Neural Computing and Applications
https://doi.org/10.1007/s00521-024-09568-3

ORIGINAL ARTICLE

Improving speed control characteristics of PMDC motor drives using nonlinear PI control

Emre Çelik[1] · Güngör Bal[2] · Nihat Öztürk[3] · Erdal Bekiroglu[4] · Essam H. Houssein[5] · Cemil Ocak[3] · Gulshan Sharma[6]

Received: 18 August 2023 / Accepted: 24 January 2024
© The Author(s) 2024

Abstract
This paper introduces a nonlinear PI controller for improved speed regulation in permanent magnet direct current (PMDC) motor drive systems. The nonlinearity comes from the exponential (Exp) block placed in front of the classical PI controller, which uses a tunable exponential function to map the speed error nonlinearly. Such a configuration has not been studied till now, thus meriting further investigation. We consider an exponential PI (EXP-PI) controller and to attain the best performance from this controller, its parameters are optimized offline using salp swarm algorithm (SSA), which borrows its inspiration from the way of forage and navigation of salps living in deep oceans. To indicate the credibility of SSA tuned EXP-PI controller convincingly, numerous experiments on speed regulation in PMDC motor have been implemented using DSP of TMS320F28335. The results obtained are also compared to similar results in the literature. It is shown that the proposed approach performs well in practice by ensuring tight tracking of the speed reference and superb torque disturbance rejection for the closed loop control. Furthermore, superior performance is achieved by the proposed nonlinear PI controller with respect to a fixed-gain PI controller.

Keywords Optimization · PMDC motor drive system · Speed control · Salp swarm algorithm · And exponential PI (EXP-PI) controller

1 Introduction

Speed regulation of electric motors is a significant necessity in many industrial applications such as CNC, robotics, air conditioning, traction motors, and pumps. In all these applications, irrespective of any disturbance, the machine speed must be kept at the predefined value so that the system performs efficiently. Nowadays, direct current (DC) motors are the main horse power of several industrial plants [1]. Due to their attractive merits such as simple structure, low-cost installation, and high performance with a broad range of speed control characteristics, it is often to see these motors under service in robotics, 3D-printers, elevators, drones, cranes, electric vehicles, and other servo applications [2, 3]. To improve closed loop control precision and performance, advanced control algorithms are always welcome [4, 5].

Speed regulation of DC motors can be accomplished by controlling armature voltage or motor field current. However, in the permanent magnet DC (PMDC) motor case, the only alternative to achieve control requirements is to adjust armature voltage within the region below rated speed. For process control and industrial systems, there is enormous volume of control methodologies such as the classical ones pertaining to PID family and its cousins such as 2DOF/

✉ Emre Çelik
emrecelik@duzce.edu.tr

1 Department of Electrical and Electronics Engineering, Engineering Faculty, Düzce University, Düzce, Türkiye

2 Department of Electrical and Electronics Engineering, Engineering Faculty, Başkent University, Ankara, Türkiye

3 Department of Electrical and Electronics Engineering, Technology Faculty, Gazi University, Ankara, Türkiye

4 Department of Software Engineering, Engineering Faculty, Gazi University, Ankara, Türkiye

5 Faculty of Computers and Information, Minia University, Minia, Egypt

6 Department of Electrical Engineering Technology, University of Johannesburg, Johannesburg, South Africa

Published online: 22 February 2024

3DOF PID [6, 7], fractional order PID (FOPID) [8], fuzzy PID (FPID) [9], etc., whereas there are also other advanced algorithms based on optimal control [10], sliding mode [11], model predictive and adaptive control [12, 13], neural network [14], and H-$\propto$ [15]. Design variations and modifications to PID, FOPID and FPID controllers are also noted in the literature to have augmented system performance [16–19]. Among the controllers above, classical PID variants are the ones that are still most implemented in practice, attributable to their ease of design, low computational cost and competing performance in most cases. However, for this type of controllers, the big stone to achieving high performance is their lack to manage nonlinearities.

To resolve this problem, several PID controllers added with nonlinear feature were also introduced [20–22]. More specifically, a sigmoidal function-based nonlinear PID controller was presented in [23, 24] to be used in robotic applications. Nonetheless, the performance of the above controllers was validated limited to simulation only and their practical feasibility was not checked.

In the present work, an exponential PI (EXP-PI) controller has been applied to speed regulation in PMDC motor. In the scheme that we introduce, the nonlinear gain in cascade with the linear fixed-gain PI controller is proposed to vary exponentially. To add more flexibility, the slope or sharpness of the exponential function along with its steady state or final value can be changed. An equivalent expression for the PI law is used as the anti-windup scheme to avoid integration overflow. Also, to reasonably set the controller parameters that offer good and stable performance in practice, a suitable cost function is defined and salp swarm algorithm (SSA) is invoked offline. Given this, no human interpterion is needed during the controller design stage. Several experimental results are presented to validate the potential of SSA tuned EXP-PI controller against changes in speed reference and external load disturbance. Moreover, comparisons with the results published recently are also presented.

The remainder of this paper is structured as follows. The PMDC motor model is provided in Sect. 2. Development of the EXP-PI concept is introduced in Sect. 3, and design of the controller is presented in Sect. 4. Section 5 is devoted to the stability of the proposed closed loop system. A study based on experimental findings is performed in Sect. 6 to indicate the credibility of our proposal. Finally, some conclusions are drawn in Sect. 7.

2 PMDC motor model

The standard equivalent circuit of a DC motor is comprised of a winding resistance R_a, a self-inductance L_a, and an induced or back electromotive force (emf) e. The terminal equation is written as

$$v = e + R_a i_a + L_a \frac{di_a}{dt}, \tag{1}$$

where v is the applied armature voltage and i_a is armature current. The expression for induced emf is derived as

$$e = K \varnothing_f \omega_m, \tag{2}$$

In Eq. 2, $\varnothing_f$ is field flux per pole, ω_m is rotational speed and K is a constant based on the motor design. In PMDC motor, as the field flux is constant, the induced emf is rewritten as

$$e = K_b \omega_m, \tag{3}$$

where K_b is the induced emf constant given by

$$K_b = K \varnothing_f \text{volt} \cdot \text{sec/rad}, \tag{4}$$

Thus, the electromagnetic torque or air gap torque is obtained by the ratio of armature power P_a to ω_m, given by

$$T_e = \frac{P_a}{\omega_m} = \frac{e i_a}{\omega_m} = K_b i_a, \tag{5}$$

By modeling the load as a moment of inertia J with a viscous friction coefficient B, then the acceleration torque T_a driving the external load is calculated as

$$J \frac{\mathrm{d}\omega_m}{\mathrm{d}t} + B\omega_m = T_e - T_L = T_a, \tag{6}$$

where T_a stands for the load torque applied. Thus, the above equations form the dynamic model of the PMDC motor with load.

3 The EXP-PI controller structure

The concept behind using the Exp block before PI controller is presented in this section. In fact, this block functions as a variable gain k, which is a nonlinear function of the error e. When $k = 1$, the resulting controller is a classical PI controller with fixed gains. On the other hand, when k is a nonlinear function of the error, i.e., exponential function, the error signal is either amplified or attenuated using the nonlinear gain based on the magnitude of error. By acting on the error signal, the Exp block produces a nonlinear input signal for the PI controller as in Fig. 1.

As seen in Fig. 1, the Exp block receives the error signal and weighs it by $\pm 1/\tau$ based on the error sign. Employing an exponential function performing as an activation

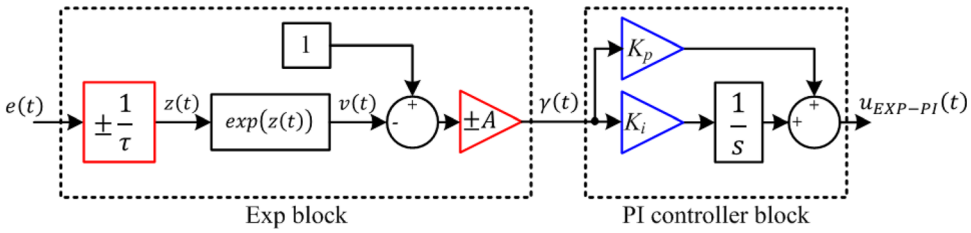


Fig. 1 The concept of EXP-PI controller

function like in a neural network architecture, a nonlinear transformation is obtained. The nonlinear behavior of this controller lies in the nature of this function. There is also one gain, namely A, to scale the output of Exp block and obtain $\gamma(t) = k \cdot e(t)$, which is the nonlinearly transformed signal of e. $\gamma(t)$ is then fed to the PI controller block and eventually the control signal $u_{\mathrm{EXP-PI}}(t)$ is generated from the EXP-PI controller output. Herein, the mathematical expression for the proposed nonlinear gain k based on exponential function can be given by

$$k = \begin{cases} A(1 - \exp(-e(t)/\tau)) & e(t) > 0 \\ -A(1 - \exp(e(t)/\tau)) & e(t) \leq 0 \end{cases}, \tag{7}$$

where $\pm A$ are the saturation limits of the function and τ is added to have an adjustable slope or sharpness. As seen in Fig. 2, A is, in fact, a gain that scales the output of the function, whereas the slope of the function can be changed proportionally to τ.

Both A and τ are considered in Eq. 7 to enable us have more degrees of freedom to design EXP-PI controller and accordingly have better system performance.

Using the nonlinear gains shown in Fig. 2, we let PI controller gains become adaptive. That is, it enables the controller to adapt its response to changeable conditions in the closed loop control system. When the error between the reference and actual values of the controlled variable is large, the gain amplifies the error to produce a large corrective control action so that the system output is driven to its reference hastily. As the time pasts and the error diminishes, the gain is decreased and thus excessive oscillations and unwanted overshoots in the response are prevented in this manner. As a result, the proposed controller is capable of accomplishing the two contradictory requirements of fast response (high initial gain) and non-oscillatory behavior (low gain), an ability that is not achievable at the same time by the classical fixed-gain PI controller. This conclusion is also valid when the system is exposed to disturbances whereby the nonlinear gain provides the system with remarkable disturbance rejection capacity.

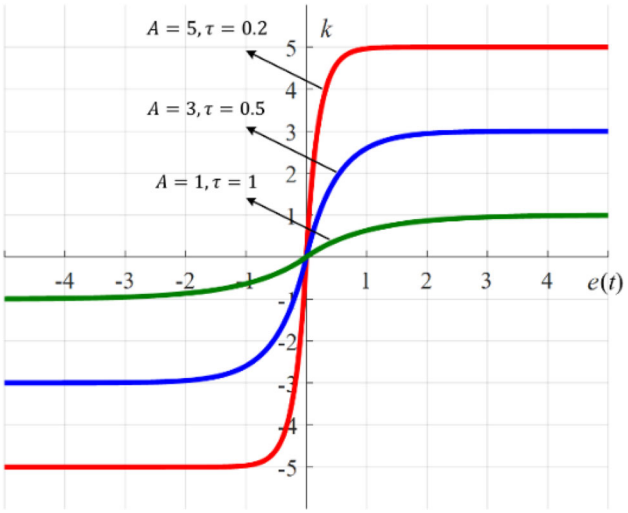


Fig. 2 Variation of the gain k as an exponential function of the error $e(t)$ for different values of A and τ

In the EXP-PI controller, beside K_p and K_i, A and τ are also the important unknowns. All these gains need to be optimized properly to achieve improved performance. The best way to accomplish this task is to employ a suitable metaheuristic algorithm.

4 Improved speed servo system

In the current study, the performance assessment of the proposed concept is made by applying it to PI control scheme, which is more acceptable and effective in industrial applications. So, the new controller takes the name "Exponential PI (EXP-PI)". In this controller, PI controller law is written as

$$u_{\mathrm{EXP-PI}}(t) = K_p \gamma(t) + K_i \int_0^t \gamma(t)\mathrm{d}t, \tag{8}$$

Although Eq. 8 yields good control effort in simulation where the controller output is not bounded, it does not work satisfactorily in real-time experimentation because of the integration wind-up problem. To evade from this problem, we take the time derivative of two sides of Eq. 8 as

$$\frac{u_{\mathrm{EXP-PI}}(k) - u_{\mathrm{EXP-PI}}(k-1)}{T} = K_p \frac{\gamma(k) - \gamma(k-1)}{T} + K_i \gamma(k), \tag{9}$$

where T is sample time with $T > 0$, (k) and $(k-1)$ denote

the current and previous sample time, respectively. Multiplying both sides of Eq. 9 by T, we find:

$$u_{\mathrm{EXP-PI}}(k) - u_{\mathrm{EXP-PI}}(k-1) = K_p[\gamma(k) - \gamma(k-1)] + K_i T\gamma(k), \quad (10)$$

In Eq. 10, the difference between the (k) and $(k-1)$ samples indicate a change for the given variable. So, Eq. 10 can be rewritten as

$$\Delta u_{\mathrm{EXP-PI}}(k) = K_p \Delta\gamma(k) + K_i T\gamma(k), \quad (11)$$

Notice from Eq. 11 that the controller rule does not contain any integration term and thus eliminates the problem of excessive accumulation owing to the integral action. It generates the change of control action, not the actual value. So, to obtain the actual control action $u_{\mathrm{EXP-PI}}(k)$, it requires to be accumulated by the previous control action as

$$u_{\mathrm{EXP-PI}}(k) = \Delta u_{\mathrm{EXP-PI}}(k) + u_{\mathrm{TSPI}}(k-1), \quad (12)$$

Therefore, the EXP-PI controller has been provided the feature of anti-windup. In the next subsection, we explain a useful guideline to select controller gains properly so that the highest possible performance from the controller is achieved.

4.1 Designed cost function

It is noticeable from Eqs. 7 and 11 that the developed EXP-PI controller demands tuning the parameters A and τ in addition to K_p and K_i. As this is not an easy task, we address the problem by considering it from the viewpoint of optimization to evade from the risk of misleading performance owing to inappropriate parameter tuning based on trial-and-error.

To evaluate the fitness of each individual during optimization, we need to measure the quality of controlled responses quantitatively. Four commonly used error-based performance criteria to achieve this goal are integral squared error (ISE), integral absolute error (IAE), integral time-multiplied absolute error (ITAE) and integral time-multiplied squared error (ITSE). Among these, because ITAE and ITSE weight error with time, they consider errors at the start of the response less significantly than those occurring after a while. As such, they tend to produce systems with sluggish initial response. Squaring the error, ISE can eliminate large errors hastily, leading to a fast response, but with some considerable oscillation. As concerned to IAE, it yields less oscillation, but comparably slower response than ISE. As a result, to accomplish the two contradictory requirements of a fast response and no or mild overshoot, combination of ISE and IAE has been considered as a cost function in [25, 26] and also in the current paper. It can be also termed as figure of demerit (FOD), which is expressed in Eq. 13.

$$\mathrm{FOD} = \omega_1 \int_0^{t_s} e_s^2 \mathrm{d}t + \omega_2 \int_0^{t_s} |e_s| \mathrm{d}t, \quad (13)$$

where e_s is speed error, t_s is the total simulation time in sec, $\omega_i (i = 1, 2)$ is a weighting factor deciding the significance of each term in Eq. 13. It is found that good results are obtained when $\omega_1 = 0.01$ and $\omega_2 = 1$.

The two terms on the right-hand side of Eq. 13 are responsible for reducing the speed error as to enhance the speed response profile. Consequently, the optimization constraints in the current study are the EXP-PI controller gains. These are restricted to a particular range during the course of optimization. Thus, this paper considers to adjust the controller gains optimally by formulating it as an optimization problem as:

Minimize FOD, subject to:

$$K_p^{\min} \le K_p \le K_p^{\max}, K_i^{\min} \le K_i \le K_i^{\max}, A^{\min} \le A \le A^{\max}, \tau^{\min} \le \tau \le \tau^{\max}, \quad (14)$$

The superscripts min and max in Eq. 14 denote the minimum and maximum values of the tunable variables, which are set to 0 and 2 for K_p, A, τ, and to 0 and 20 for K_i. To accomplish the optimization task, SSA has been applied as the recent and powerful optimizer.

4.2 Salp swarm algorithm

Salp swarm algorithm (SSA) is an optimization method recently introduced in [27]. The goal of SSA is to resolve optimization problems by mimicking the swarming behavior of salps living in oceans. Balanced exploration and exploitation propensities of this algorithm render it attractive for several applications needing for optimization. Compared to other existing algorithms, SSA is simple to code and uses relatively less computational resources. The algorithm is mainly guided by only one parameter whose value is reduced gradually during iterations to promote exploration for new areas in the initial stages and exploitation around a promising area in the final stages of optimization. Since its release, SSA has been found very prolific in diverse application areas such as classification [28], optimal conductor selection in a practical radial distribution system [29], optimization of power system stabilizer [30]/size of CMOS differential amplifier [31]/PV systems [32], optimal parameter identification of fuel cells [33], load frequency control [18, 19], economic load dispatch [34], etc. Good solution accuracy, fast convergence rate and low computation cost stressed in those researches are the attractive aspects of SSA, and the reasons why we

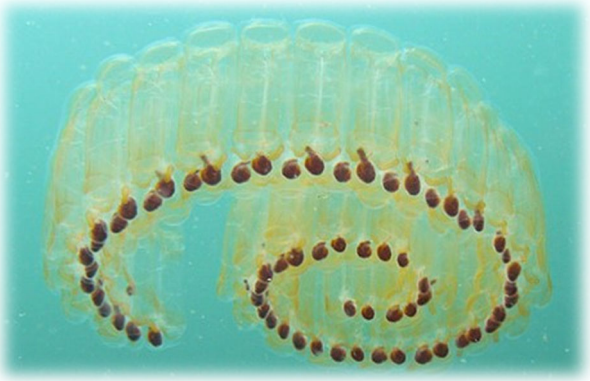


Fig. 3 Chain of the salps

prefer it to optimize the parameters of EXP-PI controller in the current study.

In SSA, the population is divided into two groups as leader and followers. The leader is the salp standing at the top of the chain, attacking in the direction of food source F, while the remaining salps are all considered as followers. They form a salp chain as in Fig. 3 while navigating and foraging in deep oceans.

For n salps (search agents) in a d-dimensional search space, the population of salps X can be generated by a $n \times d$ matrix as:

$$X = \begin{bmatrix} x_1^1 & x_1^2 & \dots & x_1^{d-1} & x_1^d \\ x_2^1 & x_2^2 & \dots & x_2^{d-1} & x_2^d \\ \vdots & \vdots & & \vdots & \vdots \\ x_n^1 & x_n^2 & \dots & x_n^{d-1} & x_n^d \end{bmatrix}, \tag{15}$$

The following equation is used to update the position of the leader.

$$x_1^j = \begin{cases} F^j + p\left(\mathrm{lb}^j + q\left(\mathrm{ub}^j - \mathrm{lb}^j\right)\right), & r \geq 0.5 \\ F^j - p\left(\mathrm{lb}^j + q\left(\mathrm{ub}^j - \mathrm{lb}^j\right)\right), & r < 0.5 \end{cases}, \tag{16}$$

where x_1^j and F^j denote the position of the leader and food source in the j th dimension, respectively. lb^j and ub^j are the lower and upper bounds of the j th dimension for the problem. q and r are random numbers from the normal distribution 0–1. F is the best-obtained solution and it is the chain's target. p is the main parameter of the algorithm defined in Eq. 17 [35].

$$p = 2e^{-(4t/T)2} \tag{17}$$

where t and T stand for the current iteration and maximum iteration number, respectively. While t increases, the value of p is reduced to achieve an equilibrium between exploration and exploitation. That is, when p is high in the earlier iterations, Eq. 16 allows the search agents to move by large steps (exploration) whereas they are forced to move over a small range (exploitation) as the value of p reduces in the last iterations. As concerned to followers ($i \geq 2$), their position update rule is given by

$$x_i^j = \frac{1}{2}\left(x_i^j + x_{i-1}^j\right), \tag{18}$$

where x_i^j is the location of the i th salp in the j th dimension. The pseudocode of SSA is provided in Algorithm 1.

Algorithm 1 Pseudocode of SSA

```
Randomly generate the salp population X
while (termination criteria are not met)
  Evaluate the fitness of each search agent (salp)
  Set F as the best individual
  Update p using Eq. 17
  for every salp
   if (i == 1)
     Update the position of the leading salp using Eq. 16
   else
     Update the position of the followers using Eq. 18
   endif
  endfor
  Check if the search agents violate the boundary constraints
endwhile
return F
```

4.3 Offline application of SSA to optimization of controller parameters

Using the drive system model and motor model in Section II, simulations are performed in a computer and the proposed EXP-PI controller parameters are obtained effectively by employing SSA. We do not explicitly know that the parameter values that we found are optimal, but we expect them as they are. The controller parameter values are then inserted into the DSP to make real-time implementations.

The EXP-PI controller has four tunable parameters; A and τ for Exp block, and K_p and K_i for PI controller. Thus, each search agent in SSA is composed of these parameters. The SSA-based optimization process starts by initializing a number of salps randomly within their specified interval. For evaluation of cost function in Eq. 13, each candidate solution is used in the simulation model to obtain speed response from the controller. After assigning a fitness value, the salps are sent back to the optimization algorithm to evolve for the next iteration. Herein, for each cost evaluation, it is assumed that load torque and speed reference are changed from 0 to 1.2 Nm and from 0 to 2750 rpm, in steps of 0.2 Nm and 250 rpm, respectively (the upper limits are chosen based on the motor nameplate). Therefore, there are 84 different cases and the final cost is calculated as mean of these 84 costs. This is expressed mathematically by

$$J = \left(\sum_{i=1}^{\mathrm{NC}} \mathrm{FOD}_i\right)/\mathrm{NC}, \tag{19}$$

where NC is the number of cases equal to 84 and J is the mean cost for each search agent in SSA. By calculating the overall performance of solutions during optimization, their robustness against changeable system conditions is improved. In doing so, the requirement for resetting the controller parameters is alleviated when a change in load torque and/or speed reference occurs.

To provide more insights into the offline optimization of controller parameters and elucidate the above discourse further, the execution of SSA is depicted in detail in Fig. 4. Note that all these steps are implemented in a computer and the final solution returned by the algorithm including the best controller parameters are used and tested in experimental work.

By following the above concept, SSA is run 50 times and the best controller parameters attained at the minimum value of J over 50 runs are provided in Table 1.

In the algorithm, the number of search agents is 50 while the maximum iteration number is set to 100. To show the convergence capacity of SSA, a few convergence graphs out of 50 runs with random initializations are saved and shown in Fig. 5. Note that the lower the value of J, the better the controller parameters and the higher performance of the system we have. As seen, after a sharp drop in the initial stage of optimization, only small improvements on the solution quality are obtained because the best solution remains relatively constant around 48.8. Although the initial points are different, SSA is able to converge to a similar final solution within 40–50 iterations in each run. This outcome justifies that the selection of maximum iteration as 100 in the present research is sufficient for SSA to approach toward the global optimum solution.

Table 1 Optimum values of EXP-PI controller parameters

A	τ	K_p	K_i	J
0.425	0.01	0.192	19.970	48.898

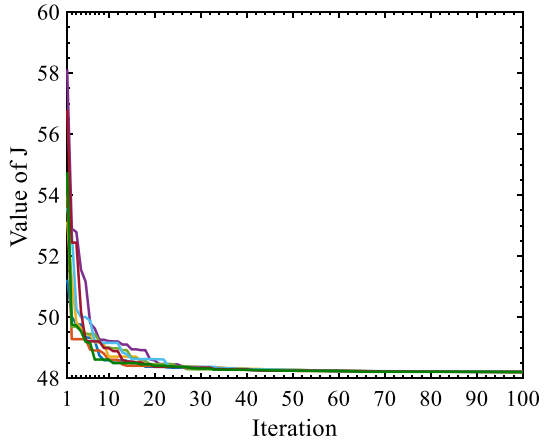


Fig. 5 SSA-based convergence trends of J

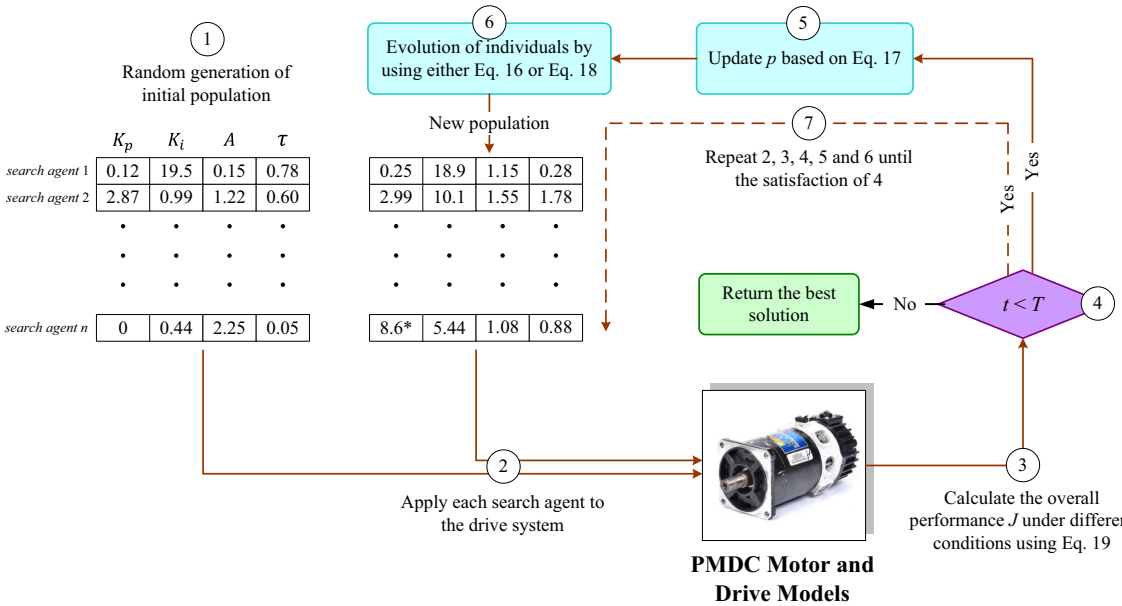


Fig. 4 Detailed execution of SSA during the optimization of controller parameters

5 Stability assessment

The stability analysis of control systems such as feedback loops is of primary importance to ensure that the system under control responds boundedly to any limited input [36, 37]. To address the stability concern, we consider the entire system as a linear time-invariant system. The open loop transfer function (TF) of the dynamical model for the DC motor is given in Eq. 20. The input and output of this function are voltage $v(s)$ and angular speed $\omega(s)$, respectively.

$$G_{\text{DCmotor}}(s) = \frac{\omega(s)}{v(s)} = \frac{k_t}{(\text{Ls} + R)(Js + B) + k_t k_e}, \tag{20}$$

where J is the motor inertia, B is the viscous-friction coefficient of the rotor shaft, R is the armature resistance, L is the motor winding resistance, k_t is the motor torque constant, which also equals to the motor back emf constant k_e. The following TF relating control input signal $u_{\text{PI}}(s)$ to processed speed error $\widehat{e}(s)$ can be written for a classical PI controller.

$$G_{\text{PI}}(s) = \frac{u_{\text{PI}}(s)}{\widehat{e}(s)} = K_p + \frac{K_i}{s} = \frac{K_p s + K_i}{s}, \tag{21}$$

As concerned to the exponential term that we use in front of the PI control scheme, for a step input, the speed error is positive (assuming no overshoot to simplify the analysis) and the nonlinear gain k in Eq. 7 becomes:

$$k = A(1 - \exp(-e(t)/\tau)), \tag{22}$$

For the Laplace transform of Eq. 22, we can arrive at:

$$G_{\text{EXP}}(s) = \frac{u_{\text{EXP}}(s)}{e(s)} = \frac{A \cdot 1/\tau}{s + 1/\tau}, \tag{23}$$

Using the relevant motor and controller parameters, the TF block diagram of the entire PMDC motor speed servo system is demonstrated in Fig. 6.

Consequently, the system has the following closed loop TF:

$$\frac{\omega(s)}{r(s)} = \frac{2.2114s + 230.35}{9.57 \cdot 10^{-7}s^4 + 1.1117 \cdot 10^{-3}s^3 + 0.176s^2 + 9.6514s + 230.35}, \tag{24}$$

It is apparent from Eq. 24 that the studied system is a fourth order. It has a zero at -104.16 and poles at $\{-985.5, -102.12, -37.18 \pm 3176i\}$. We see that all the poles have negative real parts and located far away from left to the imaginary axis in the complex plane. This finding completes the stability proof, assuring that our system is stable. The step response of the system represented by Eq. 24 is plotted in Fig. 7.

6 Experimental verification

In this section, the performance of the developed EXP-PI control scheme is evaluated on a DC motor drive system. The system block diagram is depicted in Fig. 8. It consists of a H-bridge converter to control the speed and rotation of motor, PMDC motor, permanent magnet DC generator (PMDG), DSP and other blocks that organize and synchronize the operation of the whole system. The EXP-PI controller works as the speed controller, whereas hysteresis current controller is used in the inner loop to regulate the motor armature current. PMDG is coupled with PMDC motor to apply external torque disturbances at the motor shaft, when required. As it is a PMDG, the output power and accordingly torque disturbance is speed-dependent. That is, the higher the speed, the greater the load torque.

The physical system showing the experimental prototype is given in Fig. 9. The PMDC motor model Sanyo Denki R730T-042EL7 is the motor to be controlled and the generator model Sanyo Denki T730B-569EL8N is used. The dc-link voltage is provided from Magna-Power DC power supply. In the converter, IGBTs of IXGH48N60C3D1 are adopted and the motor control algorithm is implemented in DSP of TMS320F28335 from Texas Instruments. The tested motor has a built-in tacho-generator with a voltage constant of 7.4 V/krpm, which is utilized to obtain speed responses. As there is no available torque sensor in our laboratory, we could not measure the load torque in experiments. Instead, motor armature current proportional to the applied load torque is measured with the Fluke 80i-110 s current probe. Both the speed and motor current are viewed and recorded by the Tektronix DPO 2024B oscilloscope.

The PMDC motor nominal parameters are reported in Table 2.

In order to verify the feasibility and effectiveness of the EXP-PI controller, the PMDC motor was tested under two cases of experimentation such as speed regulation and torque disturbance rejection. In speed regulation test, constant/variable step speed references and reference

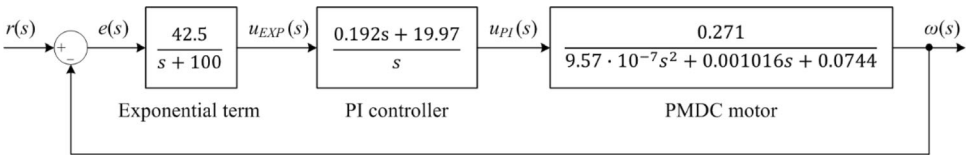


Fig. 6 TF block diagram of the entire PMDC motor speed servo system

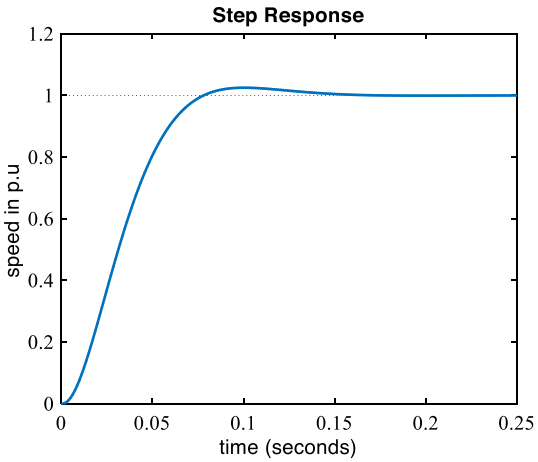


Fig. 7 Step response of the system

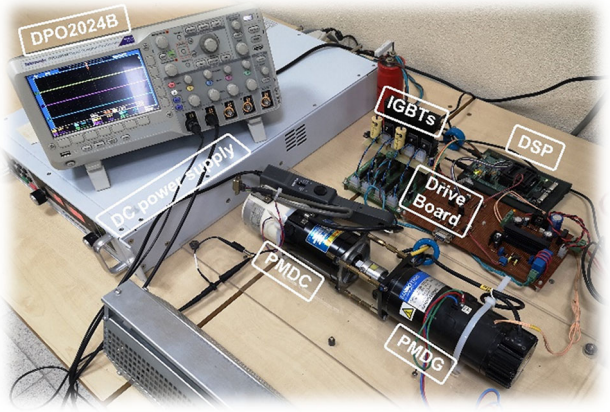


Fig. 9 The experimental system

trajectory are considered to be tracked. To show the contribution of our proposal, a comparison study with an anti-windup fixed-gain PI control scheme published previously [25] is also established. Symbiotic organisms search (SOS) algorithm was adopted in the respective paper to search for the PI controller parameters optimally, which yielded $K_p = 0.265$ and $K_p = 20$. All the results for the EXP-PI controller are obtained when the controller parameters remain fixed as in Table 1. This will also show us how robustly this controller performs.

6.1 Speed regulation performance

In the first test, speed control is performed when the step speed reference is $n^* = 1500$ rpm and the PMDG terminals are open, generating no any torque. The resulting speed and current responses are displayed in Fig. 10. The desired speed and measured speed, n^*, n, are shown with the red and turquoise traces, respectively, with a scaling factor of 270.27 rpm/div. The blue trace shows the motor armature current with a scaling factor of 5 A/div. This current

Table 2 Sanyo Denki R730T-042EL7 motor nominal parameters

Variable	Value
Rated power	300 W
Rated armature current	4.8 A
Rated voltage	80 V
Rated speed	2500 rpm
Rated torque	1.15 Nm
Armature resistance (R_a)	1.75 Ω
Electrical time constant (τ_e)	1.5 ms
Voltage constant (k_e)	28.6 V/kmin^{-1}
Nominal inertia (J)	5.8 10^{-4} kg.m^2
Encoder resolution	4000 ppr

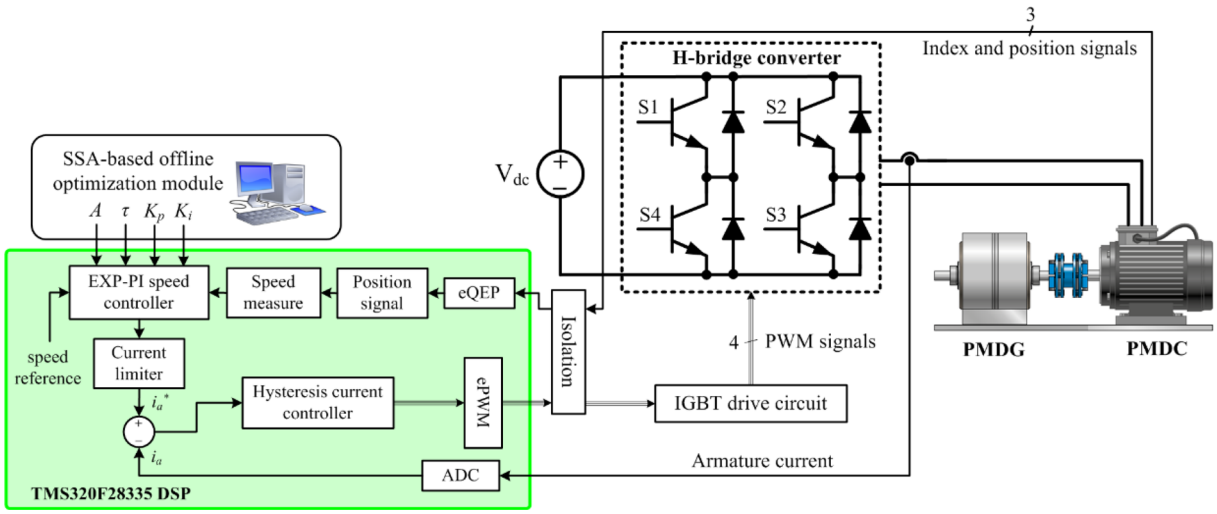


Fig. 8 The system block diagram

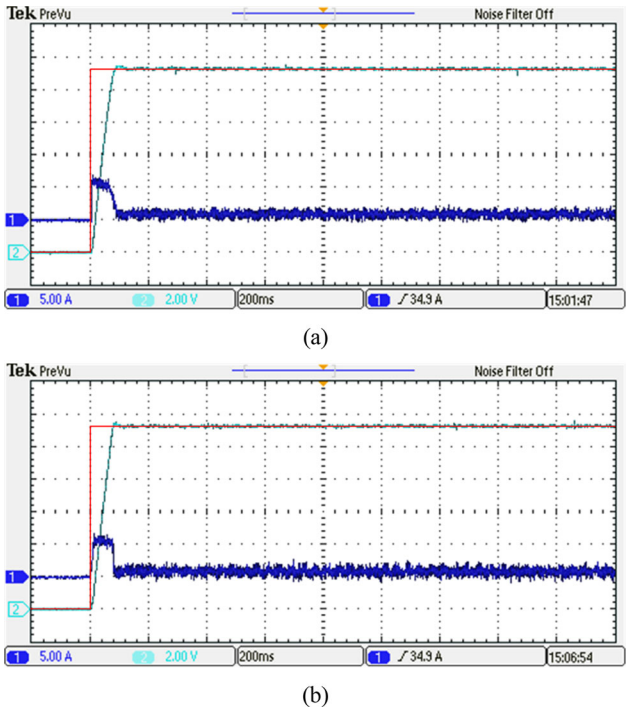


Fig. 10 Experimental responses to a step speed reference of 1500 rpm **a** PI controller **b** EXP-PI controller

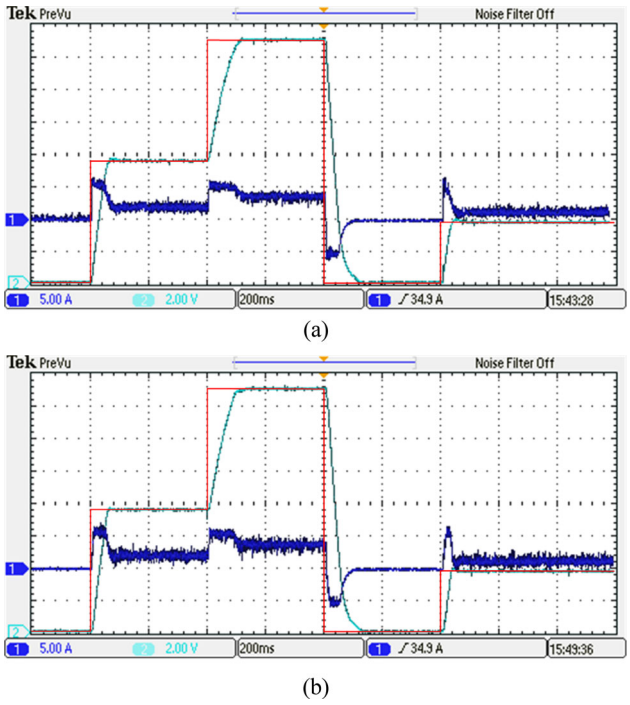


Fig. 11 Experimental responses to random step speed references **a** PI controller **b** EXP-PI controller

scaling is used in all of the reported experiments below. Although the rise times of the speed responses seem similar for both controllers, the EXP-PI controller gives a better dynamic response with less overshoot and shorter settling time. This is attributed to the fact that the time required for the current to drop to its steady state value is shorter in the EXP-PI controller comparing with that in the PI controller. It should be also noticed that this response is comparable with that derived theoretically and displayed in Fig. 7

In the following, the system transient and steady-state responses against step random changes in speed reference is analyzed. The speed reference is initially set to zero for the first 0.2. Afterward, it is set to 1000 rpm at $t = 0.2$s and 2000 rpm at $t = 0.6$ s and 0 rpm at $t = 1$ s and eventually 500 rpm at $t = 1.4$ s. In this test, the PMDG terminals are closed, thus it applies torque disturbance proportional to the rotational speed. Critical observation of the responses in Fig. 11 reveals that both controllers is able to yield similar steady state performance because the motor speed in both cases tracks its reference tightly at steady state. Nonetheless, as concerned to the transient state, the EXP-PI controller performs better than the PI controller with respect to the overshoot generation and settling time. It is also noticeable in this figure that the motor current becomes different following each speed reference because of the load torque that changes all the time as the speed varies.

In the final speed regulation exercise, both controllers are used to track a desired trajectory. The reference trajectory considered for this test is $n^* = 1500 \times (\sin(t) + \cos(0.5t))$, and it is shown in Fig. 12. This signal has a period of $T \cong 12.55$ s.

Figure 13 depicts the actual trajectories and motor currents. As in the previous experiment, the PMDG terminals are closed. To display the measured speed adequately, the speed scaling factor has been changed to 675.67 rpm/div in this test. It is worth noting that during the experiment speed regulation with the PI controller was lost intermittently as seen in Fig. 13a. In such instants, PI controller could not

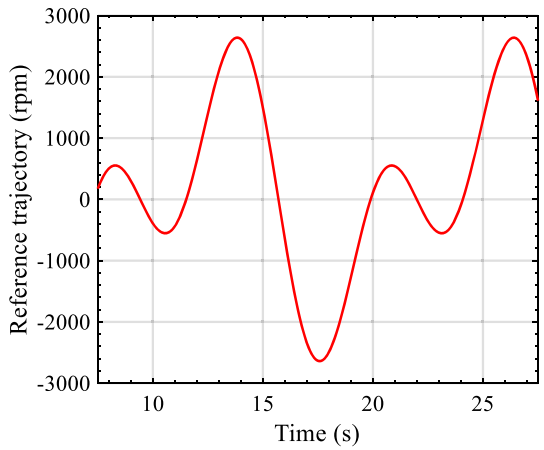


Fig. 12 Desired trajectory

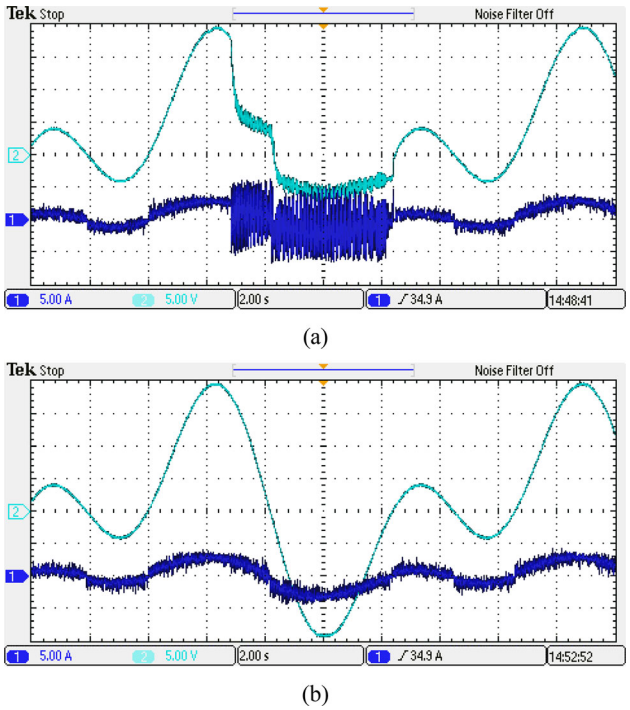


Fig. 13 Experimental responses to reference trajectory **a** PI controller **b** EXP-PI controller

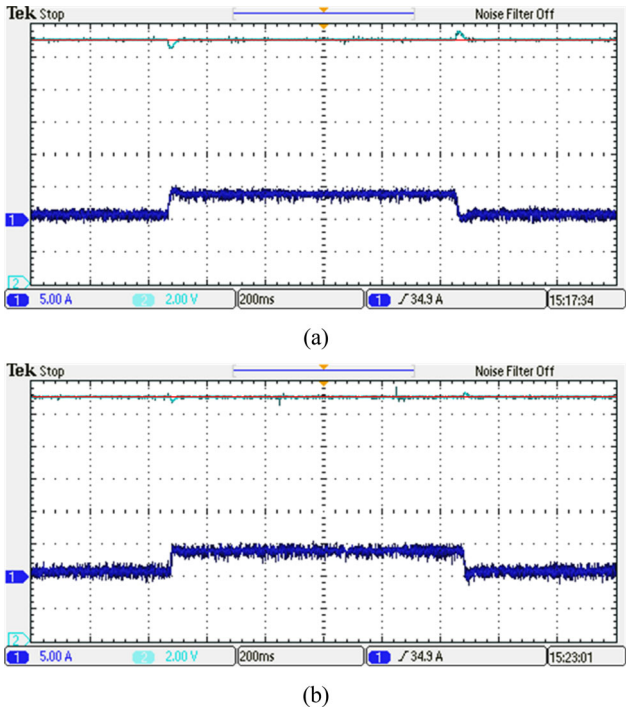


Fig. 14 Experimental responses under load disturbance **a** PI controller **b** EXP-PI controller

perform stably and it yields a totally different response than the desired trajectory. As the controller is not capable of generating the reference current properly, the motor current is found to change abruptly within allowable limits. On the other hand, response in Fig. 13b clearly shows the capacity and potential of the EXP-PI controller in allowing the system to track the reference trajectory tightly. This claim can be confirmed by measuring the period of the speed response as $T \cong 12.55$ s, which is equal to that in Fig. 12. Here, the motor current is controlled such in a way that the system tracks the desired trajectory.

6.2 Torque disturbance rejection ability

In this test, the PMDG is allowed to introduce some torque disturbances at steady state by switching its terminals on and off, respectively, while the motor rotates at 2000 rpm. When the PMDG terminals are closed, the motor current rises up to 4 A as to compensate the increased load torque. As we can see in Fig. 14a, speed response obtained with the PI controller deviates notably from and settles again to its reference after a while following torque disturbances. However, the EXP-PI controller is found to achieve a far better disturbance rejection as in Fig. 14b, where the speed response is recovered quickly by rejecting torque disturbance very fast. This affirms that the EXP-PI controller has an outstanding robustness to abrupt load changes.

7 Conclusion

A fast step response without oscillations and good torque disturbance rejection are the main objectives to be met in a perfect control system. Satisfying one of these requirements undermines the performance of others when linear controllers are utilized. This paper presents a simple solution to this problem by introducing an exponential PI controller, taking advantages of a nonlinear gain that is a function of the error. The nonlinear characteristic of this gain comes from the exponential function designed. So, EXP-PI controller is simply obtained by adding an Exp block in front of the classical PI controller. The performance of the resulting controller is shown for speed regulation in PMDC motors. As optimizer, SSA is applied offline to search for optimal parameter values of the EXP-PI controller without human interpretation. Performance obtained is checked under several experiments where a comparison with a published PI control scheme is also presented. It is found that the EXP-PI controller enjoying the automatic adjustment of the gain performs satisfactorily in all the conditions of real-time implementation. Moreover, from the comparative results it is revealed that the EXP-PI controller achieves a more desirable level of performance in both speed reference tracking and disturbance attenuation. As a result, our proposal outshines its classical and published counterpart significantly without compromising on simplicity much.

Acknowledgements Authors thank the Scientific Research Projects Coordination Unit of Düzce University for supporting this research work partially by Grant 2022.06.03.1291.

Funding Open access funding provided by the Scientific and Technological Research Council of Türkiye (TÜBİTAK).

Data availability The research data presented in the work can be made available on reasonable request.

Declarations

Conflict of interest No conflict of interest is declared in this work.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/.

References

1. Ismail AAA, Elnady A (2019) Advanced drive system for DC motor using multilevel DC/DC buck converter circuit. IEEE Access 7:54167–54178
2. Çelik E, Gör H (2019) Enhanced speed control of a DC servo system using PI + DF controller tuned by stochastic fractal search technique. J Franklin Inst 356(3):1333–1359
3. Lee CR, Kim S-K, Ahn CK (2020) Auto-tuning proportional-type synchronization algorithm for DC motor speed control applications. IEEE Trans Circuits Syst II Express Briefs 67(3):521–525
4. Ghosh S, Ghosh M, Panda GK, Saha PK (2018) Mechanical contactless computational speed sensing approach of PWM operated PMDC brushed motor: a slotting-effect and commutation phenomenon incorporated semi-analytical dynamic model-based approach. IEEE Trans Circuits Syst II Express Briefs 65(1):81–85
5. Darba A, Belie FD, D'haese P, Melkebeek JA (2016) Improved dynamic behavior in BLDC drives using model predictive speed and current control. IEEE Trans Ind Electron 63(2):728–740
6. Guha D, Roy PK, Banerjee S (2022) Quasi-oppositional JAYA optimized 2-degree-of-freedom PID controller for load-frequency control of interconnected power systems. Int J Model Simul 42(1):63–85
7. Jena NK, Sahoo S, Sahu BK, Patel NC, Mohanty KB (2020) Optimal design of a three degrees of freedom based controller for AGC of a power system with renewable energy sources. In: Proc. Int. Conf. Comput. Intell. Smart Pow. Syst. Sustainable Energ., Odisha, India, 29–31, pp. 1−6
8. Fathy A, Yousri D, Rezk H, Thanikanti SB, Hasanien HM (2022) A robust fractional-order PID controller based load frequency control using modified hunger games search optimizer. Energies 15(1):361
9. Barakat M (2022) Optimal design of fuzzy-PID controller for automatic generation control of multi-source interconnected power system. Neural Comput Appl 34:18859–21880
10. Chen X, Lin J, Liu F, Song Y (2019) Optimal control of AGC systems considering non-gaussian wind power uncertainty. IEEE Trans Power Syst 34(4):2730–2743
11. Chairez I, Utkin V (2022) Direct current motor position control by a sliding mode controlled dual three-phase AC-DC power converter. IFAC-PapersOnLine 55(9):333–338
12. Gulbudak O, Gokdag M, Komurcugil H (2023) Lyapunov-based model predictive control of dual-induction motors fed by a nine-switch inverter to improve the closed-loop stability. Int J Electr Power Energy Syst 146:108718
13. Mohamed TH, Alamin MAM, Hassan AM (2020) Adaptive position control of a cart moved by a DC motor using integral controller tuned by Jaya optimization with Balloon effect. Comput Electr Eng 87:106786
14. Nizami TK, Gangula SD, Reddy R, Dhiman HS (2022) Legendre neural network based intelligent control of DC-DC step down converter-PMDC motor combination. IFAC-PapersOnLine 55(1):162–167
15. Schwerdtner P, Voigt M (2023) Fixed-order H-infinity controller design for port-Hamiltonian systems. Automatica 152:110918
16. Çelik E, Öztürk N, Arya Y, Ocak C (2021) (1 + PD)-PID cascade controller design for performance betterment of load frequency control in diverse electric power systems. Neural Comput Appl 33:15433–15456
17. Çelik E (2021) Design of new fractional order PI–fractional order PD cascade controller through dragonfly search algorithm for advanced load frequency control of power systems. Soft Comput 25:1193–1217
18. Çelik E (2022) Performance analysis of SSA optimized fuzzy 1PD-PI controller on AGC of renewable energy assisted thermal and hydro-thermal power systems. J Ambient Intell Humaniz Comput 13:4103–4122
19. Çelik E, Öztürk N (2022) Novel fuzzy 1PD-TI controller for AGC of interconnected electric power systems with renewable power generation and energy storage devices. Eng Sci Technol Int J 35:101166
20. So G (2020) Design of an intelligent NPID controller based on genetic algorithm for disturbance rejection in single integrating process with time delay. J Mar Sci Eng 9(1):25
21. Su Y, Sun D, Duan B (2005) Design of an enhanced nonlinear PID controller. Mechatronics 15(8):1005–1024
22. Tian Y-C, Tade MO, Tang J (1999) A nonlinear PID controller with applications. IFAC Proceed 32(2):2657–2661
23. Meza JL, Santibanez V, Soto R, Llama MA (2011) Fuzzy self-tuning PID semiglobal regulator for robot manipulators. IEEE Trans Ind Electron 59(6):2709–2717
24. Seraji H (1998) A new class of nonlinear PID controllers with robotic applications. J Robot Syst 15(3):161–181
25. Çelik E, Öztürk N (2018) First application of symbiotic organisms search algorithm to off-line optimization of PI parameters for DSP-based DC motor drives. Neural Comput Appl 30:1689–1699
26. Çelik E, Öztürk N (2017) Optimal setting of PI parameters for direct current motor drives by symbiotic organisms search algorithm. J Information Technol 10(3):311–318
27. Mirjalili S, Gandomi AH, Mirjalili SZ, Saremi S, Faris H, Mirjalili SM (2017) Salp swarm algorithm: a bio-inspired optimizer for engineering design problems. Adv Eng Softw 114:163–191
28. Hussien AG, Hassanien AE, Houssein EH (2017) Swarming behaviour of salps algorithm for predicting chemical compound activities. In: Proc. IEEE 8th Int. Conf. Intell. Comput. Inf. Syst., Cairo, Egypt, 05–07, pp 315−320

29. Ismael S, Aleem S, Abdelaziz A, Zobaa A (2018) Practical considerations for optimal conductor reinforcement and hosting capacity enhancement in radial distribution systems. IEEE Access 6:27268–27277
30. Ekinci S, Hekimoglu B (2018) Parameter optimization of power system stabilizer via salp swarm algorithm. In: Proc. 5th Int. Conf. Electr. Electron. Eng., Istanbul, Turkey, 3–5, pp 143−147
31. Asaithambi S, Rajappa M (2018) Swarm intelligence-based approach for optimal design of cmos differential amplifier and comparator circuit using a hybrid salp swarm algorithm. Rev Sci Instrum 89(5):054702
32. Ridha HM, Gomes C, Hizam H, Mirjalili S (2020) Multiple scenarios multi-objective salp swarm optimization for sizing of standalone photovoltaic system. Renew Energ 153:1330–1345
33. El-Fergany AA (2018) Extracting optimal parameters of PEM fuel cells using salp swarm optimizer. Renew Energ 119:641–648
34. Reddy YVK, Reddy MD (2018) Solving economic load dispatch problem with multiple fuels using teaching learning based optimization and salp swarm algorithm. J Intell Syst Theory Appl 1(1):5–15
35. Çelik E, Öztürk N, Arya Y (2021) Advancement of the search process of salp swarm algorithm for global optimization problems. Expert Syst Appl 182:115292
36. Fatoorehchi H, Djilali S (2023) Stability analysis of linear time-invariant dynamic systems using the matrix sign function and the Adomian decomposition method. Int J Dyn Control 11:593–604
37. Fatoorehchi H, Ehrhardt M (2023) A combined method for stability analysis of linear time invariant control systems based on Hermite-Fujiwara matrix and Cholesky decomposition. Can J Chem Eng 101:7043–7052

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.