Soft Computing
https://doi.org/10.1007/s00500-024-09693-0

OPTIMIZATION

Optimization of the different controller parameters via OBL approaches based artificial ecosystem optimization involving fitness distance balance guiding mechanism for efficient motor speed regulation of DC motor

Evren Isen[1] · Serhat Duman[1]

Accepted: 17 January 2024

Abstract
This study proposes a new optimization approach, which is called as artificial ecosystem optimization algorithm with fitness-distance balance guiding mechanism by using opposite based learning methods (FDBAEO_OBLs) for the speed regulation of direct current (DC) motor. The performance of the proposed FDBAEO_OBL algorithm is tested in two different experimental studies. In the first experimental study, the proposed approach is tested in the CEC2020 benchmark test functions and the FDBAEO algorithm, which included the best OBL approach, is determined using non-parametric Wilcoxon and Friedman statistical analysis methods. Second, the parameters of proportional integral derivative (PID), tilt integral derivative (TID), proportional integral derivative with filter (PIDF), tilt integral derivative with filter (TIDF), fractional-order proportional integral derivative (FOPID), fractional-order proportional integral derivative with filter (FOPIDF), proportional integral derivative with fractional-order filter (PIDFF) and fractional-order proportional integral derivative with fractional-order filter (FOPIDFF) controller structures to be used in DC motor closed loop speed control are determined with FDBAEO_OBL, and the performances of the controllers are investigated. Integral absolute error (IAE), integral time absolute error (ITAE), integral time squared error (ITSE) and integral squared error (ISE) performance indices are used as the objective function of the operation process in which the control parameters are determined. According to the comparative step response results of the controller structures, the four best controller structures for DC motor speed regulation are determined. The performances of these controllers are examined under different simulation conditions and according to the results obtained, it is seen that the best controller structure is FOPIDFF. The FDBAEO_OBL algorithm, which is used in both benchmark test functions and DC motor speed regulation, shows an effective, durable and superior performance in finding the optimal solution values during the optimization.

Keywords Artificial ecosystem optimization · Benchmark test functions · DC motor speed regulation · Fitness-distance balance guiding mechanism · Opposite-based learning methods

1 Introduction

DC motor is a type of motor that is frequently used in many applications due to its ease of use, low cost, and durability. It is preferred in robot arms in industrial applications, household electrical appliances, and electric vehicles in daily life. Depending on the requirements at the point of use of the DC motor, which converts electrical energy into motion energy, closed-loop speed control is widely used in motor control. It is expected that the motor speed settles to the determined reference value as quickly as possible, with minimum overshoot, and without error. To meet these conditions, the controller used must be well designed.

PID controller is widely used in closed-loop speed control used in motor speed regulation. It generates the required control signal to reduce the applied error signal to

✉ Evren Isen
eisen@bandirma.edu.tr

[1] Electrical Engineering, Engineering and Natural Sciences Faculty, Bandirma Onyedi Eylul University, Bandirma, Turkey

Published online: 12 July 2024

zero. The gain coefficient (K_p), integral coefficient (K_i) and derivative coefficient (K_d) in the PID controller, which consists of gain, integral and derivative operations, determine the performance of the controller. The K_p coefficient ensures that the output settles to the reference quickly. However, if it is not selected at the appropriate value, it causes an overshoot from the reference value. In addition, the K_p coefficient helps to reduce the steady-state error. The integrator in the controller ensures that the steady state error is reduced to zero by continuously adding the error. The negative effect of the integrator is that it can cause oscillation in the control signal when the sign of error signal changes. The derivative expression, which is the other part of the controller, calculates the slope of the error and provides to reduce settling time and the percentage of overshoot.

PID controller, which is used as a controller in many system controls such as voltage regulation (Hemeida et al. 2023), temperature control (Xu and Liu 2022), inverter control in grid-connected photovoltaic systems (Malarvili and Mageshwari 2022), DC–DC boost converter (Li et al. 2023), zeta converter (Arun and Manigandan 2021), is also widely used in motor applications (Baidya et al. 2023; Viaene et al. 2022; Jeon et al. 2018; Liu et al. 2021; Choi et al. 2015; Zhang and Gao 2022). A well-designed controller is required for the motor to achieve the desired reference value (speed, torque, and position) (Hasanhendoei et al. 2023; Amyal et al. 2023) and to maintain the speed value despite the disturbances. At this point, the controller coefficients gain importance. Various traditional methods such as Ziegler–Nichols (Ziegler and Nichols 1942), Cohen–Coon (Cohen and Coon 1953), Gain and Phase Method (Åstrom and Hägglund 1984) are used to determine the controller coefficients. Beetle antennae search algorithm (Mourtas et al. 2023), gazelle simplex optimizer (Ekinci et al. 2023a), hybrid Lévy flight distribution and Nelder–Mead algorithm (Izci 2021), slime mould algorithm (Izci and Ekinci 2021), particle swarm optimization (PSO) (Feng et al. 2021), ant colony optimization (Liang et al. 2019) and gray wolf optimization (Sahoo and Panda 2018) are also different optimization methods used lately. ITAE, ITSE, ISE and IAE are used as objective functions in determining the coefficients in optimization methods (Veinovic et al. 2022; Zeng et al. 2020; Bouakkaz et al. 2020; Isen 2022). PID controller has been widely used as a controller in many applications until recent years. Different controllers, though, have been developed and are being utilized to attain performance above that of this controller, nowadays. The most commonly used one is FOPID controller (Xia et al. 2023; Izci et al. 2022a). FOPIDF controller is obtained by adding N constant first order filter to the FOPID controller. Here the filter is used to eliminate noise and harmonics (Tripathy et al. 2021). In addition, TID controller (Ahmed et al. 2022) used in high frequency control is a fractional order controller type. TIDF controller, which has a similar structure to PIDF, has a fractional integrator and tilt coefficient instead of proportional coefficient. It provides simpler adjustment, a higher rate of interference prevention, and less sensitivity to changes in system parameters (Sahu et al. 2016). Unlike the PID controller, the PIDF controller has a derivative filter coefficient. This filter is used to reduce the derivative kick (Singh et al. 2021; Sahu et al. 2014). With the combination of PID with filter N and PD with filter N controllers, a dual-stage controller structure is created (Singh et al. 2023).

In recent years, advances in technology and rapid developments in computer architectures have increased the applicability of digital controller structures. Parameters of digital controllers determined according to user experience and classical calculation methods may not perform adequately in systems. The parameters of the controller structures are chosen using the optimization techniques recently presented in the literature to overcome these drawbacks resulting from user experience and conventional calculation methods. Zhao et al. (2020) recently introduced the artificial ecosystem optimization (AEO) method to the literature as a new population-based optimization algorithm that is motivated by the life process features of living species in the ecosystem. Since 2020, this method has been employed in more than 280 academic works, drawing the interest of several scientists from other disciplines. In these studies, it has been shown that the performance of the AEO algorithm is better than other optimization algorithms. Stated differently, its good exploration/exploitation capabilities in finding the global solution, its remarkable ability to converge to the optimum solution and its success in solving large-scale optimization problems make the AEO algorithm more valuable, important and preferable than other algorithms. Sonmez et al. (2022) developed the optimal solution and convergence capability of the AEO algorithm by using the selection method based on the fitness-distance balance (FDB) guiding mechanism. The authors named the algorithm they developed with the proposed FDB method as the FDB artificial ecosystem optimization (FDBAEO) algorithm.

In different studies in the literature, opposition-based learning (OBL) strategies have been used in the initial population to improve the performance of optimization algorithms and in local search operators that will enable the algorithms to avoid local solution traps. The FDB guidance mechanism plays an important role in identifying solution candidates that do not improve the search process in optimization algorithms and removing them from the solution space. Moreover, it reduces the risk of premature convergence, a phenomenon that can occur depending on the design steps of the algorithm. The FDB mechanism helps algorithms in avoiding local solution traps, encouraging an increase in the diversity of solutions within the solution space. Considering the advantages of both design operators presented in the literature to improve the performance of optimization algorithms, the FDBAEO algorithm, which includes AEO and OBL strategies containing these two design operators, has been presented to the literature for the first time. In this study, the use of OBL strategies in the initial population enabled the proposed algorithm to increase the solution diversity in the solution space and its ability to search for the optimal solution. Moreover, the FDB method is used to update the value of the solution candidate's position in the next iteration, thus reducing the risk of premature convergence of the algorithm. In other words, it enables the algorithm to get rid of local solution traps and contributed to its rapid convergence to the global solution point. In particular, the performances of both basic AEO and FDBAEO algorithms have been improved by using the advantages and superior features of these two design operators together. FDBAEO algorithm developed using OBL strategies is used to determine the parameters of different controller structures used in closed-loop speed control of the DC motor. In DC motor speed control, the two main criteria are the controller design and optimization algorithms used for adjusting the controller parameters to their optimum values. When the recent studies are examined, optimization algorithms are seen as one of the most effective system parameters in determining the parameters of the controller structures used in different fields of science and increasing the performance of the controllers. Eight different controller topologies with presented controller performance in various systems are used in this study to regulate the speed of a DC motor. While these controllers can improve system performance according to system operating conditions, controller parameters determined based on user experience or mathematical approaches may sometimes be insufficient to improve system performance. Considering this situation, the parameters of eight different controller structures used to improve the quality of the work and the performance of the controllers are adjusted using the proposed FDBAEO_OBL algorithm.

The specific main contributions of the study can be summarized as follows:

FDBAEO_OBL is introduced as a new, improved and powerful optimization algorithm. The performance of the proposed algorithm and its success in finding the optimal solution have been demonstrated by simulation studies on CEC 2020 benchmark test functions. In order for the FDBAEO algorithm developed using OBL methods to converge or find the optimal solution value in the benchmark test functions, the efficiency of the exploration and balanced search features used in the algorithm design has been demonstrated by testing in different dimensional search spaces and populations.

- The utilization of opposite-based learning (OBL) strategies in generating contrasting solutions within the initial population of the FDBAEO algorithm has enhanced its capability to converge toward optimal solutions. The incorporation of OBL strategies has improved the algorithm's ability to converge while also improving the balance between exploration and exploitation characteristics. The introduction of the FDBAEO_OBL algorithm was presented to the literature for the first time in this study.
- The results derived from non-parametric Wilcoxon and Friedman tests prove that the FDBAEO_OBL algorithm is an effective and robust algorithm in solving optimization problems in various disciplines.
- In this study the operational effectiveness of eight different controller structures—PID, FOPID, TID, TIDF, PIDF, FOPIDF, PIDFF, and FOPIDFF—whose parameters were optimized using the FDBAEO_OBL method, is thoroughly examined. Moreover, this study focuses on the speed regulation of a DC motor using optimized controller structures, and the results are thoroughly analyzed and discussed. This research concept is presented for the first time in this study for DC motor speed regulation.
- The comparative findings of the system's transient and dynamic responses across several simulation studies demonstrate the FOPIDFF controller's superior structure, durability, and effectiveness above other controller structures created with the proposed FDBAEO_OBL algorithm.

This study is presented in seven sections with subsections after the introduction. These sections are prepared as follows.

- The electrical and mechanical description of the DC motor structure is given in the second section.
- In the third section, the controller structures used to improve system performance are introduced.
- Section 4 and subsections describe the improved FDBAEO algorithm using four different OBL methods in detail.
- Information on the statistical evaluation of the results obtained from the standards, benchmark test functions and algorithms taken into account in the implementation of experimental studies is given in detail in Sect. 5.
- To highlight the effectiveness, efficiency, and robustness of the proposed FDBAEO OBL algorithm in obtaining the ideal solution, a thorough examination of the outcomes of two separate experimental experiments is presented in Sect. 6.
- The results are finally explained in Sect. 7 along with suggestions for additional research.

2 Definition of DC motor structure

The equivalent circuit model of the externally-excited type DC motor that is considered in this paper is seen in Fig. 1. It has an armature circuit and an exciting circuit as shown in the figure (Izci et al. 2021).

Equation (1) is obtained when the electrical equations of the armature circuit of the motor is written. In this equation, R_a and L_a is armature resistance and inductance, i_a is armature current, e_b is back-emf voltage, and e_a is armature voltage.

$$e_a(t) = R_a i(t) + L_a \frac{di(t)}{dt} + e_b(t) \tag{1}$$

The back-emf voltage is calculated depending on (2) which is obtained with K_b electromotor force constant and ω angular speed. Also, e_b changes with derivative of θ theta depending on the time as seen in Eq. (2).

$$e_b = K_b\omega = K_b \frac{d\theta}{dt} \tag{2}$$

The torque T that is given in (3) is the sum of inertia and friction torques. It is calculated with J inertia torque of motor, ω angular speed and B motor friction constant. The torques is also calculated depending on K motor torque constant and i_a armature current.

$$T = J\frac{d\omega}{dt} + B\omega = Ki_a \tag{3}$$

If the angle is used to calculated the torque, (4) can be used as seen below.

$$T = J\frac{d^2\theta}{dt^2} + B\omega \tag{4}$$

The open-loop transfer function of the motor that is derived from (1) to (3) is given in (5).

$$\frac{\omega(s)}{E_a(s)} = \frac{K}{JL_a s^2 + (R_a J + BL_a)s + (KK_b + R_a B)} \tag{5}$$

The block diagram of the transfer function is seen Fig. 2. The input of the model is the applied voltage to motor terminals whereas the output is speed of shaft. In this block diagram, T_L defines load torque, and it is assumed zero in the transfer function.

The values of motor parameters that are seen in (5) are given in Table 1. They are used in the simulation studies in the paper.

3 The controller structures used for improving the system performance

The closed-loop speed control of the system is seen in Fig. 3. The voltage required to reduce the error between the speed reference and the measured speed to zero is calculated by the controller. The calculated voltage E_a is applied to the motor model shown in Fig. 2 and the DC motor runs. Thus, the speed control of the motor is provided.

In this system, motor speed control is provided by the controller. Therefore, the performance of the controller is very important. Many structures are used as controllers in the literature. The block diagrams of the controllers examined in this study and used in the system are shown in Fig. 4.

The PID controller seen in Fig. 4a is the most frequently used controller from past to present (Ekinci et al. 2023b; Izci et al. 2023). The controller which mathematical expression is given in (6) consists of proportional, integral and derivative terms. Each has coefficient called K_p, K_i, and K_d that impacts the controller performance. These parameters determine the settling time, percentage of overshoot, and the steady-state error of the system at transition moments. As seen in Fig. 4a, the block diagram of the conventional PID controller is given in Fig. 4a. The error signal is given to the proportional, integral and derivative controller and the results are summed to obtain the PID output.

$$TF_{PID}(s) = K_p + \frac{K_i}{s} + K_d s \tag{6}$$

When the PID controller in Fig. 4a is added as a controller to the system structure seen in Fig. 3, the transfer function of the entire system is seen in (7). This 3rd order transfer function is used in system simulation and controller optimization.

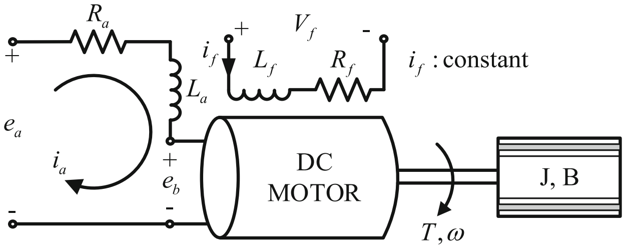


Fig. 1 DC motor model

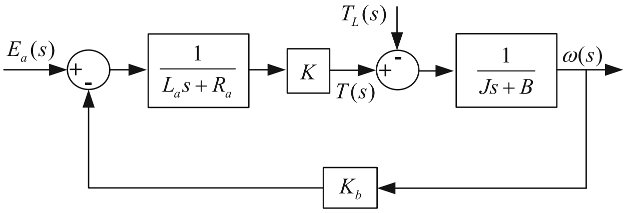


Fig. 2 DC motor block diagram

Table 1 Motor parameters (Sabir 2014)

Parameter	Value	Description
R_a	1 Ω	Armature resistance
L_a	0.5 H	Armature inductance
J	0.01 kg·m^2	Rotor inertia
B	3×10^{-5} N·m·s/rad	Viscous friction coefficient
K	23×10^{-3} kg·m/A	Torque constant
K_b	23×10^{-3} V·s	Electromotor force constant

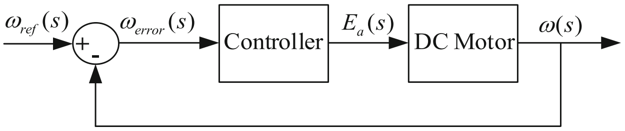


Fig. 3 Block diagram of DC motor control system

$$TF_{SYS}(s) = \frac{KK_d s^2 + KK_p s + KK_i}{s^3 JL + (JR + BL + KK_d)s^2 + (BR + K^2 + KK_p)s + KK_i} \tag{7}$$

Tilt integral derivative controller has three components as PID controller. The difference between the two controllers is that TID controller proportional term (K_t) is multiplied by a transfer function as seen in (8). The parameter of n is a real number different from zero (Topno and Chanana 2016; Rai and Das 2022).

$$TF_{TID}(s) = \frac{K_t}{s^{1/n}} + \frac{K_i}{s} + K_d s \tag{8}$$

The block diagram is seen in Fig. 4b. The input of the controller is an error signal and the output is the control signal. The transfer function depending on the motor and TID controller parameters is given in (9).

$$\begin{aligned} A_1 &= (\mathrm{JR} + \mathrm{BL} + \mathrm{KK}_d) \\ A_2 &= (\mathrm{BR} + \mathrm{KK}_d) \end{aligned}$$

$$\mathrm{TF}_{\mathrm{SYS}}(s) = \frac{\mathrm{KK}_d s^{2+(1/n)} + \mathrm{KK}_t s + \mathrm{KK}_i s^{1/n}}{s^{(3+(1/n))}\mathrm{JL} + A_1 s^{(2+(1/n))} + A_2 s^{(1+(1/n))} + \mathrm{KK}_t s + \mathrm{KK}_i s^{1/n}} \tag{9}$$

PI controller is preferred in many applications because of its simple structure and applicability. In order for the system to respond quickly in transient situations, the PID controller obtained by adding a derivative component to the PI controller is used. The derivative component, however, may produce unfavorable outcomes in practical applications if there is noise and a rapid change in the signal that the controller receives. The PIDF controller shown in (10) is produced for this situation by adding a filter to the derivative component. In the controller, which has a similar structure to the PID controller, N refers to the filter coefficient of derivative (Guha et al. 2017).

$$\mathrm{TF}_{\mathrm{PIDF}}(s) = K_p + \frac{K_i}{s} + K_d s \frac{N}{s + N} \tag{10}$$

The transfer function of the DC motor speed control system using the PIDF controller, which block diagram is seen in Fig. 4c, is given in (11).

$$\begin{aligned} B_1 &= (\mathrm{JR} + \mathrm{BL} + \mathrm{JLN}) \\ B_2 &= (\mathrm{BR} + K^2 + \mathrm{JRN} + \mathrm{BLN} + \mathrm{KK}_p + \mathrm{KK}_d N) \\ B_3 &= (\mathrm{BRN} + K^2 N + \mathrm{KK}_p N + \mathrm{KK}_i) \end{aligned}$$

$$\mathrm{TF}_{\mathrm{SYS}}(s) = \frac{(\mathrm{KK}_p + \mathrm{KK}_d N)s^2 + (\mathrm{KK}_p N + \mathrm{KK}_i)s + \mathrm{KK}_i N}{s^4 \mathrm{JL} + B_1 s^3 + B_2 s^2 + B_3 s + \mathrm{KK}_i N} \tag{11}$$

Tilt integral derivative with filter controller, which transfer function is seen in (12), is like the combination of PID and FOPID. It has the advantages of both controllers. The controller in the block diagram shown in Fig. 4d differs from the other two controllers in that it immediately suppresses the disturbances (Chiranjeevi et al. 2021).

$$TF_{TIDF}(s) = \frac{K_t}{s^{1/n}} + \frac{K_i}{s} + K_d s \frac{N}{s + N} \tag{12}$$

The transfer function of the system using the DC motor model and the controller model applied for speed control is seen in (13).

FOPID controller is an extended version of PID controller (Izci and Ekinci 2023). In the PID controller, the

$$\begin{aligned} C_1 &= (\mathrm{JR} + \mathrm{BL} + \mathrm{JLN}) \\ C_2 &= (\mathrm{BR} + \mathrm{KK}_b + \mathrm{JRN} + \mathrm{BLN} + \mathrm{KK}_d N) \\ C_3 &= (\mathrm{BRN} + \mathrm{KK}_b N + \mathrm{KK}_i) \end{aligned} \tag{13}$$

$$\mathrm{TF}_{\mathrm{SYS}} = \frac{\mathrm{KK}_d N s^{2+(1/n)} + \mathrm{KK}_t s^2 + \mathrm{KK}_i s^{1+(1/n)} + \mathrm{KK}_t N_s + \mathrm{KK}_i N s^{1/n}}{s^{4+(1/n)}\mathrm{JL} + C_1 s^{3+(1/n)} + C_2 s^{2+(1/n)} + \mathrm{KK}_t s^2 + C_3 s^{1+(1/n)} + \mathrm{KK}_t N s}$$

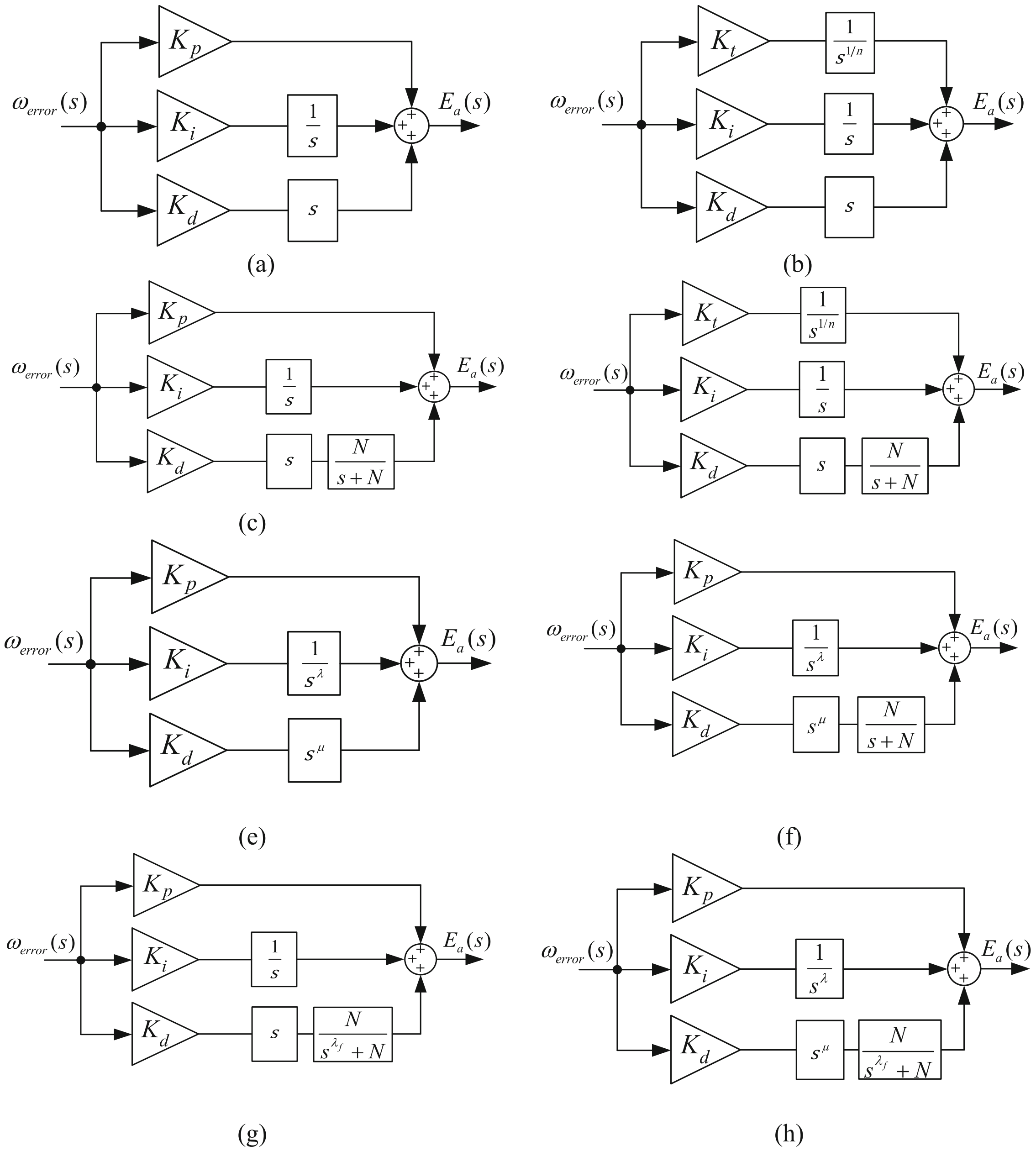


Fig. 4 Block diagrams of the controllers **a** PID, **b** TID, **c** PIDF, **d** TIDF, **e** FOPID, **f** FOPIDF, **g** PIDFF, **h** FOPIDFF

integrator and derivative components are of the 1st order. Controller performance is adjusted by coefficients only. In the FOPID controller, whose transfer function is given in (14) and the block diagram is given in Fig. 4e, λ and μ parameters also affect the controller performance. Depending on the 0 and 1 condition of these coefficients, the controller becomes P, PI, PD and PID controller (Divya et al. 2022; Tang et al. 2012).

$$\mathrm{TF}_{FOPID}(s) = K_p + \frac{K_i}{s^{\lambda}} + K_d\frac{s^{\lambda+\mu}}{s^{\lambda}} \tag{14}$$

The transfer function of the system obtained by using FOPID controller and DC motor model is given in (15). As can be seen, besides circuit parameters and controller coefficients, λ and μ parameters are included in the system transfer function and affect the performance.

$$D_1 = (\mathrm{JR} + \mathrm{BL})$$
$$D_2 = (\mathrm{BR} + K^2 + K_pK)$$
$$\mathrm{TF}_{\mathrm{SYS}} = \frac{\mathrm{KK}_d s^{\mu+\lambda} + K_pKs^{\lambda} + K_iK}{s^{2+\lambda}\mathrm{JL} + D_1 s^{1+\mu} + D_2 s^{\lambda} + K_dKs^{\lambda+\mu} + K_iK} \tag{15}$$

The FOPIDF controller is obtained by filtering the derivative component in the FOPID controller as seen in Fig. 4f. This feature improves the performance of the FOPID controller and filters out high-frequency noise in the control signal. The derivative filter not only reduces noise but also ensures that the transfer function is applicable (Dey et al. 2022). N, μ and λ parameters determine the filter performance in the FOPIDF controller, which transfer function is given in (16) and transfer function of the whole system to which it is applied in (17).

$$\mathrm{TF}_{FOPIDF}(s) = K_p + \frac{K_i}{s^{\lambda}} + K_d s^{\mu} \frac{N}{s+N} \tag{16}$$

$$\begin{aligned}
E_1 &= (\mathrm{JR} + \mathrm{BL} + \mathrm{JLN}) \\
E_2 &= (\mathrm{BR} + K^2 + K_p K + \mathrm{JRN} + \mathrm{BLN}) \\
E_3 &= (\mathrm{BRN} + K^2 N + \mathrm{KK}_p N) \\
&\mathrm{TF}_{\mathrm{SYS}}(s) \\
&= \frac{\mathrm{KK}_p s^{1+\lambda} + K_p \mathrm{KN} s^{\lambda} + K_i K s + \mathrm{KK}_i N + \mathrm{KK}_d N s^{\lambda+\mu}}{s^{3+\lambda}\mathrm{JL} + s^{2+\lambda} E_1 + E_2 s^{\lambda+1} + E_3 s^{\lambda} + \mathrm{KK}_d N s^{\lambda+\mu} + s\mathrm{KK}_i + \mathrm{KK}_i N}
\end{aligned} \tag{17}$$

The block diagram and transfer function of the PIDFF controller, which is obtained by adding a fractional order filter to the derivative component in the PID controller, is shown in Fig. 4g and (18). As you can see, unlike the PID controller, there is a filter with fractional order (λ_f) and gain (N). Apart from the coefficients in the classical PID controller, these parameters also need to be optimized (Sahin 2019).

$$\mathrm{TF}_{\mathrm{PIDFF}}(s) = K_p + \frac{K_i}{s} + K_d s \frac{N}{s^{\lambda_f} + N} \tag{18}$$

As seen in (19), a complex transfer function appears when the controller is used together with the system.

$$\begin{aligned}
F_1 &= (\mathrm{BR} + K^2 + \mathrm{KK}_p) \\
F_2 &= (\mathrm{JRN} + \mathrm{BLN} + \mathrm{KK}_d N) \\
F_3 &= (\mathrm{NBR} + K^2 N + \mathrm{KK}_p N) \\
F_4 &= (\mathrm{JR} + \mathrm{BL}) \\
&\mathrm{TF}_{\mathrm{SYS}}(s) \\
&= \frac{\mathrm{KK}_p s^{1+\lambda_f} + \mathrm{NK}_d s^2 K_t \mathrm{KK}_i s^{\lambda_f} + K_p \mathrm{KN} s + \mathrm{KK}_i N}{s^{3+\lambda_f}\mathrm{JL} + s^{2+\lambda_f} F_4 + s^{1+\lambda_f} F_1 + s^3 \mathrm{JLN} + s^2 F_2 + s F_3 + \mathrm{KK}_i s^{\lambda_f} + \mathrm{KK}_i N}
\end{aligned} \tag{19}$$

The fractional-order PID with fractional filter controller is the final controller and has the most complicated structure among all the controllers employed. Figure 4h shows its block diagram. There is a fractional-order integral and derivative component, just like in the FOPID controller, and a fractional-order filter in the derivative component, like in the PIDFF controller. As seen in (20), the number of parameters that need to be optimized is more than FOPID and PIDFF controllers, and it is 7 in total. The transfer function of the system using FOPIDFF is also given in (21) and it has a very complex structure.

$$TF_{FOPIDFF}(s) = K_p + \frac{K_i}{s^{\lambda}} + K_d s^{\mu} \frac{N}{s^{\lambda_f} + N} \tag{20}$$

$$\begin{aligned}
G_1 &= (jR + \mathrm{BL}) \\
G_2 &= (\mathrm{BR} + K^2 + \mathrm{KK}_p) \\
G_3 &= (\mathrm{JRN} + \mathrm{BLN}) \\
G_4 &= (\mathrm{BRN} + K^2 N + \mathrm{KK}_p N) \\
G_5 &= \mathrm{JLN} \\
G_6 &= \mathrm{KK}_d N \\
G_7 &= \mathrm{KK}_i N \\
G_8 &= \mathrm{KK}_i \\
&\mathrm{TF}_{\mathrm{SYS}}(s) \\
&= \frac{\mathrm{KK}_p s^{\lambda+\lambda_f} + \mathrm{KK}_p N s^{\lambda} + \mathrm{KK}_i N + \mathrm{KK}_d N s^{\lambda+\mu}}{s^{2+\lambda+\lambda_f} JL + G_1 s^{\lambda+\lambda_f+1} + G_2 s^{\lambda+\lambda_f} + G_5 s^{2+\lambda} + G_3 s^{\lambda+1} + G_6 s^{\lambda+\mu} + G_4 s^{2\lambda} + G_8 s^{\lambda_f} + G_7}
\end{aligned} \tag{21}$$

The optimizable parameters of the controllers and number of them are given in Table 2. FOPIDFF controller has maximum optimizable control parameters of 7, whereas PID has three parameters among the controllers studied in the paper.

Table 2 Optimizable parameters of the controllers

Controller	Optimizable Parameters	Number
PID	K_p, K_i, K_d	3
TID	K_t, K_i, K_d, n	4
PIDF	K_p, K_i, K_d, N	4
TIDF	K_t, K_i, K_d, n, N	5
FOPID	K_p, K_i, K_d, λ, μ	5
FOPIDF	K_p, K_i, K_d, λ, μ, N	6
PIDFF	K_p, K_i, K_d, λ_f, N	5
FOPIDFF	K_p, K_i, K_d, λ, λ_f, μ, N	7

4 Improved artificial ecosystem optimization algorithm with fitness-distance balance guiding mechanism by using opposite based learning methods

4.1 Artificial ecosystem optimization algorithm

The AEO algorithm was presented to the literature in 2020 by Zhao et al. (2020) as a new population-based meta-heuristic optimization algorithm inspired by the behavior of living organisms in the ecosystem, production, consumption, and decomposition characteristics. AEO has been used in nearly 200 scientific studies since 2020 to solve real-world engineering problems and different fields of science. While the production phase in the AEO algorithm aims at the discovery and usage trends of the optimization process, the consumption phase aims to develop the discovery feature of the algorithm in the optimization process. Decomposing is defined as a process used to improve the use process of the algorithm. AEO follows some rules stated below to search for the most appropriate solution in the solution process of an optimization problem.

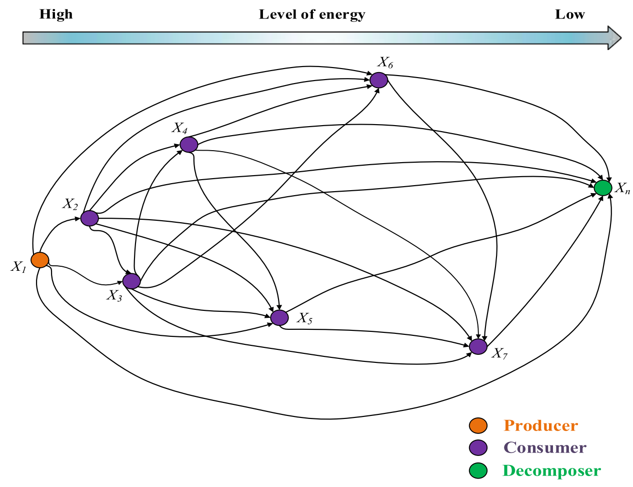


Fig. 5 The ecosystem of AEO algorithm

(1) The ecosystem including the producer, consumer, decomposer organisms should be defined as the population.
(2) It should contain only one producer and one decomposer in the population.
(3) Consumers selected as carnivores, herbivores, or omnivores should each have the same probability of generating other solution candidates in the population.

The ecosystem of the AEO algorithm is shown in Fig. 5. In the ecosystem constituting the population, the producer with the highest energy level X_1, which constitutes the population in the ecosystem is determined as the solution candidate and the decomposer with the lowest energy level X_n (best individual) is determined as the solution candidate. The producer with the highest energy level is determined as X_1 the solution candidate and the decomposer with the lowest energy level is determined as X_n (best individual) the solution candidate in the ecosystem that constitutes population Consumers representing equally distributed carnivorous, herbivorous, and omnivorous organisms for other solution candidates in the population are shown in detail in Fig. 5. The producer, consumer and decomposer processes of the algorithm are explained as follows (Zhao et al. 2020; Sonmez et al. 2022; Yousi et al. 2020; Rizk-Allah et al. 2021).

4.1.1 Production phase

In the optimization process of the AEO algorithm, balancing the discovery and usage trends of the algorithm to find the best value is expressed as the purpose of the production phase. The producer (the worst solution candidate) in the population is updated as in (22) using an individual (X_{rand}) that is constituted with usage of the best individual (X_n) and limit values of the variables in the population. The solution candidate, which is updated during the search process, directs the other solution candidates in the population to search in different regions of the solution space of the problem (Zhao et al. 2020; Sonmez et al. 2022; Topno and Chanana 2016; Rai and Das 2022; Guha et al. 2017; Chiranjeevi et al. 2021; Divya et al. 2022; Tang et al. 2012; Dey et al. 2022; Sahin 2019; Yousi et al. 2020; Rizk-Allah et al. 2021). The equations of σ ve X_{rand} parameters used in (22) are given in (23) and (24).

$$X_1(t+1) = (1-\sigma)X_n(t) + \sigma X_{rand}(t) \tag{22}$$

$$\sigma = (1 - \frac{t}{T_{\max}})r_1 \tag{23}$$

$$X_{rand} = r(U - L) + L \tag{24}$$

Here n is defined as the population size. σ is expressed as a linear weight coefficient used to move the individual to the best individual. t and T_{max} represent the current iteration and maximum iteration numbers, respectively. r_1 is a random number between [0,1]. r is a random number vector generated depending on the number of variables between [0,1]. U and L are specified as the upper and lower limits of the variables in the optimization problem.

4.1.2 Consumption phase

After the production process is completed in the ecosystem, the consumption operator is used in the algorithm for the consumers in the system. In the consumption operator, each consumer is fed by eating consumers with lower energy levels, producers, or both to obtain food energy and complete the energy flow in the ecosystem. This feeding process is simulated as the discovery process of the AEO algorithm. This process can be explained as (25) with a consumption factor with similar properties to Lévy Flight as a random walk without any parameters (Zhao et al. 2020; Sonmez et al. 2022; Topno and Chanana 2016; Rai and Das 2022; Guha et al. 2017; Chiranjeevi et al. 2021; Divya et al. 2022; Tang et al. 2012; Dey et al. 2022; Sahin 2019; Yousi et al. 2020; Rizk-Allah et al. 2021).

$$C = \frac{1}{2}\frac{v_1}{|v_2|} \quad v_1 \sim N(0,1), \quad v_2 \sim N(0,1) \tag{25}$$

Different types of consumers show different consumption behaviors in the consumption process. If the randomly selected individual from the population is herbivore, it feeds only on producer X_1 like consumers X_2 and X_5, as can be seen in Fig. 5. This process is expressed mathematically as in (26).

$$X_i(t+1) = X_i(t) + C(X_i(t) - X_1(t)) \ , \quad i \in \ [2, \ldots, n] \tag{26}$$

If the solution candidate selected from the population is carnivorous, it is fed with a consumer with a higher energy level than its own energy level, as seen in Fig. 5 (as in X_4 and X_6 consumers). The consumption behavior of the solution candidate carnivore can be defined by (27).

$$X_i(t+1) = X_i(t) + C\big(X_i(t) - X_j(t)\big) \ , \quad i \in \ [3, \ldots, n] \ , \ j = randi([2, i-1]) \tag{27}$$

If the i_{th} solution candidate selected from the population is omnivorous, it is fed by both the producer X_1 and another consumer with a higher energy level than its own. The consumption behavior of the omnivorous consumer is mathematically formulated as in (28).

$$\begin{cases} X_i(t+1) = X_i(t) + C(r_2 X_i(t) - X_1(t)) + (1 - r_2) \\ \qquad\qquad i \in \ [3, \ldots, n] \ , \ j = randi([2, i-1]) \end{cases} \tag{28}$$

Here, r_2 is defined as a random number between 0 and 1.

4.1.3 Decomposition phase

Decomposition can be expressed as the most important process in terms of the functioning of the ecosystem, where when an individual dies, its remains are decomposed by decomposers such as bacteria or fungi. This process is simulated mathematically with the parameters D (*dissociation factor*) and, e and h (*weighting coefficients*) in the algorithm, as in (29). In the parsing process, the position of the i_{th} solution candidate is updated via the position of the X_n parser. It also ensures that the next position of each solution candidate in the search space is located around the parser X_n (best individual), and this process is formulated as follows (Zhao et al. 2020; Sonmez et al. 2022; Topno and Chanana 2016; Rai and Das 2022; Guha et al. 2017; Chiranjeevi et al. 2021; Divya et al. 2022; Tang et al. 2012; Dey et al. 2022; Sahin 2019; Yousi et al. 2020; Rizk-Allah et al. 2021).

$$\begin{gathered} X_i(t+1)) = X_n(t) + D \cdot (e \cdot X_n(t) - h \cdot X_i(t)), \ \forall \ i \ \in \ n \\ D = 3u \ , \quad N(0,1) \ , \quad e = r_3 \cdot randi([12]) - 1 \ , \quad h = 2 \cdot r_3 - 1 \end{gathered} \tag{29}$$

4.2 AEO algorithm involving fitness-distance balance guiding mechanism

The FDBAEO algorithm, which includes a fitness-distance balance guiding mechanism, was presented to the literature by Sonmez et al (2022) in 2022. The performance of the algorithm has been increased by using the FDBAEO algorithm in the parsing process, which is one of the most important stages of the ecosystem. The FDB method aims to select the solution candidate from the population that can contribute the most to the search process after the best individual X_n used in the parsing process of the AEO. Thus, it was stated by the authors that the basic AEO algorithm provided a better balance between discovery and exploitation processes (Sonmez et al. 2022). The FDB method, which can be adapted to all optimization algorithms and creates solution diversity in the search space of algorithms, was first introduced to the literature in 2020 by Kahraman et al. (2020). This method has been adapted to optimization algorithms and has been used in the solution of different science fields and engineering problems in nearly 70 scientific studies since 2020. FDB has two important features that consider the fitness values of the solution candidates and the distances from the best solution candidate (P_{best}) in the population. The implementation steps in which the FDB guidance mechanism is introduced can be expressed as follows. (Aras et al. 2021; Guvenc et al. 2021; Duman et al. 2021, 2022; Kahraman et al. 2022; Bakir et al. 2022; Cengiz et al. 2021; Suicmez et al. 2021).

(i) i. In meta-heuristic optimization algorithms, n-unit solution candidates are created in the search space within the specified limit values of the control variables according to the determined population number (n). The vector (P) of the created solution candidates and their fitness values vector (F) can be expressed by the (30) below.

$$P = \begin{bmatrix} p_1 \\ \vdots \\ p_2 \end{bmatrix} = \begin{bmatrix} x_{11} & \cdots & x_{1m} \\ \vdots & \ddots & \vdots \\ x_{n1} & \cdots & x_{nm} \end{bmatrix}_{n \times m}, \quad F = \begin{bmatrix} f_1 \\ \vdots \\ f_n \end{bmatrix}_{n \times 1} \tag{30}$$

(ii) In a solution space with the number of solution candidates (n) and the number of variables (m) in the population, the Euclid distance of the i_{th} solution candidate p_i to the best solution candidate of the population (p_{best}) is formulated in (31).

$$\forall_{i=1}^{n} P_i, D_{p_i} = \sqrt{\left(p_{i[1]} - p_{best[1]}\right)^2 + \left(p_{i[2]} - p_{best[2]}\right)^2 + \ldots + \left(p_{i[m]} - p_{best[m]}\right)^2} \tag{31}$$

(iii) D_P is defined by (32) as a row vector containing numerical values representing the distance of solution candidates in the population from p_{best}.

$$D_p \equiv \begin{bmatrix} d_1 \\ \cdot \\ \cdot \\ \cdot \\ d_n \end{bmatrix}_{n \times 1} \tag{32}$$

(iv) While calculating the FDB score values of the solution candidates in the population, the fitness values vector (F) and the distance vector (D_p) are used. It is desired that these two parameters do not dominate each other in the calculation of the FDB value. Therefore, FDB values (S_p) of possible solution candidates are calculated as in (33) by normalizing F and D_p between [0,1].

$$\forall_{i=1}^{n} P_i, \quad S_{p_{[i]}} = w \cdot normF_{[i]} + (1 - w) \cdot normD_{p_{[i]}} \tag{33}$$

The weight coefficient w which is set to 0.5, is used to determine the effects of normalized fitness and distance values on the FDB score.

(i) The n-dimensional vector of FDB scores (S_p) in which the solution candidates are represented in the search space is shown in (34)

$$S_p \equiv \begin{bmatrix} s_1 \\ \cdot \\ \cdot \\ \cdot \\ s_n \end{bmatrix}_{n \times 1} \tag{34}$$

After the S_p vector showing the FDB scores of the possible solution candidates in the search space is created, the selection of individuals from the population can be done using the greedy or probabilistic selection method. The solution candidate with the highest FDB score value is used in the selection process.

The pseudocode of the steps of applying the FDB selection method in the algorithm is shown in Algorithm 1.

4.3 FDBAEO algorithm including opposite based learning methods

This section discusses the improvement of the FDBAEO algorithm using opposition-based learning (OBL) methods. Such approach strategies take into account the position that is opposed to the current solution candidate's position. By using the values of the fitness function of the solution candidates, it is determined whether the inverse is better than the current solution candidate and the best ones are selected for use in the next generations (Tizhoosh 2005; Elaziz et al. 2017; Elaziz and Mirjalili 2019; Ewees et al. 2018; Ekinci et al. 2022). OBL methods focus on increasing the rate of convergence to the optimal solution by improving the exploration capability of the FDBAEO algorithm. The OBL strategies used for the proposed algorithm in this study are described mathematically below.

i. In the method presented by Tizhoosh (2005) to the literature, $x \in [L, U]$ is defined as the inverse of the real number $\bar{x}$, as seen in (35).

$$\bar{x} = L + U - x \tag{35}$$

The definition of x in the multidimensional search space can be formulated with generalization as seen in (36) (Tizhoosh 2005).

$$\bar{x}_j = L_j + U_j - x_j \ , \quad x \in R^{nn} \ , \quad j = 1, 2, ..., nn \tag{36}$$

Here, j represents the number of variables in the solution candidate. The size of the solution candidate in the search space is defined as nn.

ii. Rahnamayan et al. (2007) presented the Quasi-OBL (QBL) method to the literature to improve the performance of the differential evolution algorithm with the help of the following mathematical equation. The quasi-opposite solution ($\overline{x}_q$) of solution candidate x is generated as a random solution candidate uniformly distributed between the midpoint (*mid*) and the opposite solution ($\overline{x}$) as shown in (37).

$$\overline{x}_q = mid + (mid - \overline{x}) \times rand \tag{37}$$

$$\begin{aligned} mid &= (U + L)/2 \\ \overline{x}_{qr} &= U + (mid - \overline{x}) \times rand \end{aligned} \tag{38}$$

iv. Kaucic (2013) provided the convergence of the particle swarm optimization algorithm to the global solution by using super-opposite based learning method in 2013. The performance of the algorithm proposed by the author has been verified by testing 100 global optimization problems.

Algorithm 1 The pseudo code of the AEO algorithm involving fitness-distance balance guiding mechanism

```
Start
    Generate the initial population (Ecosystem) by randomly generating $X_i$ solution candidates.
        for i=1:n do
            Compute the fitness function $fit_i$ and $X_{best}$
        end for
        while (Termination criterion not met do: up to maxFEs)
            Update the position of solution candidate $X_1$ using Equation (22) Production Phase
            for i=2:n do Consumption Phase
                If (rand<1/3) update the position of solution candidate $X_i$ using Equation (26) //(herbivore)
                Else If (1/3< rand<2/3) update the of solution candidate $X_i$ using Equation (27) //(carnivore)
                Else update the of solution candidate $X_i$ using Equation (28) //(omnivore)
                End if
            end for
            for i=1:n do
                Compute the fitness function $fit_i$ and update the $X_{best}$ so far
            end for
            for i=1:n do Decomposition Phase
                Compute distance of each solution candidate using Equation (31)
                Compute FDB score for each solution candidate using Equation (33)
            end for
            Create $D_p$ and $S_p$ vectors using Equation (32) and Equation (34)
            for i=1:n do
                Update the position of solution candidate $X_i$ using as follow Equation
                    $X_i(t+1) = X_{FDB}(t) + D(eX_n(t) - hX_i(t))$        $i = 1, \ldots, n$
                Compute the fitness function value $fit_i$
            end for
            Update the $X_{best}$ so far
        end while
    Show the best solution $X_{best}$
```

iii. Ergezer et al (2009) proposed the Quasi-Reflection OBL (QROBL) method to improve the convergence rate of the biogeography-based optimization algorithm. In this method, the quasi-reflected solution ($\overline{x}_{qr}$,) of the x solution candidate is mathematically defined as a uniformly distributed random solution as in (38).

Equation (39) represents the uniformly distributed super-opposite solution ($\overline{x}_{so}$) of the solution candidate x.

$$\overline{x}_{so} = \begin{cases} \overline{x} + (U - \overline{x}) \times \text{rand} & \text{if } \overline{x} > \text{mid} \\ L + (\overline{x} - L) \times \text{rand} & \text{otherwise} \end{cases} \tag{39}$$

The pseudocode of OBL methods showing the application steps in the FDBAEO algorithm is shown in Algorithm 2.

Algorithm 2 The pseudo code of the FDBAEO algorithm including OBL strategies

```
Start
  Generate the initial population (Ecosystem) by randomly generating $X_i$ solution candidates.
  Create an oppositional population according to OBL strategies $(\tilde{x}_j, \tilde{x}_q, \tilde{x}_{qr}, \tilde{x}_{so})$
  Select OBL strategy
      for i=1:n do
          Compute the fitness function $fit_i$
          Compute the fitness function $\widetilde{fit}_i$ according to the selected OBL strategy
          if $\widetilde{fit}_i < fit_i$
          $fit_i = \widetilde{fit}_i$;
          $X_i$ = The solution candidate of the selected OBL strategy $(\tilde{x}_j, \tilde{x}_q, \tilde{x}_{qr}, \tilde{x}_{so})$
          else
          $fit_i = fit_{i;}$
          $X_i = X_{i;}$
          end if
          Compute the $X_{best}$
      end for
      while (Termination criterion not met do: up to maxFEs)
          Update the position of solution candidate $X_1$ using Equation (22) Production Phase
          for i=2:n do Consumption Phase
              If (rand<1/3) update the position of solution candidate $X_i$ using Equation (26)
          //(herbivore)
              Else If (1/3< rand<2/3) update the of solution candidate $X_i$ using Equation (27)
          //(carnivore)
              Else update the of solution candidate $X_i$ using Equation (28) //(omnivore)
              End if
          end for
          for i=1:n do
              Compute the fitness function $fit_i$ and update the $X_{best}$ so far
          end for
          for i=1:n do Decomposition Phase
              Compute distance of each solution candidate using Equation (31)
              Compute FDB score for each solution candidate using Equation (33)
          end for
          Create $D_p$ and $S_p$ vectors using Equation (32) and Equation (34)
          for i=1:n do
              Update the position of solution candidate $X_i$ using as follow Equation
                  $X_i(t+1) = X_{FDB}(t) + D(eX_n(t) - hX_i(t)) \qquad i = 1, \ldots, n$
              Compute the fitness function value $fit_i$
          end for
          Update the $X_{best}$ so far
      end while
    Show the best solution $X_{best}$
```

5 Experimental settings of the study

An experimental study is conducted to verify the success of the OBL-based FDBAEO algorithm, which is developed for the speed control of the DC motor from electrical machines, in convergence and finding the optimal solution. The FDBAEO algorithm based on OBL methods is defined as follows.

- FDBAEO_OBL1: FDBAEO algorithm based on the classical OBL method
- FDBAEO_OBL2: FDBAEO algorithm based on Quasi-OBL method
- FDBAEO_OBL3: FDBAEO algorithm based on Quasi-Reflection OBL method
- FDBAEO_OBL4: FDBAEO algorithm based on Super OBL method

Comparison of the performances of FDBAEO algorithms based on the above mentioned OBL methods, classical AEO and FDBAEO algorithms is performed under the conditions described below.

(a) CEC2020 benchmark test is used to test the global solution search performance of FDBAEO algorithms based on AEO, FDBAEO and OBL methods.

(b) 10000*D maximum function evaluations (maxFEs) are used as termination criteria in the search space. Thus, a fair comparison environment is created for the algorithms to search for the global solution during the optimization.

(c) To evaluate the convergence performance of the algorithms to the general solution in different search spaces, experimental studies are carried out in the

Benchmark test system for the dimensions of 5/10/15/20/30/50/100.

(d) 30 independent trials are conducted for each test problem in the CEC2020 benchmark test system. Statistical analysis studies are carried out to make the results obtained from these experimental studies more meaningful. In these studies, non-parametric Wilcoxon pairwise and Friedman statistical test methods are used.

The performances of FDBAEO algorithms, classical AEO and FDBAEO algorithms based on the proposed OBL methods are described as follows in two different experimental studies.

(i) First subsection: the successful performances of the proposed method and other algorithms are investigated in the CEC2020 (Yue et al. 2019) benchmark test functions.

(ii) The parameters of the controllers used in the speed control of the DC motor are optimized by using the most successful algorithm among the FDBAEO algorithms based on OBL methods. Thus, it is aimed to design the most suitable controller for the speed control problem of the DC motor.

6 Simulation studies

6.1 Identifying the best FDBAEO algorithm based on OBL methods in CEC 2020 Benchmark Test Functions

CEC 2020 benchmark test functions are used to test the success and ability of the FDBAEO algorithm designed according to four different OBL strategies in finding the global solution point in these test functions. In order to make a fair comparison between the proposed approach, AEO and FDBAEO algorithms, the adjustment parameters of the algorithms determined according to the user experience are accepted as presented in the literature in the basic AEO algorithm and simulation studies are carried out.

6.1.1 Statistical analysis

In the statistical analysis subsection, the Friedman method is used to statistically analyze the results obtained from the experimental studies of the FDBAEO_OBL, AEO and FDBAEO algorithms. By means of this non-parametric method, the algorithm that successfully finds the global solution point is determined. The problem size in each of the benchmark test functions in CEC 2020 has been determined as 5/10/15/20/30/50 and 100. In addition, two different case studies are determined as population numbers of 100 and 200. The Friedman results and the success ranking of the algorithms according to the statistical analysis of the results obtained from the simulation studies on these problem sizes and population numbers are presented in detail in Tables 3 and 4, respectively.

The Friedman values from 14 different trials are presented in Table 3. According to these values, different population numbers and mean values of Friedman results for each algorithm are demonstrated in Table 4. The last row in Table 4 shows the Friedman overall mean values and the ranking of the algorithms based on these values. Under the simulation conditions determined for the CEC 2020 benchmark test functions, it is clearly seen that the best performing algorithm compared to its competitors in finding the global solution is FDBAEO_OBL3 (FDBAEO algorithm based on Quasi-Reflection OBL method). Also, the algorithm with the worst performance is the FDBAEO algorithm. In addition, according to the results in Table 4, it is seen that each of the FDBAEO algorithms designed according to the OBL strategies is better than the AEO and FDBAEO algorithms.

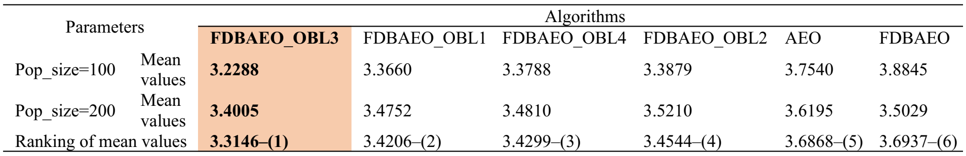
The results obtained from the CEC2020 benchmark test functions are used in the Wilcoxon pairwise test to compare the classical AEO algorithm with other algorithms and the Wilcoxon results of the algorithms against the AEO algorithm are given in Table 5.

At the end of the optimization process, in order for the Wilcoxon binary test results to be statistically meaningful, the victory of the algorithm against the AEO algorithm is defined with the " + " symbol, the draw with the " = " symbol, and the defeat with the "–" symbol. The FDBAEO_OBL3 algorithm, which ranked first among the algorithms according to the Friedman test results, is evaluated for the situation where the population number is 100 in the CEC 2020 benchmark test functions. According to the evaluation results, FDBAEO_OBL3 performed 4/8/7/1/1 times better than the classical AEO algorithm in 5/10/15/20/50 problem sizes. The proposed approach and the AEO algorithm in these problem sizes have obtained similar results 2/2/2/9/9 times. In the 30- and 100-dimensional search space, the proposed approach and the AEO algorithm converged to a similar result 10/10 times. For the 5/10/15/20/30/50/100-dimensional search space, when the population number is 200, the FDBAEO_OBL3 algorithm victories the AEO algorithm 1/4/4/6/5/4/4 times. Moreover, the proposed approach is drawn 3/5/3/1/3/5/2 times and defeated 6/1/3/3/2/1/4 times with the AEO algorithm. In other words, the FDBAEO_OBL3 algorithm has been compared with the AEO algorithm 140 times in the simulation studies conducted with different populations and sizes, and according to the results of the comparison, 49

Table 3 Freidman test results of the optimization algorithms

Algorithms	CEC 2020						
	Pop_size = 100						
	D = 5	D = 10	D = 15	D = 20	D = 30	D = 50	D = 100
AEO	3.39500	4.99667	4.54000	3.29667	3.34000	3.34000	3.37000
FDBAEO	3.60167	3.14667	3.16667	4.74667	4.45333	4.23000	3.84667
FDBAEO_OBL1	3.62833	3.10667	3.29000	3.22667	3.46667	3.43333	3.41000
FDBAEO_OBL2	3.56833	3.20000	3.36333	3.30333	3.24333	3.50667	3.53000
FDBAEO_OBL3	3.22500	3.16333	3.31667	3.10667	3.13667	3.18667	3.46667
FDBAEO_OBL4	3.58167	3.38667	3.32333	3.32000	3.36000	3.30333	3.37667
Algorithms	CEC 2020						
	Pop_size = 200						
	D = 5	D = 10	D = 15	D = 20	D = 30	D = 50	D = 100
AEO	2.31000	3.83000	3.45000	4.16333	4.10667	4.02333	3.45333
FDBAEO	3.77333	3.56667	3.59000	3.28667	3.41000	3.40000	3.49333
FDBAEO_OBL1	3.54333	3.57333	3.54667	3.42000	3.35333	3.39667	3.49333
FDBAEO_OBL2	3.82333	3.32333	3.53333	3.33333	3.51667	3.45333	3.66333
FDBAEO_OBL3	3.60667	3.37000	3.39667	3.36000	3.29000	3.33333	3.44667
FDBAEO_OBL4	3.94333	3.33667	3.48333	3.43667	3.32333	3.39333	3.45000

Table 4 Friedman test ranking of the optimization algorithms



Parameters		Algorithms					
		FDBAEO_OBL3	FDBAEO_OBL1	FDBAEO_OBL4	FDBAEO_OBL2	AEO	FDBAEO
Pop_size=100	Mean values	**3.2288**	3.3660	3.3788	3.3879	3.7540	3.8845
Pop_size=200	Mean values	**3.4005**	3.4752	3.4810	3.5210	3.6195	3.5029
Ranking of mean values		**3.3146–(1)**	3.4206–(2)	3.4299–(3)	3.4544–(4)	3.6868–(5)	3.6937–(6)

The color was used to show the best results

wins, 66 draws and 25 losses have been achieved. To better understand and evaluate these results, the percentage comparison of wins, draws and defeats in test functions against the AEO algorithm of the proposed approach is given in Table 6. According to the results in Table 6, where wins, draws and defeats are evaluated as percentages, it is clearly seen that the FDBAEO_OBL3 algorithm is superior to the AEO algorithm in 9 of 14 different simulation studies.

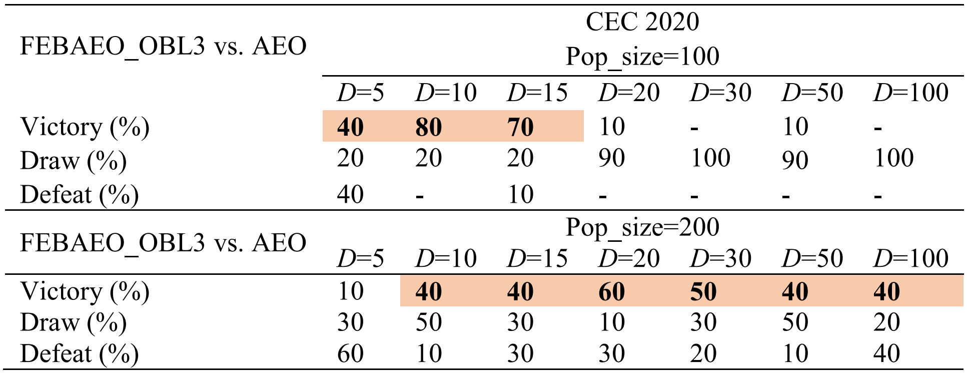
6.1.2 Convergence analysis

This subsection presents the convergence curves and boxplot figures of the OBL variants of the AEO, FDBAEO and FDBAEO algorithms. Convergence and boxplot analysis aims to better understand and interpret optimization algorithms for search and convergence to the global solution point in different problem types. Convergence curves and boxplot graphs in cases where the population numbers of test problems selected from the CEC 2020 benchmark test functions are 100 and 200 are shown in Figs. 6 and 7, respectively. Figures 6 and Fig. 7a, d and g show the variation of the error value obtained depending on the convergence of the algorithms to the optimal solution value according to the maximum fitness function evaluation criterion. The graphs showing the change in convergence values of the best fitness values obtained depending on the iteration are shown in Figs. 6 and 7b, e and h. The performance of all optimization algorithms is tested by performing 30 independent trials for each simulation case. Boxplot plots are given in Figs. 6 and 7c, f and i for better interpretation of the results from these independent runs. Boxplot plots show the limits of the best fitness values that the optimization algorithms found in independent trials. In other words, it gives information about the frequency of convergence of the algorithm to the optimal solution after

Table 5 Wilcoxon pairwise comparison results for the optimization algorithms

vs. AEO + / = /-	CEC 2020						
	Pop_size = 100						
	D = 5	*D* = 10	*D* = 15	*D* = 20	*D* = 30	*D* = 50	*D* = 100
FDBAEO	4/2/4	8/2/0	7/2/1	1/2/7	0/5/5	1/5/4	3/2/5
FDBAEO_OBL1	4/2/4	9/1/0	7/2/1	0/10/0	0/10/0	0/10/0	0/10/0
FDBAEO_OBL2	4/2/4	9/1/0	7/2/1	0/10/0	0/10/0	0/9/1	0/10/0
FDBAEO_OBL3	4/2/4	8/2/0	7/2/1	1/9/0	0/10/0	1/9/0	0/10/0
FDBAEO_OBL4	4/2/4	9/1/0	7/2/1	1/9/0	0/10/0	1/9/0	1/9/0
vs. AEO + / = /–	Pop_size = 200						
	D = 5	*D* = 10	*D* = 15	*D* = 20	*D* = 30	*D* = 50	*D* = 100
FDBAEO	1/3/6	2/7/1	4/3/3	6/1/3	4/4/2	4/5/1	4/2/4
FDBAEO_OBL1	1/3/6	3/6/1	4/2/4	6/1/3	4/4/2	4/5/1	3/3/4
FDBAEO_OBL2	1/3/6	4/5/1	4/2/4	6/1/3	4/4/2	4/5/1	4/2/4
FDBAEO_OBL3	1/3/6	4/5/1	4/3/3	6/1/3	5/3/2	4/5/1	4/2/4
FDBAEO_OBL4	1/3/6	4/5/1	5/1/4	6/2/2	4/4/2	4/5/1	4/2/4

Table 6 Comparison of FDBAEO_OBL3 algorithm against AEO algorithm as a percentage of victory, draw and defeat



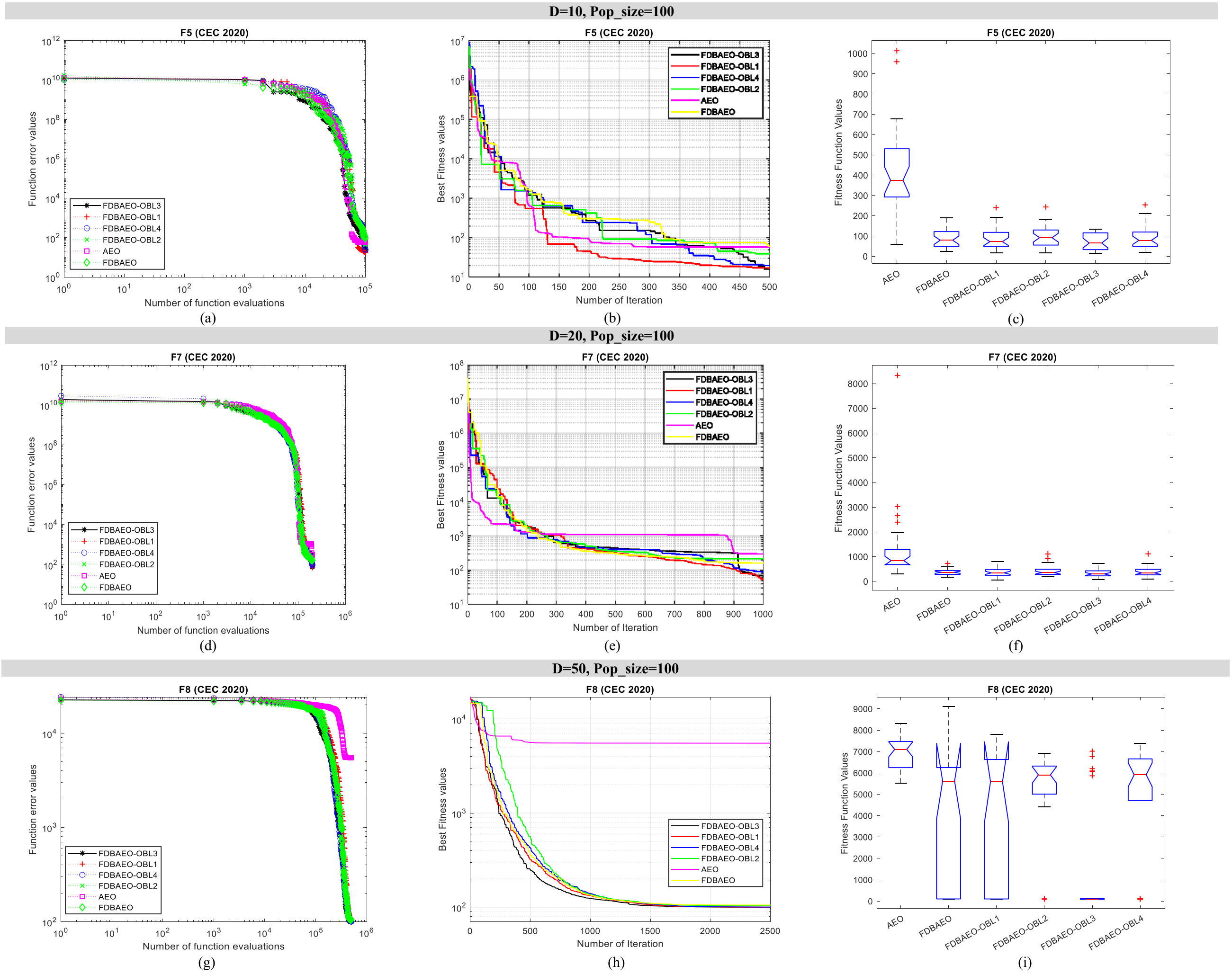
FEBAEO_OBL3 vs. AEO	CEC 2020						
	Pop_size=100						
	D=5	*D*=10	*D*=15	*D*=20	*D*=30	*D*=50	*D*=100
Victory (%)	**40**	**80**	**70**	10	-	10	-
Draw (%)	20	20	20	90	100	90	100
Defeat (%)	40	-	10	-	-	-	-
FEBAEO_OBL3 vs. AEO	Pop_size=200						
	D=5	*D*=10	*D*=15	*D*=20	*D*=30	*D*=50	*D*=100
Victory (%)	10	**40**	**40**	**60**	**50**	**40**	**40**
Draw (%)	30	50	30	10	30	50	20
Defeat (%)	60	10	30	30	20	10	40

The color was used to show the best results

30 trials. It is clear from the boxplot graphs that depending on the problem type, the proposed FDBAEO_OBL3 algorithm always converges to the optimal solution value of the problem or closer value to optimal solution. Moreover, these graphs confirm the results of both Friedman and Wilcoxon non-parametric statistical analysis.

6.2 Application of the FDBAEO_OBL3 algorithm in controller design of DC motor

The speed control of the DC motor is realized in MATLAB/Simulink environment with the developed FDBAEO_OBL3 algorithm. Figure 8 displays the flow diagram or block diagram illustrating the optimization procedure whereby the controller parameters utilized for DC motor speed control are optimized via the use of the proposed algorithm. The speed error between the reference speed and the measured actual speed in the control structure is computed and provided to the controller whose parameters are optimized, as can be seen in the block diagram in Fig. 8. The speed control of the motor is achieved by applying the signal that makes the error produced by the controller to zero to the DC motor. PID, FOPID, TID, TIDF, PIDF, FOPIDF, PIDFF, and FOPIDFF controllers are used to provide speed regulation of the DC motor. The optimization algorithm's objective functions are created based on the performance indices. The optimized controller parameters are represented in the



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

Fig. 6 Convergence and boxplot shapes of algorithms for Pop_size = 100

optimization process by the solution candidate with the lowest objective function value. The DC motor speed regulation circuit tests these parameters and the outcomes are examined.

6.2.1 Using performance indices as objective functions during optimization

The performance indices that are commonly used in the literature are IAE (Ghith and Tolba 2023), ISE (Izci et al. 2022a), ITAE (Majhi et al. 2021) and ITSE (Barakat et al. 2022). The equations of the indices that are used as the objective functions are presented in Eqs.(40)–(43). They are used in the study to evaluate the performance of the controllers that are optimized.

$$IAE = \int_0^T |e(t)|dt \tag{40}$$

$$ISE = \int_0^T e^2(t)dt \tag{41}$$

$$ITAE = \int_0^T t|e(t)|dt \tag{42}$$

$$ITSE = \int_0^T te^2(t)dt \tag{43}$$

6.2.2 Evaluation of simulation results obtained according to performance indices

In Table 7, rise time, settling time, overshoot, peak value, peak time and steady-state error values of the controllers optimized with the FDBAEO_OBL3 method based on IAE, ITAE, ITSE and ISE performance indices are shown.

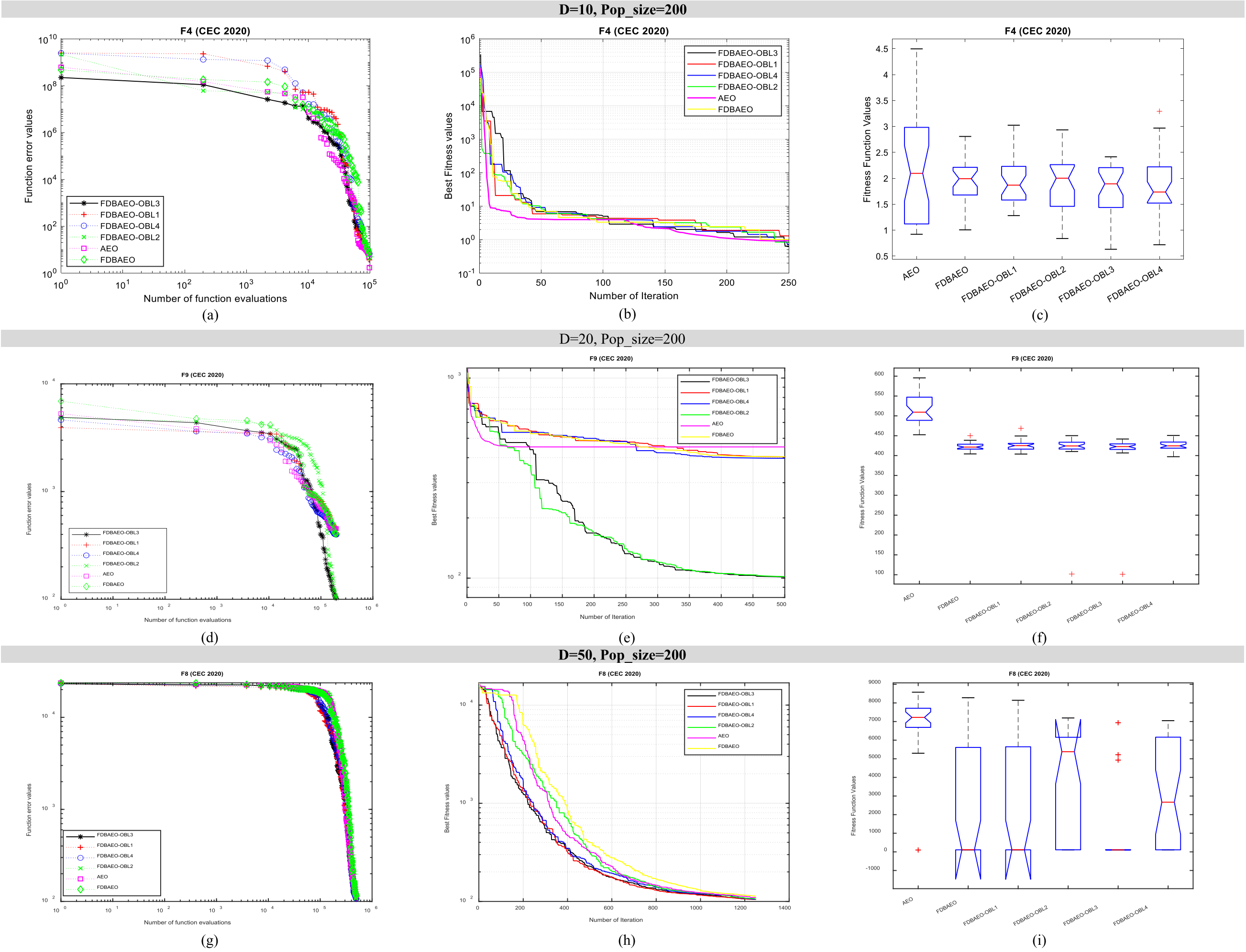


Fig. 7 Convergence and boxplot shapes of algorithms for Pop_size = 100

As a result of these data, the best performance indices of each controller in different performance parameters are determined and highlighted. In the PID controller, all the best results are obtained with the ITSE index, while in the FOPID controller, the best rise time is obtained with the IAE index and the other best results are obtained with the ITSE index. As seen in the table, these determinations are made for all controller and performance indices.

Table 8 is created by ranking 1–4 from the best to the worst in the performance results seen in Table 7. The best result is represented by 1, while 4 is used for the worst result. Then, the scores of the parameters are averaged and the average values are obtained. Based on these values, the average value ranking is made. Thus, the performance index with the best score is determined for each controller. According to this method, as seen in Table 8, ITAE for PID controller, ITSE for FOPID, ISE for TID, IAE for TIDF, ITAE for PIDF, IAE for FOPID, ITAE for PIDFF and IAE for FOPIDFF are the best performance indices. They are highlighted in the table.

Table 9 is obtained depending on Table 8. The best performance index for each controller is determined based on "Ranking of mean values" in Table 8, and the best transient response values are given in Table 9 for each controller. In addition, the best controller for each transient response parameter is determined by scoring the controllers. The best controller for transient response is selected by averaging the scores of the controllers. As seen in Table 9, the first and best performance controller is FOPIDFF. The best four controllers that are FOPIDFF, FOPID, FOPIDF and PID between eight controllers are determined depending on the Table 9 and they are used for the performance analysis in the rest of the work. The step performance of the controllers is seen Fig. 9. As seen in Fig. 9a, all controllers settle to the reference value with successful steady-state response. The maximum overshoot occurs in FOPIDF controller while the minimum overshoot occurs in FOPID controller. The minimum settling time and rise time belongs to FOPIDFF and FOPIDF, respectively. The results can be seen in Table 9 and Fig. 9b.

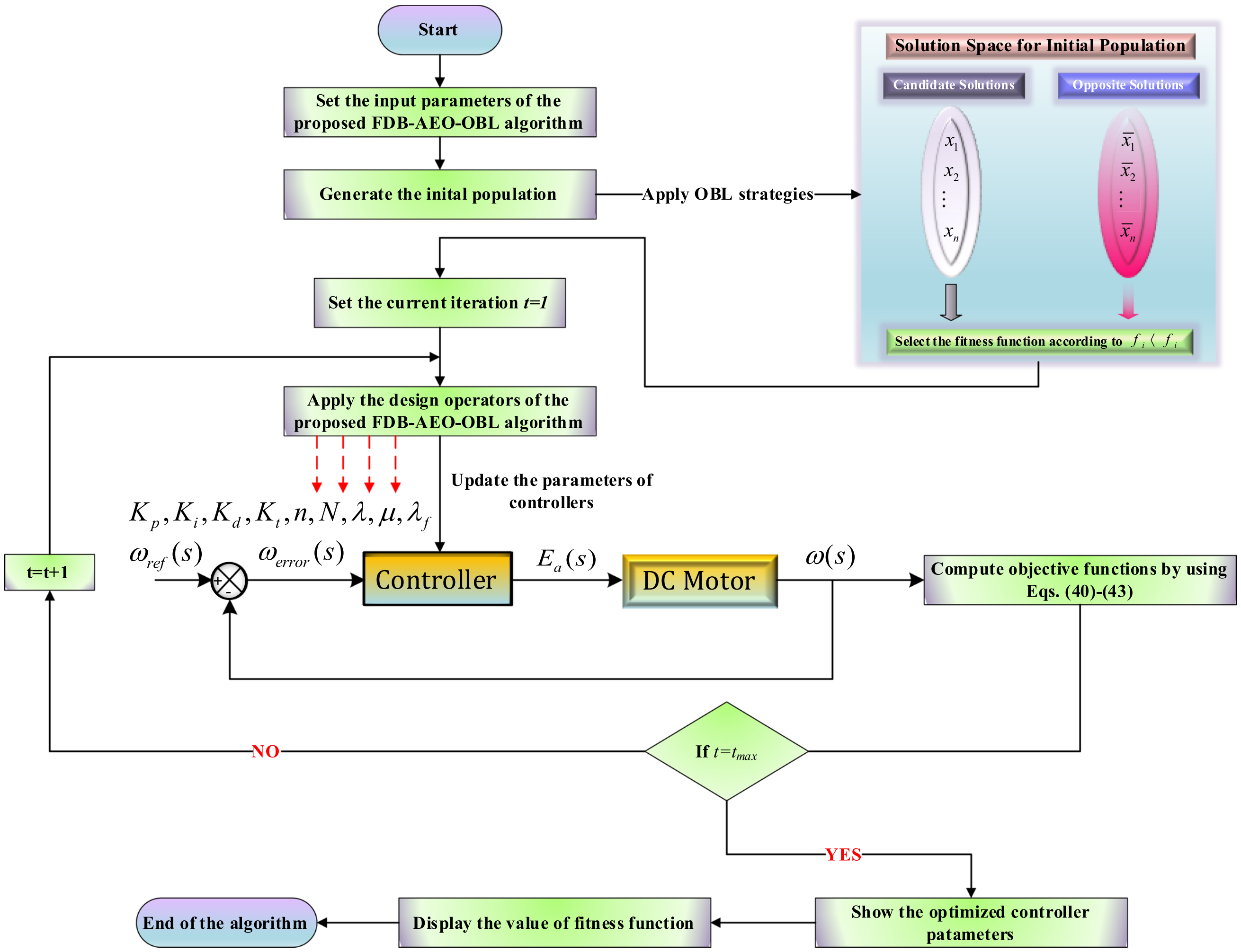

Start
Set the input parameters of the proposed FDB-AEO-OBL algorithm
Generate the inital population
Apply OBL strategies
Solution Space for Initial Population
Candidate Solutions
Opposite Solutions
x_1
x_2
x_n
$\bar{x}_1$
$\bar{x}_2$
$\bar{x}_n$
Select the fitness function according to $f_i < f_i$
Set the current iteration $t=1$
Apply the design operators of the proposed FDB-AEO-OBL algorithm
Update the parameters of controllers
$K_p, K_i, K_d, K_t, n, N, \lambda, \mu, \lambda_f$
$\omega_{ref}(s)$
$\omega_{error}(s)$
+
-
Controller
$E_a(s)$
DC Motor
$\omega(s)$
Compute objective functions by using Eqs. (40)-(43)
t=t+1
NO
If $t=t_{max}$
YES
Show the optimized controller patameters
Display the value of fitness function
End of the algorithm

Fig. 8 Flowchart of the optimization process of the proposed FDB-AEO-OBL algorithm

FOPID and FOPIDFF controllers have almost the same response, but FOPIDFF has little higher performance.

In this study, eight distinct controller structures with different design parameters were used to ensure speed regulation of the DC motor and to contribute the motor operate more steadily under changing operating conditions. Stated differently, appropriate determination of the parameters in a controller structure ensures that the systems can operate stable and reliably. In light of this circumstance, Table 10 provides a detailed display of the parameters that were acquired by applying the proposed FDBAEO-OBL technique to identify the most suitable controller settings. Additionally, Table 10 displays the optimal parameter values of the top four controller architectures, based on the outcomes of several objective functions. Four controller structures that differ from each other in terms of computational complexity demonstrated more effective performance than TID, TIDF, PIDF and PIDFF controllers in providing speed regulation of the DC motor. Determining the parameters of a controller structure using optimization algorithms affects the calculation time of the optimization algorithm during the optimization process. The quantity of parameters in the controller structures that need to be optimized is the cause of this. The computational complexity of controller structures with fewer parameters is lower than that of controller structures with more parameters when controller structures with optimized parameters are assessed in terms of application. Nonetheless, the sampling frequency needs to be determined in accordance with the controller structure with the highest computational complexity if it is intended to test the performance of controllers in systems equitably.

To examine the dynamic response of the controllers depending on the speed reference change of the motor, five different speed refences are applied to the controllers in different time intervals that are given in Table 11. The start-up speed reference is set to 1, and it increases to 1.2 and 1.5 in the next two intervals. In the last two intervals, the speed reference decreases to 0.6 and 0.8.

Table 7 Performance criteria of the controllers FDBAEO_OBL3

Controller	Transient responses	Performance Indices				Controller	Transient responses	Performance Indices			
		IAE	ITAE	ITSE	ISE			IAE	ITAE	ITSE	ISE
PID	Rise time (s)	0.064	0.0606	0.0663	0.0704	PIDF	Rise time (s)	0.0526	0.0513	0.0551	0.0573
	Settling time (s)	0.462	0.318	0.492	0.675		Settling time (s)	0.498	0.361	0.674	1.16
	Overshoot (%)	23	18.7	26.6	35.7		Overshoot (%)	46.9	40.3	54.5	65.2
	Peak value (V)	1.23	1.19	1.27	1.36		Peak value (V)	1.47	1.4	1.54	1.65
	Peak time (s)	0.161	0.153	0.166	0.177		Peak time (s)	0.143	0.137	0.15	0.162
	SS_Error (V)	0	0	0	0		SS_Error (V)	0	0	0	0
FOPID	Rise time (s)	0.00536	0.0402	0.0241	0.0155	FOPIDF	Rise time (s)	0.0147	0.0335	0.0153	0.00626
	Settling time (s)	0.0794	0.189	0.0503	0.143		Settling time (s)	0.0848	0.165	0.083	-
	Overshoot (%)	-	12.5	-	0.268		Overshoot (%)	21.6	24.5	19.4	32.9
	Peak value (V)	0.995	1.13	0.993	1.00268		Peak value (V)	1.22	1.25	1.19	1.33
	Peak time (s)	0.225	0.0939	0.078	0.547		Peak time (s)	0.0358	0.0854	0.0367	0.0166
	SS_Error (V)	0.005	0	0	0		SS_Error (V)	0	0.01	0	0.02
TID	Rise time (s)	0.07666	07674	0.07675	0.07617	PIDFF	Rise time (s)	0.0518	0.0502	0.054	0.0567
	Settling time (s)	0.566	0.566	0.566	0.566		Settling time (s)	0.95	0.441	1.52	3.14
	Overshoot (%)	12	12	12	12		Overshoot (%)	63.3	44.8	69.8	77.9
	Peak value (V)	1.12	1.12	1.12	1.12		Peak value (V)	1.63	1.45	1.7	1.78
	Peak time (s)	0.258	0.259	0.259	0.258		Peak time (s)	0.146	0.134	0.156	0.16
	SS_Error (V)	0	0	0	0		SS_Error (V)	0	0	0	0
TIDF	Rise time (s)	0.0742	0.0766	0.0769	0.076	FOPIDFF	Rise time (s)	0.0224	0.0369	0.0243	0.00434
	Settling time (s)	1.93	2.22	3.09	3.74		Settling time (s)	0.0416	0.167	0.043	0.146
	Overshoot (%)	72.9	75.5	79.8	83.7		Overshoot (%)	0.661	4.62	0.734	2.85
	Peak value (V)	1.73	1.75	1.8	1.84		Peak value (V)	1.00661	1.05	1.00734	1.03
	Peak time (s)	0.206	0.215	0.218	0.218		Peak time (s)	0.07	0.0898	0.0713	0.00876
	SS_Error (V)	0	0	0	0		SS_Error (V)	0.003	0.01	0.00734	0

The color was used to show the best results

Figure 10 shows the dynamic response of the controllers for five different speed references. All the controllers have good steady-state response by settling the reference value. The dynamic response of the controllers in Region 1–Region 5 is shown in detail in Fig. 10b–f. As can be seen from the figures, the FOPID and FOPIDFF controller shows good performance in terms of settling time, overshoot and steady state response. The controllers have similar responses as in Fig. 9. A Bode diagram of the controllers can be seen in Fig. 11.

Four different scenarios are created to examine how the system responded at different motor parameters. In the scenarios created, the motor armature resistance is 0.75 and 1.25, and the torque constant is 0.0184 and 0.0276. The scenarios created using these values are given in Table 12.

When Scenario I in Table 12 is applied, the change in the response of the system can be seen in Fig. 12. When compared with the response obtained from the basic motor parameters used in the study, it is seen that there is a decrease in the overshoot value of the FOPIDF controller. The time to reach steady state of the PID controller has increased, however, the controller is successful in the steady state response. The numerical expressions of the response performances of the controllers are shown in Table 13.

In the second scenario, the moment constant is increased by keeping the armature resistance constant. In this case, the step responses of the controllers and the transient response at the beginning are seen in Fig. 13. The numerical values of the response performances of the controllers in these changes are given in Table 14. As can be seen from the response curves, the overshoot of the PID controller decreases while the overshoot of the FOPIDF controller increases. When examined in terms of steady-state response, the PID controller enabled the system to settle in a shorter time. Other transient response performance values are given in Table 13. When examined in terms of steady state, all controllers ensured that the system is set to the reference value despite the parameter change.

The third scenario is created by increasing the motor armature resistance and using the torque constant at the value used in Scenario I. The performance data of the controllers of Scenario III, whose step response is seen in Fig. 14, are given in Table 15. While the PID controller overshoot performance is the same as in Scenario II, the FOPIDF overshoot value is the least compared to Scenario I and Scenario II. FOPIDFF controller response reached the reference value in the shortest time. In the steady state response, all controllers reached the reference value. Performance values are given in detail in Table 15.

In Scenario IV, which is the last scenario, the moment constant is increased while the armature resistance is kept constant. When the response changes and performance values seen in Fig. 15 and Table 16 are examined, the PID controller result improved in terms of overshoot and the FOPIDF controller response worsened. While FOPIDFF gives the best result in terms of settling time, the time is shortened compared to Scenario III. In the steady state response, all the controllers ensured that the system is set to the reference value.

Table 8 Performance scoring of the controllers

Controller	Transient responses	Performance Indices				Controller	Transient responses	Performance Indices			
		IAE	ITAE	ITSE	ISE			IAE	ITAE	ITSE	ISE
PID	Rise time (s)	2	1	3	4	PIDF	Rise time (s)	2	1	3	4
	Settling time (s)	2	1	3	4		Settling time (s)	2	1	3	4
	Overshoot (%)	2	1	3	4		Overshoot (%)	2	1	3	4
	Peak time (s)	2	1	3	4		Peak time (s)	2	1	3	4
	Mean values	2	1	3	4		Mean values	2	1	3	4
	Ranking of mean values	2	1	3	4		Ranking of mean values	2	1	3	4
FOPID	Rise time (s)	1	4	3	2	FOPIDF	Rise time (s)	2	4	3	1
	Settling time (s)	2	4	1	3		Settling time (s)	2	3	1	4
	Overshoot (%)	1	3	1	2		Overshoot (%)	2	3	1	4
	Peak time (s)	3	2	1	4		Peak time (s)	2	4	3	1
	Mean values	1.75	3.25	1.5	2.75		Mean values	2	3.5	2	2.5
	Ranking of mean values	2	4	1	3		Ranking of mean values	1	3	1	2
TID	Rise time (s)	2	3	4	1	PIDFF	Rise time (s)	2	1	3	4
	Settling time (s)	1	1	1	1		Settling time (s)	2	1	3	4
	Overshoot (%)	1	1	1	1		Overshoot (%)	2	1	3	4
	Peak time (s)	1	2	2	1		Peak time (s)	2	1	3	4
	Mean values	1.25	1.75	2	1		Mean values	2	1	3	4
	Ranking of mean values	2	3	4	1		Ranking of mean values	2	1	3	4
TIDF	Rise time (s)	1	3	4	2	FOPIDFF	Rise time (s)	2	4	3	1
	Settling time (s)	1	2	3	4		Settling time (s)	1	4	2	3
	Overshoot (%)	1	2	3	4		Overshoot (%)	1	4	2	3
	Peak time (s)	1	2	3	3		Peak time (s)	2	4	3	1
	Mean values	1	2.25	3.25	3.25		Mean values	1.5	4	2.5	2
	Ranking of mean values	1	2	3	3		Ranking of mean values	1	4	3	2

The color was used to show the best results

Table 9 The best controllers depending on the performance scoring in Table 8

Transient responses	Controllers							
	PID	FOPID	TID	TIDF	PIDF	FOPIDF	PIDFF	FOPIDFF
Rise time (s)	0.0606	0.0241	0.07617	0.0742	0.0513	0.0147	0.0502	0.0224
Settling time (s)	0.318	0.0503	0.566	1.93	0.361	0.0848	0.441	0.0416
Overshoot (%)	18.7	0	12	72.9	40.3	21.6	44.8	0.661
Peak value (V)	1.19	0.993	1.12	1.73	1.4	1.22	1.45	1.00661
Peak time (s)	0.153	0.078	0.258	0.206	0.137	0.0358	0.134	0.07
SS_Error (V)	0	0	0	0	0	0	0	0.003
Ranking of transient responses of the controllers								
Transient responses	PID	FOPID	TID	TIDF	PIDF	FOPIDF	PIDFF	FOPIDFF
Rise time (s)	6	3	8	7	5	1	4	**2**
Settling time (s)	4	2	7	8	5	3	6	**1**
Overshoot (%)	4	1	3	8	6	5	7	**2**
Peak time (s)	6	3	8	7	5	1	4	**2**
Mean values	5	2.25	6.5	7.5	5.25	2.5	5.25	**1.75**
Ranking of mean values	4	2	6	7	5	3	5	**1**

The color was used to show the best results

6.2.3 Computational complexity of the proposed algorithm

Computational complexity in optimization algorithms tells researchers how long it takes for an algorithm to solve a given problem; this time varies depending on the program's design parameters. In the literature, computational complexity is expressed in a variety of ways. To help readers better comprehend computational complexity, the methodology used in related articles (Ekinci et al. 2022; Izci et al. 2023) is applied to the discussion of computational complexity in this study. The computational complexity of the baseline AEO and intended FDBAEO-OBL algorithms is calculated considering three main criteria. These can be expressed as initializing the solution space,

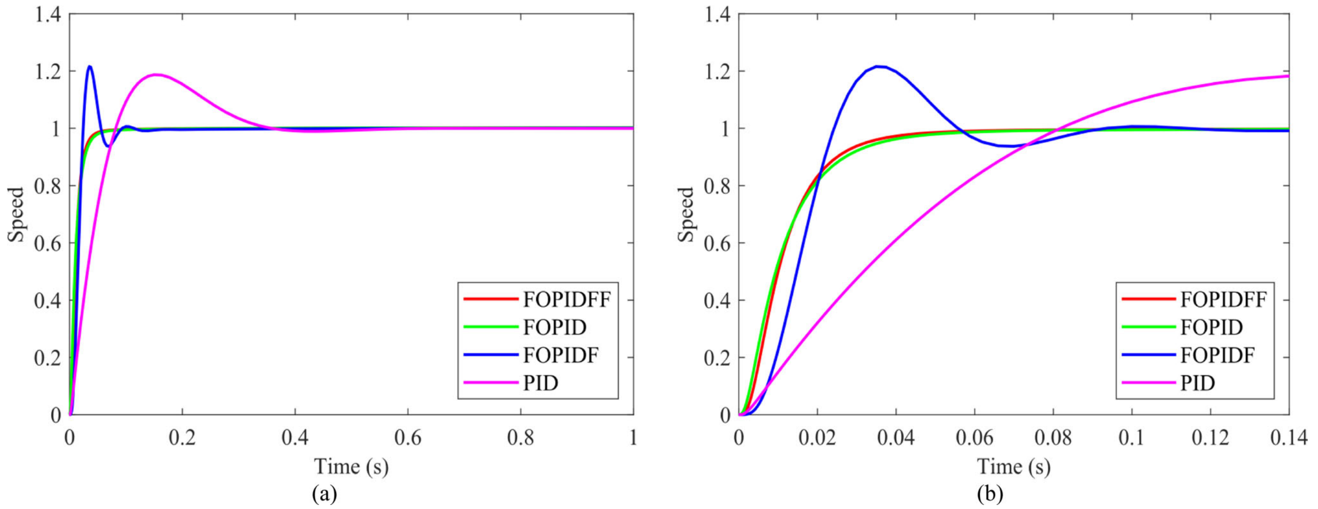


Fig. 9 Step response of the controllers **a** Step response, **b** Transient response at start

Table 10 Optimized parameter values of the four best controllers

Parameter	PID	FOPID	FOPIDF	FOPIDFF
K_p	49.9854	30.7432	26.4429	25.1659
K_i	0.3459	9.3411	9.1942	9.8338
K_d	3.9757	16.0164	18.4174	16.2685
λ	–	1.0011	0.4823	0.1537
μ	–	1.0203	1.0206	1.0203
N	–	–	121.7686	146.8065
λ_f	–	–	1	0.2517

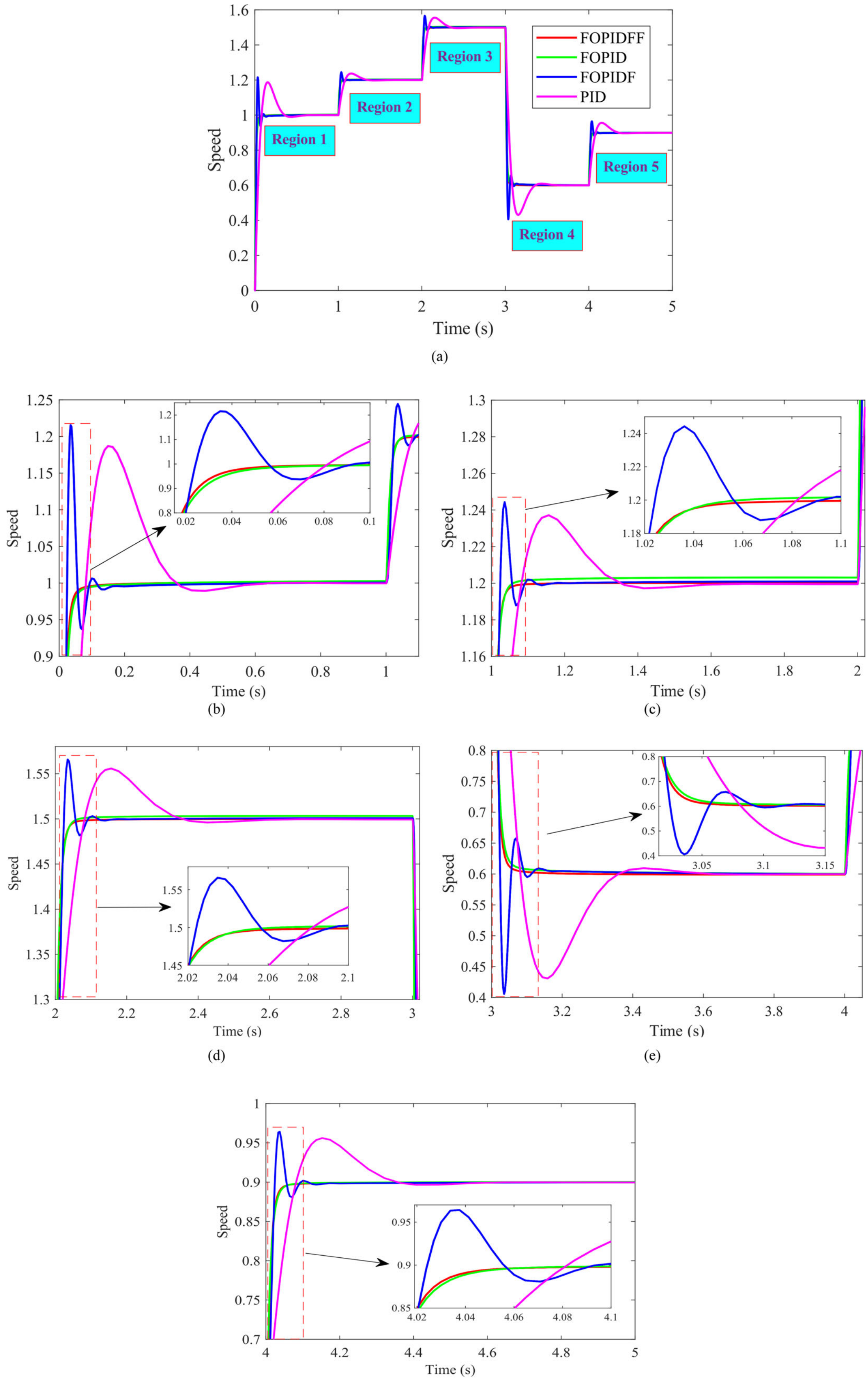
Table 11 Five different speed reference values in different time intervals

Time	Speed reference
$0 \leq t < 1$	1
$1 \leq t < 2$	1.2
$2 \leq t < 3$	1.5
$3 \leq t < 4$	0.6
$4 \leq t$	0.8

calculating the fitness function of each solution candidate in the solution space depending on the design steps, and updating the positions of the solution candidates. The computational complexity of the AEO algorithm can be computed using these three criteria. CC(AEO) = CC(N) + CC($N \times t_{\max}$)- + CC($N \times d \times t_{\max}$), where, N, $t_{\max}$, and d can be expressed as the size of the solution space, the maximum number of iterations, and the dimension of the problem, respectively. However, the computational complexity of the proposed FDBAEO_OBL algorithm can be given as CC(FDBAEO_OBL) = CC(AEO) + CC($N \times t_1$)- + CC($2 \times N \times t_{\max}$). CC($N \times t_1$) shows the effect of OBL strategies used in the initial population on the computational complexity. CC($2 \times N \times t_{\max}$) represents the additional computational load brought about by the FDB guiding mechanism. The proposed FDBAEO_OBL algorithm appears to have an additional computational overhead, both because it is used in the initial population of OBL strategies and because the FDB guidance mechanism is used in updating solution candidates. Although this may seem to be a disadvantage of the algorithm in terms of calculation time, improving the ability to find the optimal solution, quickly converge to the optimal solution, and avoid local solution traps are expressed as the biggest advantages of the algorithm.

7 Conclusion

This study aims to improve the performance of the FDBAEO algorithm by using classical OBL, Quasi-OBL, Quasi-Reflection OBL and Super OBL methods. These four different OBL methods are used to increase the solution diversity to find the global point of the solution space created in the initial population of the optimization algorithm and their performance is tested in different simulation studies. In the first simulation study, variations of the FDBAEO algorithm based on OBL methods are investigated in different dimensional search spaces and population numbers in terms of their ability to converge to the optimal solution, exploration, exploitation, and avoidance of problem-based local solution traps, using the CEC2020 benchmark test functions. According to the experimental results obtained from different simulation scenarios, non-parametric Wilcoxon and Friedman tests are applied to statistically evaluate the optimal solution search performance of the algorithm. According to statistical


FOPIDFF
FOPID
FOPIDF
PID
Region 1
Region 2
Region 3
Region 4
Region 5
Speed
Time (s)

(a)

(b)

(c)

(d)

(e)

(f)

◀**Fig. 10** Dynamic response of the controllers **a** normal version, **b** zoom of region 1, **c** zoom of region 2, **d** zoom of region 3, **e** zoom of region 4, **f** zoom of region 5

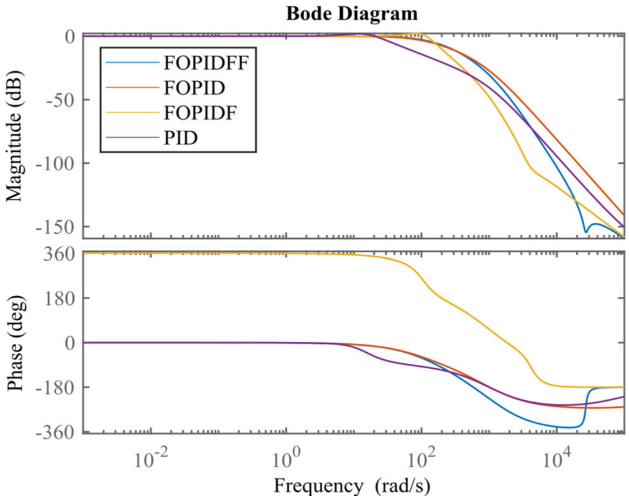


Fig. 11 Bode plots of the controllers

results, FDBAEO algorithm based on Quasi-Reflection OBL (FDBAEO_OBL3) method is more successful than FDBAEO algorithms with other OBL versions, basic AEO and FDBAEO algorithms.

Table 13 The performances of the controller in Scenario I

Transient responses	Controllers			
	PID	FOPID	FOPIDF	FOPIDFF
Rise time (s)	0.0695	0.0308	0.0178	0.0279
Settling time (s)	0.489	0.142	0.0903	0.0535
Overshoot (%)	22.3	0.0236	14.9	0.107
Peak value (V)	1.223	1.000236	1.149	1.00107
Peak time (s)	0.177	0.142	0.0413	0.0954

The proposed FDBAEO_OBL3 method is used in the second simulation study to set the controller parameters used for speed regulation of DC motors which is electrical machine used in many electrical engineering applications. The parameters of eight different controller structures are determined by the proposed optimization algorithm and their performances are tested in different simulation scenarios. Depending on the rise time, settling time, overshoot and peak time values, the performances of the controllers are evaluated with statistical scoring values according to the results obtained from different objective functions. Based on these statistical scoring results, the four best controller structures for DC motor speed regulation are determined. To evaluate the performance of these

Table 12 Five different speed reference values in different time intervals

Motor parameter	Scenario I	Scenario II	Scenario III	Scenario IV
R_a [Ω]	0.75	0.75	1.25	1.25
K [kg·m/A]	0.0184	0.0276	0.0184	0.0276

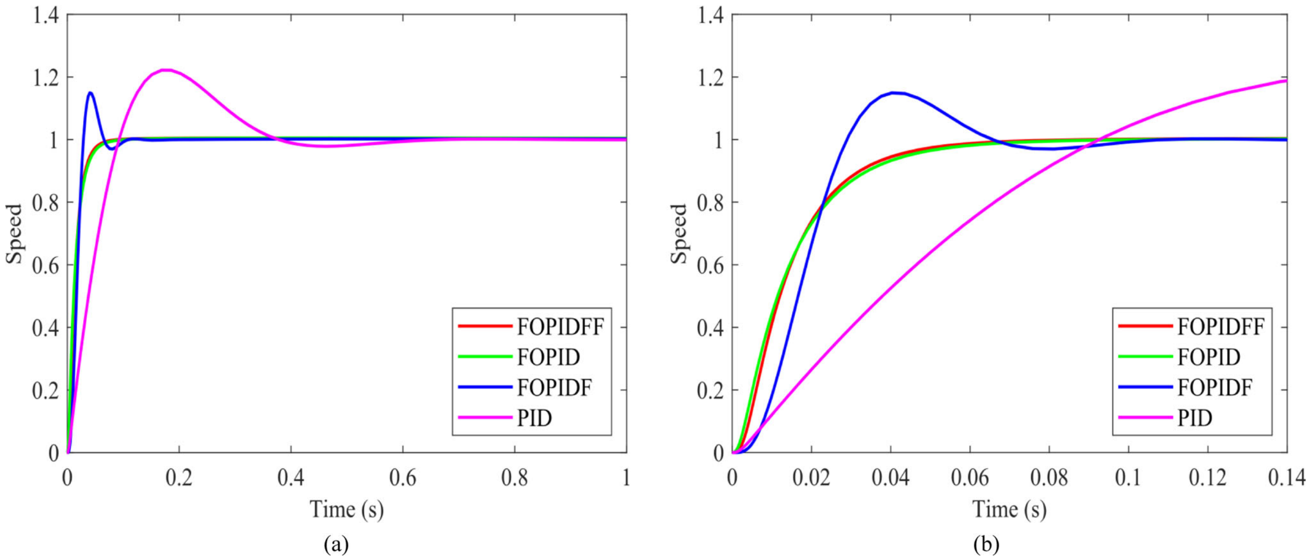


Fig. 12 Step response of the controller for Scenario I **a** Step response, **b** Transient response at start

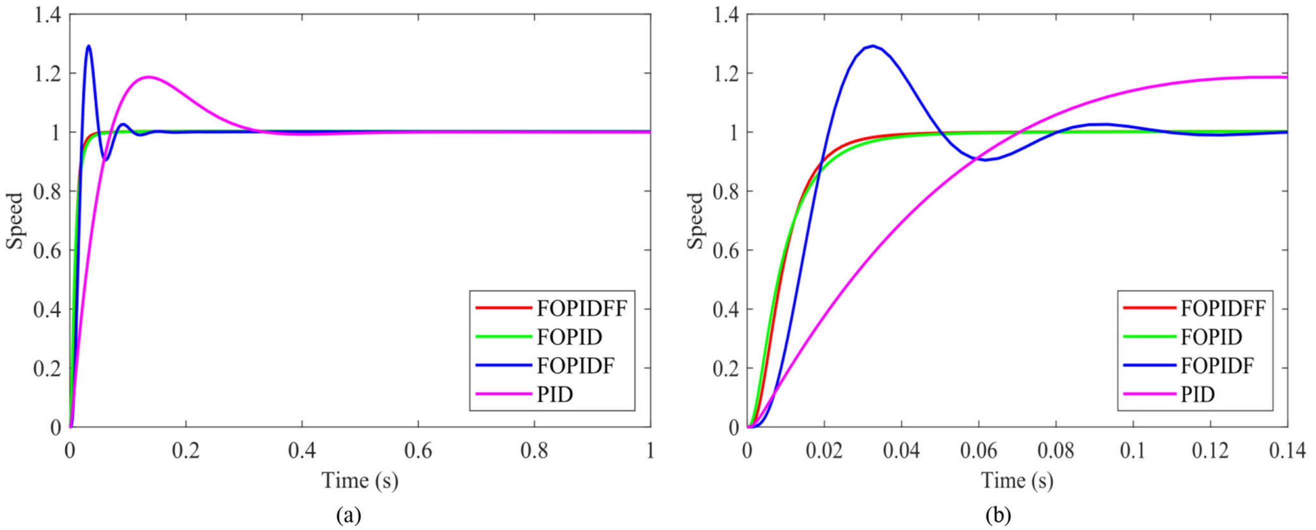


Fig. 13 Step response of the controller for Scenario II **a** Step response, **b** Transient response at start

Table 14 The performances of the controller in Scenario II

Transient responses	Controllers			
	PID	FOPID	FOPIDF	FOPIDFF
Rise time (s)	0.0519	0.0188	0.0126	0.0159
Settling time (s)	0.292	0.0372	0.0975	0.0314
Overshoot (%)	18.6	0	29.2	0
Peak value (V)	1.18	1	1.29	1
Peak time (s)	0.135	0.07	0.0323	0.06

Table 15 The performances of the controller in Scenario III

Transient responses	Controllers			
	PID	FOPID	FOPIDF	FOPIDFF
Rise time (s)	0.0728	0.0325	0.0181	0.0294
Settling time (s)	0.356	0.0775	0.0989	0.0689
Overshoot (%)	18.7	–	13.3	–
Peak value (V)	1.187	0.989	1.13	0.991
Peak time (s)	0.183	0.14	0.0411	0.15

controllers in detail, simulation studies are carried out under different scenarios. According to the comparative results obtained from the simulation studies, it is clearly seen that the FOPIDFF controller, whose parameters are adjusted using the proposed FDBAEO_OBL3 method, exhibits a more successful and superior performance than

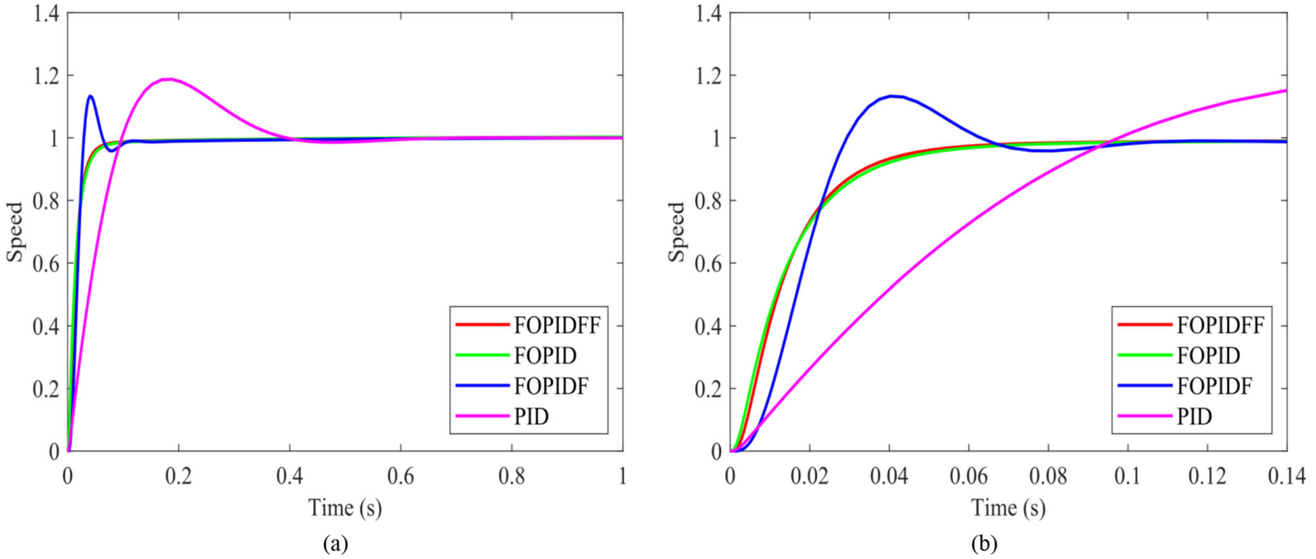


Fig. 14 Step response of the controller for Scenario III **a** Step response, **b** Transient response at start

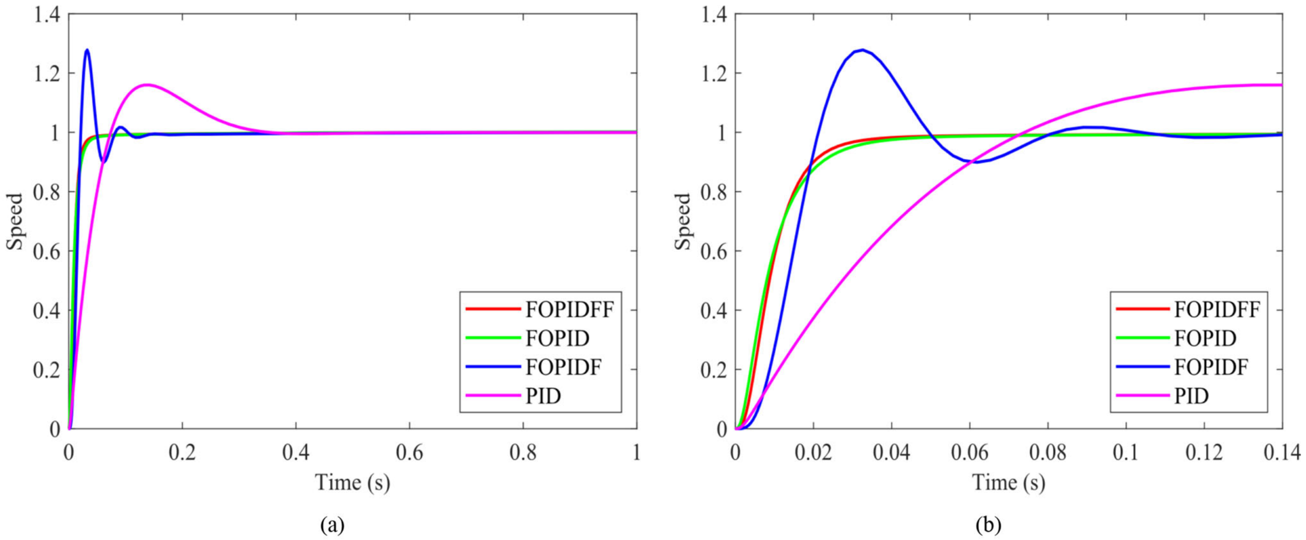


Fig. 15 Step response of the controller for Scenario IV **a** Step response, **b** Transient response at start

Table 16 The performances of the controller in Scenario IV

Transient responses	Controllers			
	PID	FOPID	FOPIDF	FOPIDFF
Rise time (s)	0.054	0.0195	0.0127	0.0164
Settling time (s)	0.295	0.0445	0.0777	0.037
Overshoot (%)	16	–	27.7	–
Peak value (V)	1.16	0.989	1.27	0.989
Peak time (s)	0.137	0.07	0.0322	0.06

other controller structures. All the simulation studies show that the controller coefficients that provides the desired high performance in DC motor speed control can be determined with the optimization method.

The FDBAEO_OBL3 method can be used in solving optimization problems in different fields of science and especially in solving power systems planning and operation problems such as optimal power flow, economic dispatch, combined heat and power economic dispatch, energy management of micro-grid and transmission expansion planning.

Acknowledgments This research was funded by The Scientific and Technological Research Council of Türkiye-Turkish Academic Network and Information Center.

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

Data availability There is no additional data associated with this manuscript.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

Ethical approval This article does not contain any studies with human participants or animals performed by any of the authors.

References

Ahmed M, Magdy G, Khamies M, Kamel S (2022) Modified TID controller for load frequency control of a two-area interconnected diverse-unit power system. Int J Electr Power Energy Syst 135(107528):1–20

Amyal FA, Szamel L, Hamouda M (2023) An enhanced direct instantaneous torque control of switched reluctance motor drives using ant colony optimization. Ain Shams Eng J 14(5):11–15

Aras S, Gedikli E, Kahraman HT (2021) A novel stochastic fractal search algorithm with fitness-distance balance for global numerical optimization. Swarm Evol Comput 61(100821):1–15

Arun S, Manigandan T (2021) Design of ACO based PID controller for zeta converter using reduced order methodology. Microprocess Microsyst 81(103629):1–11

Åström KJ, Hägglund T (1984) Automatic tuning of simple regulators with specifications on phase and amplitude margins. Automatica 20(5):645–651

Baidya D, Dhopte S, Bhattacharjee M (2023) Sensing system assisted novel PID controller for efficient speed control of DC motors in electric vehicles. IEEE Sens Lett 7(1):6000604

Bakir H, Guvenc U, Kahraman HT, Duman S (2022) Improved Lévy flight distribution algorithm with FDB-based guiding mechanism for AVR system optimal design. Comput Ind Eng 168(108032):1–29

Barakat M, Donkol A, Hamed HFA, Salama GM (2022) Controller parameters tuning of water cycle algorithm and its application to load frequency control of multi-area power systems using TD-TI cascade control. Evol Syst 13:117–132

Bouakkaz MS, Boukadoum A, Boudebbouz O, Fergani N, Boutasseta N, Attoui I, Bouraiou A, Necaibia A (2020) Dynamic performance evaluation and improvement of PV energy generation systems using Moth flame optimization with combined fractional order PID and sliding mode controller. Sol Energy 199:411–424

Cengiz E, Yilmaz C, Kahraman HT, Suicmez C (2021) Improved Runge Kutta optimizer with fitness-distance balance-based guiding mechanism for global optimization of high-dimensional problems. Duzce Univ J Sci Technol 9:135–149

Chiranjeevi T, Babu NR, Pandey SK, Patel RK, Gupta UK, Vais RI, Kumar A, Kumar D, Chaudhary A, Sonkar A, Pandey U (2021) Maiden application of flower pollination algorithm-based tilt integral derivative controller with filter for control of electric machines. Mater Today Proc 47(10):2541–2546

Choi HH, Yun HM, Kim Y (2015) Implementation of evolutionary fuzzy PID speed controller for PM synchronous motor. IEEE Trans Industr Inf 11(2):540–547

Cohen GH, Coon GA (1953) Theoretical consideration of retarded control. Trans Am Soc Mech Eng 75(5):827–834

Dey S, Banerjee S, Dey J (2022) Design of an improved robust fractional order PID controller for magnetic levitation system based on atom search optimization. Sadhana 47(188):1–19

Divya N, Manoharan S, Arulvadivu J, Palpandian P (2022) An efficient tuning of fractional order PID controller for an industrial control process. Mater Today Proc 57(4):1654–1659

Duman S, Kahraman HT, Guvenc U, Aras S (2021) Development of a Levy flight and FDB-based coyote optimization algorithm for global optimization and real-world ACOPF problems. Methodol Appl 25:6577–6617

Duman S, Kahraman HT, Sonmez Y, Guvenc U, Kati M (2022) A powerful meta-heuristic search algorithm for solving global optimization and real-world solar photovoltaic parameter estimation problems. Eng Appl Artif Intell 111(104763):1–31

Ekinci S, Izci D, Eker E, Abualigah L (2022) An effective control design approach based on novel enhanced aquila optimizer for automatic voltage regulator. Artif Intell Rev 56:1731–1762

Ekinci S, Izci D, Yilmaz M (2023a) Efficient speed control for DC motors using novel gazelle simplex optimizer. IEEE Access 11:105830

Ekinci S, Izci D, Eker E, Abualigah L, Thanh CL, Khatir S (2023b) Hunger games pattern search with elite opposite-based solution for solving complex engineering design problems. Evol Syst. https://doi.org/10.1007/s12530-023-09526-9

Elaziz MA, Mirjalili S (2019) A hyper-heuristic for improving the initial population of whale optimization algorithm. Knowl-Based Syst 172:42–63

Elaziz MA, Oliva D, Xiong S (2017) An improved opposition-based sine cosine algorithm for global optimization. Expert Syst Appl 90:484–500

Ergezer M, Simon D, Du D (2009) Oppositional biogeography-based optimization. In: Proceedings of the 2009 IEEE International Conference on Systems, Man, and Cybernetics. IEEE, pp 1009–1014

Ewees AA, Elaziz MA, Houssein EH (2018) Improved grasshopper optimization algorithm using opposition-based learning. Expert Syst Appl 112:156–172

Feng H, Ma W, Yin C, Cao D (2021) Trajectory control of electro-hydraulic position servo system using improved PSO-PID controller. Autom Constr 127(103722):1–14

Ghith ES, Tolba FAA (2023) Tuning PID controllers based on hybrid arithmetic optimization algorithm and artificial gorilla troop optimization for micro-robotics systems. IEEE Access 11:27138–27154

Guha D, Roy PK, Banerjee S (2017) Study of differential search algorithm based automatic generation control of an interconnected thermal-thermal system with governor dead-band. Appl Soft Comput 52:160–175

Guvenc U, Duman S, Kahraman HT, Aras S, Katı M (2021) Fitness–distance balance based adaptive guided differential evolution algorithm for security-constrained optimal power flow problem incorporating renewable energy sources. Appl Soft Comput 108(107421):1–35

Hasanhendoei GHR, Afjei E, Naseri M, Azad S (2023) Automatic and real time phase advancing in BLDC motor by employing an electronic governor for a desired speed-torque/angle profile. ePrime: Adv Electr Eng, Electron Energy 4(100111):1–11

Hemeida M, Osheba D, Alkhalaf S, Fawzy A, Ahmed M, Roshdy M (2023) Optimized PID controller using Archimedes optimization algorithm for transient stability enhancement. Ain Shams Eng J. https://doi.org/10.1016/j.asej.2023.102174

Isen E (2022) Determination of different types of controller parameters using metaheuristic optimization algorithms for buck converter systems. IEEE Access 10:127984–127995

Izci D (2021) Design and application of an optimally tuned PID controller for DC motor speed regulation via a novel hybrid Lévy flight distribution and Nelder–Mead algorithm. Trans Inst Meas Control 43(14):3195–3211

Izci D, Ekinci S (2021) Comparative performance analysis of slime mould algorithm for efficient design of proportional–ıntegral–derivative controller. Electrica 21(1):151–159

Izci D, Ekinci S (2023) Fractional order controller design via gazelle optimizer for efficient speed regulation of micromotors. e-Prime - Adv Electr Eng Electron Energy 6:1–9

Izci D, Ekinci S, Zeynelgil HL, Hedley J (2021) Fractional order PID design based on novel ımproved slime mould algorithm. Electr Power Compon Syst 49(910):901–918

Izci D, Ekinci S, Eker E, Kayri M (2022a) Augmented hunger games search algorithm using logarithmic spiral opposition-based learning for function optimization and controller design. J King Saud Univ – Eng Sci. https://doi.org/10.1016/j.jksues.2022.03.001

Izci D, Hekimoglu B, Ekinci S (2022b) A new artificial ecosystem-based optimization integrated with Nelder–Mead method for PID controller design of buck converter. Alex Eng J 61(3):2030–2044

Izci D, Ekinci S, Mirjalili S (2023) Optimal PID plus second-order derivative controller design for AVR system using a modified Runge Kutta optimizer and Bode's ideal reference model. Int J Dyn Control 11:1247–1264

Jeon H, Lee J, Han S, Kim JH, Hyeon CJ, Kim HM, Kang H, Ko TK, Yoon YS (2018) PID control of an electromagnet-based rotary HTS flux pump for maintaining constant field in HTS synchronous motors. IEEE Trans Appl Supercond 28(4):5207605

Kahraman HT, Aras S, Gedikli E (2020) Fitness-distance balance (FDB): a new selection method for meta-heuristic search algorithms. Knowl-Based Syst 190(105169):1–27

Kahraman HT, Bakir H, Duman S, Katı M, Aras S, Guvenc U (2022) Dynamic FDB selection method and its application: modeling and optimizing of directional overcurrent relays coordination. Appl Intell 52:4873–4908

Kaucic M (2013) A multi-start opposition-based particle swarm optimization algorithm with adaptive velocity for bound constrained global optimization. J Glob Optim 55:165–188

Li P, Wang Y, Zuo Z (2023) Robust multiple frequency design on voltage-mode control of DC–DC boost converters. J Franklin Inst 360(2):1207–1225

Liang Z, Fu L, Li X, Feng Z, Sleigh JW, Lam HK (2019) Ant colony optimization PID control of hypnosis with propofol using Renyi permutation entropy as controlled variable. IEEE Access 7:97689–97703

Liu Y, Gao J, Zhong Y, Zhang L (2021) Extended state observer-based IMC-PID tracking control of PMLSM servo systems. IEEE Access 9:49036

Majhi SK, Sahoo M, Pradhan R (2021) Oppositional crow search algorithm with mutation operator for global optimization and application in designing FOPID controller. Evol Syst 12:463–488

Malarvili S, Mageshwari S (2022) Nonlinear PID (N-PID) controller for SSSP grid connected inverter control of photovoltaic systems. Electr Power Syst Res 211(108175):1–13

Mourtas SD, Kasimis C, Katsikis VN (2023) Robust PID controllers tuning based on the beetle antennae search algorithm. Mem - Mater, Devices, Circuits Syst 4(100030):1–5

Rahnamayan S, Tizhoosh HR, Salama MMA (2007) Quasi-oppositional differential evolution. In: 2007 IEEE Congress on Evolutionary Computation (CEC 2007). IEEE, pp 2229–2236

Rai A, Das DK (2022) The development of a fuzzy tilt integral derivative controller based on the sailfish optimizer to solve load frequency control in a microgrid, incorporating energy storage systems. J Energy Storage 48(103887):1–11

Rizk-Allah RM, El-Fergany AA (2021) Artificial ecosystem optimizer for parameters identification of proton exchange membrane fuel cells model. Int J Hydrogen Energy 46:37612–37627

Sabir MM, Khan JA (2014) Optimal design of PID controller for the speed control of DC motor by using metaheuristic techniques. Adv Artif Neural Syst. https://doi.org/10.1155/2014/126317

Sahin E (2019) Design of a PID controller with fractional order derivative filter for automatic voltage regulation in power systems. In: 4th International Symposium on Innovative Approaches in Engineering and Natural Sciences 4(6):23–27

Sahoo BP, Panda S (2018) Improved grey wolf optimization technique for fuzzy aided PID controller design for power system frequency control. Sustain Energy, Grids Netw 16:278–299

Sahu RK, Panda S, Padhan S (2014) Optimal gravitational search algorithm for automatic generation control of interconnected power systems. Ain Shams Eng J 5(3):721–733

Sahu RK, Panda S, Biswal A, Sekhar GTC (2016) Design and analysis of tilt integral derivative controller with filter for load frequency control of multi-area interconnected power systems. ISA Trans 61:251–264

Singh K, Amir M, Ahmad F, Khan MA (2021) An integral tilt derivative control strategy for frequency control in multimicrogrid system. IEEE Syst J 15(1):1477–1488

Singh B, Bishnoi SK, Sharma M, Singh P, Dhundhara S (2023) An application of nature inspired algorithm based dual-stage frequency control strategy for multi micro-grid system. Ain Shams Eng J. https://doi.org/10.1016/j.asej.2023.102125

Sonmez Y, Duman S, Kahraman HT, Kati M, Aras S, Guvenc U (2022) Fitness-distance balance based artificial ecosystem optimisation to solve transient stability constrained optimal power flow problem. J Exp Theor Artif Intell. https://doi.org/10.1080/0952813X.2022.2104388

Suicmez C, Kahraman HT, Yilmaz C, Isik MF, Cengiz E (2021) Improved slime-mould-algorithm with fitness distance balance-based guiding mechanism for global optimization problems. Duzce Univ J Sci Technol 9:40–54

Tang Y, Cui M, Hua C, Li L, Yang Y (2012) Optimum design of fractional order PIλDμ controller for AVR system using chaotic ant swarm. Expert Syst Appl 39:6887–6896

Tizhoosh HR (2005) Opposition-based learning: a new scheme for machine intelligence. In: Proceedings of the 2005 International Conference on Computational Intelligence for Modelling, Control and Automation, and International Conference on Intelligent Agents, Web Technologies and Internet Commerce (CIMCA-IAWTIC'05). IEEE. pp 1–7

Topno PN, Chanana S (2016) Automatic generation control using optimally tuned tilt integral derivative controller. In: 2016 IEEE First International Conference on Control, Measurement and Instrumentation (CMI). IEEE. pp 206–210

Tripathy S, Debnath MK, Kar SK (2021) Jaya algorithm tuned FO-PID controller with first order filter for optimum frequency control. In: 2021 1st Odisha International Conference on Electrical Power Engineering, Communication and Computing Technology (ODICON). IEEE. pp1–6

Veinovic S, Stojic D, Joksimovic D (2022) Optimized four-parameter PID controller for AVR systems with respect to robustness. Int J Electr Power Energy Syst 135(107529):1–12

Viaene JD, Ceulemans D, Derammelaere S, Stockman K (2022) Sensorless load angle control for energy optimal sinusoidal driven BLDC motor applications. IEEE/ASME Trans Mechatron 27(4):2290–2300

Xia T, Zhang Z, Hong Z, Huang S (2023) Design of fractional order PID controller based on minimum variance control and application of dynamic data reconciliation for improving control performance. ISA Trans 133:91–101

Xu Z, Liu J (2022) Research on temperature control of liposome high pressure homogenizer based on genetic algorithm optimization PID. Procedia Comput Sci 208:330–337

Yousi D, Babu TS, Mirjalili S, Rajasekar N, Elaziz MA (2020) A novel objective function with artificial ecosystem-based optimization for relieving the mismatching power loss of large-scale photovoltaic array. Energy Convers Manag 225(113385):1–18

Yue CT, Price KV, Suganthan PN, Liang JJ, Ali MZ, Qu BY, Awad NH, Biswas PP (2019) Problem definitions and evaluation criteria for the CEC 2020 special session and competition on single objective bound constrained numerical optimization. Computational Intelligence Laboratory, Zhengzhou University, Zhengzhou, China: Technical Report

Zeng W, Zhu W, Hui T, Chen L, Xie J, Yu T (2020) An IMC-PID controller with particle swarm optimization algorithm for MSBR core power control. Nucl Eng Des 360(110513):1–7

Zhang R, Gao L (2022) The Brushless DC motor control system based on neural network fuzzy PID control of power electronics technology. Optik 271(169879):1–13

Zhao W, Wang L, Zhang Z (2020) Artificial ecosystem-based optimization: a novel nature-inspired meta-heuristic algorithm. Neural Comput Appl 32:9383–9425

Ziegler JG, Nichols NB (1942) Optimum settings for automatic controllers. Trans Am Soc Mech Eng 64(8):759–765