0.0275 (T_s is set to 0.01). The response of the PIDD^2, PID/MOL [9], and PID/GA [3] controllers are tested at steady state by subjecting a disturbance load signals of values equal to +10% and −10% of the set point at times 3 and 5 s respectively. Figs. 18 and 19 show the set point responses due to the unit step input at $t = 0$ and responses due to load disturbances at $t = 3$ and 5 s along with the controller outputs.

Compared to PID/MOL [9] and PID/GA [3] controllers, the proposed PIDD^2 with NMF (PIDD^2/NMF) posses an improved set point and load disturbance responses as shown in Fig. 18. The responses of PIDD^2 with filtered first and second derivative actions (PIDD^2/N_1N_2) are faster than those of PID/MOL [9] and PID/GA [3] controllers, however, it has the highest maximum overshoots values.

In AVR control system, the controller actions are carried out as a response to load disturbance (regulating system) not to set point changes (tracking system). Therefore, in Fig. 19, only the responses to load disturbances are shown. It can be observed that the range of the controller output signals for the PIDD^2/NMF, PID/MOL [9], and PID/GA [3] controllers are ±3.8, ±0.7, and ±0.44 respectively. However, for PIDD^2/N_1N_2 controller, the range of the controller output signal exceeds ±4.

Controllers are designed to work with nonlinear behavior of process actuators. The actuator device, such as the amplifier in the AVR system, has a limited range of input and output operation. Such limitations appear at the input of the actuator and are modeled with a non-linear element having saturation characteristics. Moreover, when abrupt change occurs in the system output due to a load disturbance, the controller output will exhibit a large spike values similar to those of the PIDD^2/NMF shown in Fig. 19. These spikes are mainly due to the first and second derivative actions and could be detrimental to the operation of the actuator unit. To avoid subjecting the actuator unit to such large controller output values, a constrained action defined by the maximum and minimum output range limits of the actuator. However, in this case the integral action will produce an inaccurate and highly excessive value causing oscillation and slowing down the transient response. This behavior is called the integrator windup problem. This can be solved by several anti-windup algorithms such as the configuration suggested by Wilkie et al. [37].

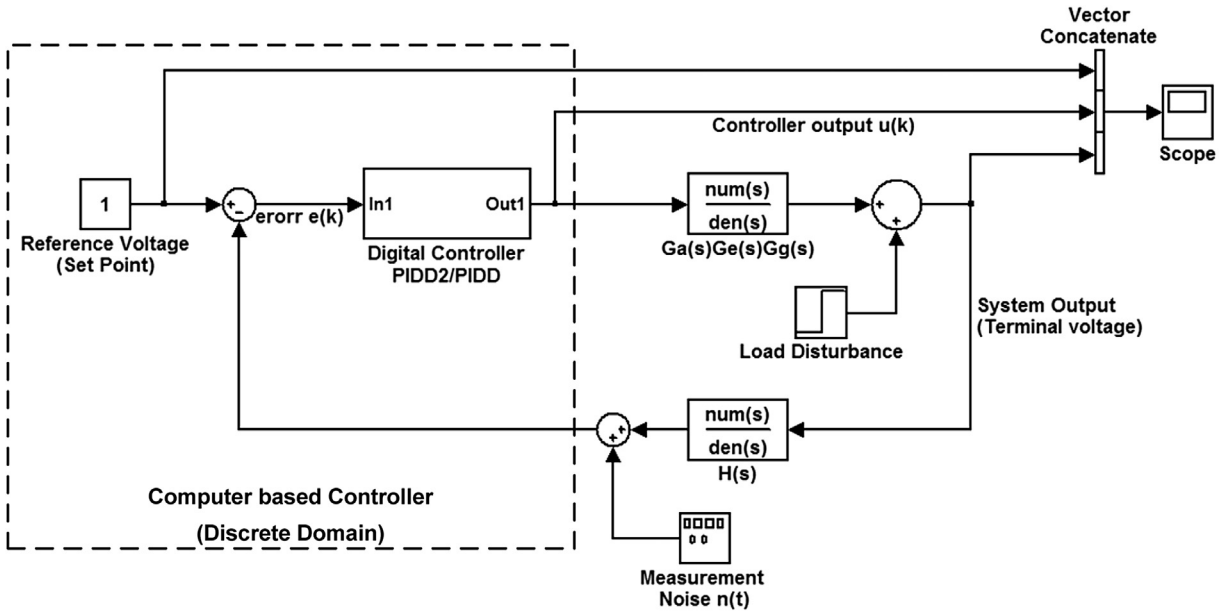

Fig. 17. Simulink model of the AVR control system.

Table 10
Controller subsystem specifications.

Controller	K_p	K_i Integration method	K_d Filter method	K_{d2} Filter method	Filter coefficient(s)	Controller formula $C(z)$
PIDD^2	2.7784	1.8521 Trapezoidal	0.9997 NMF^1	0.07394 NMF^2	–	$K_p + \frac{K_i T_s}{2}\left[\frac{z+1}{z-1}\right] + K_d NMF^1 + K_{d2} NMF^2$
PIDD^2	2.7784	1.8521 Trapezoidal	0.9997 Backward Euler	0.07394 Backward Euler	$N_1 = 30$ $N_2 = 80$	$K_p + \frac{K_i T_s}{2}\left[\frac{z+1}{z-1}\right] + K_d\left[\frac{N_1}{1+\frac{N_1 T_s z}{z-1}}\right] + K_{d2}\left[\frac{N_2}{1+\frac{N_2 T_s z}{z-1}}\right]^2$
PID/MOL [9]	0.5857	0.4189 Trapezoidal	0.1772 Backward Euler	–	$N = 30$	$K_p + \frac{K_i T_s}{2}\left[\frac{z+1}{z-1}\right] + K_d\left[\frac{N}{1+\frac{N T_s z}{z-1}}\right]$
PID/GA [3]	0.8861	0.7984 Trapezoidal	0.3158 Backward Euler	–	$N = 30$	

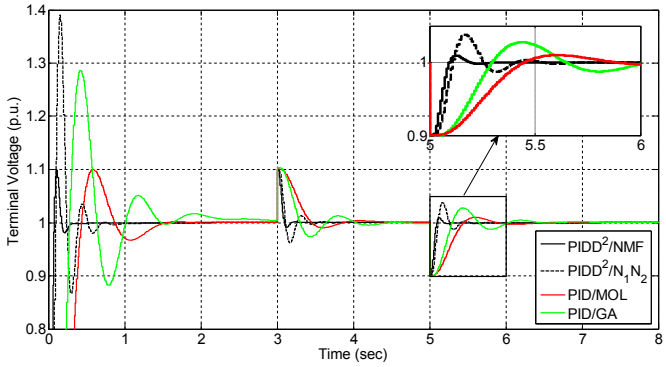

Fig. 18. Set-point and disturbance responses the AVR control system.

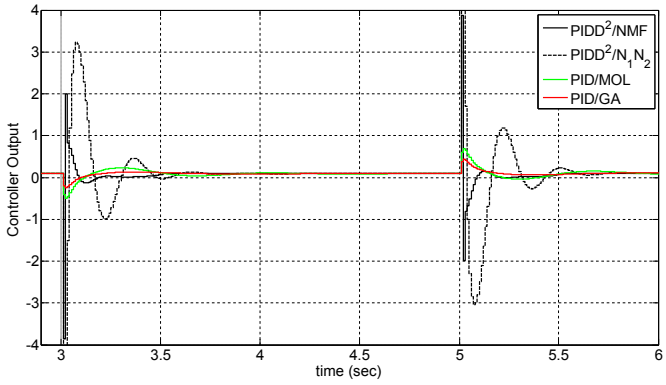

Fig. 19. Controller output.

The results as summarized in Table 11 indicate that the response of the proposed PIDD^2/NMF controller outperforms the responses of the PIDD^2/N$_1$N$_2$, PID/MOL [9], and PID/GA [3] in terms of maximum overshoot, rise time, and settling time. The best response performance indices values of the proposed PIDD^2/NMF controller are highlighted in bold.

Table 11
Performance comparison.

Controller	Set-point response			Load-disturbance response		
	M_p%	t_r	t_s	M_p%	t_r	t_s
PIDD^2/NMF	**10**	**0.08**	**0.16**	**9.3**	**0.09**	**0.18**
PIDD^2/N$_1$N$_2$	39	0.10	0.49	38	0.11	0.50
PID/MOL [9]	10	0.43	1.27	10	0.45	0.28
PID/GA [3]	29	0.28	1.36	28	0.30	1.36

8. Conclusion

In this paper, a novel PID plus second order derivative controller (PIDD^2) is proposed to control AVR system. The proposed PIDD^2 consists of four control terms; proportional, integral, derivative, and second derivative. The PSO algorithm with the integral of time multiplied by absolute error (ITAE) performance criterion is used to tune the four gains of the PIDD^2 controller. The performance of the AVR with PIDD^2 is compared with several PID controllers tuned by recently proposed approaches, such as MOL, GA, ABC, DEA, and LUS. In addition, the proposed PIDD^2 is compared with the FOPID controller designed by using CAS algorithm. Simulation results show a superior response performance of the proposed PIDD^2. Moreover, the frequency response, zero-pole, and robustness analysis performed on the PIDD^2 controller showed more robust stability and better performance characteristics than the PID and FOPID controllers.

References

[1] P. Kundur, N.J. Balu, M.G. Lauby, Power System Stability and Control, vol. 7, McGraw-Hill, New York, 1994.
[2] A. Kiam Heong, G. Chong, L. Yun, PID control system analysis, design, and technology, IEEE Trans. Control Syst. Technol. 13 (4) (2005) 559–576.
[3] G. Zwe-Lee, A particle swarm optimization approach for optimum design of PID controller in AVR system, IEEE Trans. Energy Convers. 19 (2) (2004) 384–391.
[4] P. Wang, D.P. Kwok, Optimal design of PID process controllers based on genetic algorithms, Control Eng. Pract. 2 (4) (1994) 641–648.
[5] V. Mukherjee, S.P. Ghoshal, Intelligent particle swarm optimized fuzzy PID controller for AVR system, Electr. Power Syst. Res. 77 (12) (2007) 1689–1698.
[6] M. Kashki, Y. Abdel-Magid, M. Abido, A reinforcement learning automata optimization approach for optimum tuning of PID controller in AVR system, in: D.-S. Huang, et al. (Eds.), Advanced Intelligent Computing Theories and Applications. With Aspects of Artificial Intelligence, Springer Berlin, Heidelberg, 2008, pp. 684–692.
[7] H. Gozde, M.C. Taplamacioglu, Comparative performance analysis of artificial bee colony algorithm for automatic voltage regulator (AVR) system, J. Franklin Inst. 348 (8) (2011) 1927–1946.
[8] G. Reynoso-Meza, et al., Controller tuning using evolutionary multi-objective optimisation: current trends and applications, Control Eng. Pract. 28 (0) (2014) 58–73.
[9] S. Panda, B.K. Sahu, P.K. Mohanty, Design and performance analysis of PID controller for an automatic voltage regulator system using simplified particle swarm optimization, J. Franklin Inst. 349 (8) (2012) 2609–2625.
[10] P.K. Mohanty, B.K. Sahu, S. Panda, Tuning and assessment of proportional–integral–derivative controller for an automatic voltage regulator system employing local unimodal sampling algorithm, Electr. Power Compon. Syst. 42 (9) (2014) 959–969.
[11] H. Zhu, et al., CAS algorithm-based optimum design of PID controller in AVR system, Chaos Solitons Fractals 42 (2) (2009) 792–800.
[12] L. Li, et al., An optimization method inspired by "chaotic" ant behavior, Int. J. Bifurcation Chaos 16 (08) (2006) 2351–2364.
[13] L. Li, et al., Parameters identification of chaotic systems via chaotic ant swarm, Chaos Solitons Fractals 28 (5) (2006) 1204–1211.
[14] L. Li, et al., Chaos–order transition in foraging behavior of ants, Proc. Natl. Acad. Sci. 111 (23) (2014) 8392–8397.
[15] L. Li, Y. Yang, H. Peng, Fuzzy system identification via chaotic ant swarm, Chaos Solitons Fractals 41 (1) (2009) 401–409.
[16] S. Das, et al., A novel fractional order fuzzy PID controller and its optimal time domain tuning based on integral performance indices, Eng. Appl. Artif. Intell. 25 (2) (2012) 430–442.

[17] A. Rajasekhar, R. Kumar Jatoth, A. Abraham, Design of intelligent PID/PIλDμ speed controller for chopper fed DC motor drive using opposition based artificial bee colony algorithm, Eng. Appl. Artif. Intell. 29 (0) (2014) 13–32.
[18] R. El-Khazali, Fractional-order controller design, Comput. Math. Appl. 66 (5) (2013) 639–646.
[19] I. Podlubny, Fractional-order systems and PI/sup/spl lambda//D/sup/spl mu//-controllers, IEEE Trans. Auto. Control 44 (1) (1999) 208–214.
[20] A. Biswas, et al., Design of fractional-order PIλDμ controllers with an improved differential evolution, Eng. Appl. Artif. Intell. 22 (2) (2009) 343–350.
[21] N. Aguila-Camacho, M.A. Duarte-Mermoud, Fractional adaptive control for an automatic voltage regulator, ISA Trans. 52 (6) (2013) 807–815.
[22] M. Zamani, et al., Design of a fractional order PID controller for an AVR using particle swarm optimization, Control Eng. Pract. 17 (12) (2009) 1380–1387.
[23] I. Pan, S. Das, Chaotic multi-objective optimization based design of fractional order PIλDμ controller in AVR system, Int. J. Electr. Power Energy Syst. 43 (1) (2012) 393–407.
[24] Y. Tang, et al., Optimum design of fractional order PIλDμ controller for AVR system using chaotic ant swarm, Expert Syst. Appl. 39 (8) (2012) 6887–6896.
[25] F. Merrikh-Bayat, S.-N. Mirebrahimi, M.-R. Khalili, Discrete-time Fractional-order PID Controller: Definition, Tuning, Digital Realization and Experimental Results. arXiv preprint arXiv:1405.0144, 2014.
[26] I. Pan, S. Das, Frequency domain design of fractional order PID controller for AVR system using chaotic multi-objective optimization, Int. J. Electr. Power Energy Syst. 51 (0) (2013) 106–118.
[27] K. Ogata, Modern Control Engineering, Prentice Hall, 2010.
[28] J. Kennedy, R. Eberhart, Particle swarm optimization, in: IEEE International Conference on Neural Networks, 1995.
[29] A. Ertas, Optimization of fiber-reinforced laminates for a maximum fatigue life by using the particle swarm optimization. Part II, Mech. Compos. Mater. 49 (1) (2013) 107–116.
[30] S. Yuhui, R. Eberhart, A modified particle swarm optimizer, in: The 1998 IEEE International Conference on Evolutionary Computation Proceedings, 1998. IEEE World Congress on Computational Intelligence, 1998.
[31] J. Kennedy, The particle swarm: social adaptation of knowledge, in: IEEE International Conference on Evolutionary Computation, 1997.
[32] K.J. Åström, B. Wittenmark, Computer-controlled Systems: Theory and Design, Courier Dover Publications, 2011.
[33] L. Yun, A. Kiam Heong, G.C.Y. Chong, PID control system analysis and design, IEEE Control Syst. 26 (1) (2006) 32–41.
[34] R.A. Krohling, J.P. Rey, Design of optimal disturbance rejection PID controllers using genetic algorithms, IEEE Trans. Evol. Comput. 5 (1) (2001) 78–82.
[35] Y. Del Valle, et al., Particle swarm optimization: basic concepts, variants and applications in power systems, IEEE Trans. Evol. Comput. 12 (2) (2008) 171–195.
[36] A. Oustaloup, et al., Frequency-band complex noninteger differentiator: characterization and synthesis, IEEE Trans. Circuits Syst. I Fundam. Theory Appl. 47 (1) (2000) 25–39.
[37] J. Wilkie, M. Johnson, R. Katebi, Control Engineering: an Introductory Course, Palgrave, 2003.