Journal of Electrical Engineering & Technology (2022) 17:3565–3580
https://doi.org/10.1007/s42835-022-01149-5

ORIGINAL ARTICLE

Optimal Tuning of Servo Motor Based Linear Motion System Using Optimization Algorithm

H. Ramesh[1] · S. Arockia Edwin Xavier[2]

Received: 29 September 2021 / Revised: 26 May 2022 / Accepted: 6 June 2022 / Published online: 11 July 2022

Abstract
Linear motion systems with servo drives are employed in high-precision machine tool applications. The PID controller is commonly employed in servo-based linear motion systems to correct positioning inaccuracies caused by thermal expansion of the ball screw assembly and encoder measurement. Different classical and heuristic approaches are used for optimal PID tuning of servo controllers used in linear motion systems. Integral-based or performance-index- based error minimizing functions found in the literature do not meet all of a dynamic system's performance requirements. In this paper, the multi-objective cost function using both the integral time absolute error function and performance index parameters such as rise time, settling time, and peak overshoot is formulated based on the non-dominated solutions of the pareto front obtained using a multi-objective genetic algorithm (MOGA). The proposed objective function is used to tune the PID controller model of a linear motion system using the particle swarm optimization algorithm, the BAT algorithm, the whale optimization algorithm, and the aquila optimizer. The simulation and validation results show that the MOGA-based multi-objective function outperforms standard error minimizing objective functions and classical fractional order PID control algorithms in tuning PID Servo controllers of linear motion systems.

Keywords Linear motion system · Ball screw assembly · PID controller · FOPID · MOGA · PSO · WOA · AO · BAT

1 Introduction

Ball screw-based linear motion systems are widely used in precision motion tool applications [1]. The performance of the servo controller used in linear motion systems depends on its response to the machine dynamics. The PID control algorithm is employed in most industrial controllers due to its simple, efficient and easy implementation. It can use proportional action to correct errors, integral action to eliminate steady state offsets, and derivative action to anticipate the future [2]. The PID-based servo controller used in linear motion systems should be optimally tuned to respond to the positional errors due to feed screw pitch and torsion errors [3], temperature induced errors [4, 5], and encoder measurement errors [6]. The traditional PID tuning algorithms [7, 8] use complex equations which require domain expertise to design an optimal controller for motion control applications. Also, since these algorithms focus on specific operating characteristics of the system, they will not respond appropriately when those values change. Hence, the tuning of PID controllers using meta-heuristic optimization algorithms has been proposed by researchers in recent decades [9]. The optimization algorithms used in PID controllers search for optimal tuning parameters. The objective function is defined in the optimization algorithms to meet the specific performance criterion [10].

Kitsios and Pimenides designed a PID controller for a servo motor using genetic algorithm (GA) techniques. Since the integral squared error (ISE) function weights errors equally independent of time which results in a long settling time, the integral time squared error (ITSE) is used as performance criteria to improve the step response of a controller [11]. Mirzal et al. compared the results of objective functions of integral time absolute error (ITAE), integral absolute error (IAE), mean squared error (MSE), ITSE, and ISE,

✉ H. Ramesh
rameshh@tce.edu

S. Arockia Edwin Xavier
saexeee@tce.edu

[1] Department of Mechatronics Engineering, Thiagarajar College of Engineering, Madurai, India

[2] Department of Electrical and Electronic Engineering, Thiagarajar College of Engineering, Madurai, India

in tuning GA based PID controllers [12]. Mohd Sazli et al. implemented a PID controller using GA and a Differential Evolution (DE) algorithm by assigning MSE and IAE as their objective functions [13]. Renato A. Krohling and Joost P. Rey designed a GA based optimal PID algorithm which uses ITSE as its performance index [14]. A GA based optimized shaped permanent magnet model is used for improving the performance parameters of the permanent magnet Vernier machine [15, 16]. Anil kumar and Giriraj kumar used MSE, IAE, ITAE, and ISE as their error minimizing performance indices for tuning a PID controller using the whale optimization algorithm [17]. WaelNaji et al. proposed GA based optimization of a PID controller for a multi variable process in which ITAE is chosen as a main performance criterion due to its shorter settling time and overshoot [18].

Since the classical error minimizing functions such as ITAE, ISE, IAE, and MSE are insufficient for enhancing optimal tuning of PID parameters, the time domain parameters such as settling time (T_s), rise time (T_r), steady state error (Ess) and percentage of overshoot (M) are included in the objective function. The weights are selected for the above criteria based on the performance requirements of the user [19]. Latha et al. proposed a PSO based multi-objective algorithm for tuning the PID controller of stable and unstable systems. The multi-objective function is formulated using a weighted sum of ISE and time domain constants such as overshoot M_p and settling time t_s. The weights for the above three constraints are selected as $w1 = w2 = 1$ and $w3 = 0.5$ [20]. Arturo Y et al. proposed a GA based multi-objective function which uses a weighted sum of ISE, MSE, and peak overshoot value for tuning servo systems. The weights for the above constraints are selected as $w1 = w2 = 0.3$and $w3 = 0.4$ [21]. Andrey et al. proposed a GA based multi objective optimization technique which uses two objective functions independently to provide better reference tracking and disturbance rejection [22]. A GA-based optimized model is used for improving the performance parameters of axial flux permanent magnet machine (AFPM) [23, 24]. Oguzhan Karahan proposed a multi-objective cost function that includes four performance parameters such as steady state error, overshoot, settling time, and rise time for optimal tuning of PID parameters using the cuckoo search algorithm [25]. M. H. A. Hassan used a modified ITAE function for tuning the PID parameters of a brushed dc motor [26]. Ayman A.aly proposed a multi-objective function based on ITAE, peak overshoot, and steady state error, in which the weights for the objective function are selected randomly by a user [27].

According to the research papers, researchers choose the error minimizing functions ISE, IAE, MSE, ITSE, and ITAE for tuning the PID controller depending on the nature of the applications. Few scholars use a weighted sum function that combines any one or more of the error minimization functions with performance criteria functions (T_r, T_s, M_p) to achieve optimal performance results. But, the weights for these functions are selected based on user requirements or through an error and trial approach.

In this paper, a new multi-objective function that includes ITAE and time domain parameters such as overshoot, rise time, and settling time is used for optimal tuning of servo controller parameters. The weights for the above performance criteria functions are obtained from MOGA pareto optimal solutions. The novel multi-objective function is evaluated with the model of a servo based linear motion system using PSO, WOA, BAT, and AO algorithms. The obtained results using the proposed objective function show superior results to those obtained using conventional error minimizing functions such as ITAE and typical PID and FOPID controller algorithms.

2 Mathematical Modeling of Linear Motion System

Figure 1 shows the schematic diagram of a linear motion system that consists of a DC servo motor and ball screw assembly. The rotary motion provided by the DC servo motor is converted into a linear motion using the ball screw

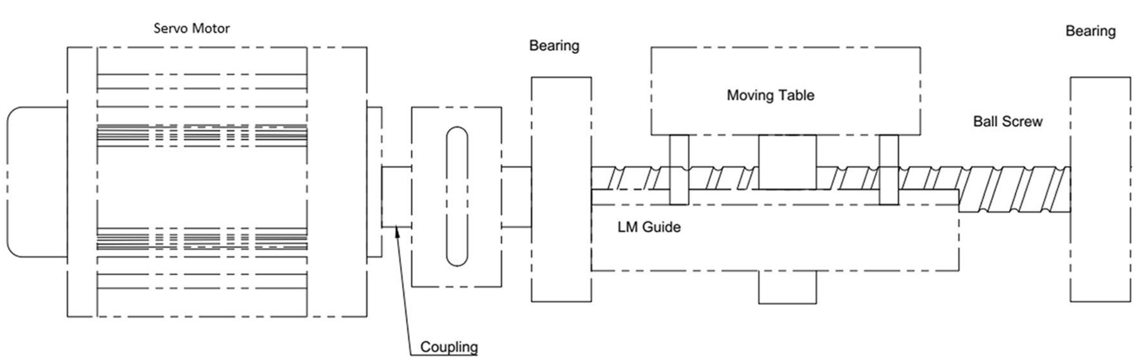


Fig. 1 Schematic diagram of linear motion system

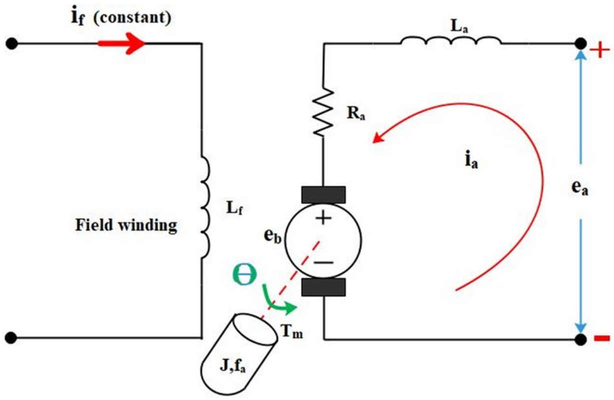


Fig. 2 Electrical equivalent circuit of DC servo motor

assembly. Ball screws are often stated in terms of lead, which is the linear movement the nut makes per one screw revolution.

The transfer function for the linear motion system is derived using the equations of the DC servo motor and ball screw assembly [28]. The electrical circuit of the DC servo motor is given in Fig. 2. The transfer function of a DC servo motor [29] is given in Eq. (1).

$$\frac{\theta(s)}{V_a(s)} = \frac{k_1}{\left[J_m s^2 + B_m s\right]\left[L_a s + R_a\right] + k_t k_b s} \tag{1}$$

where θ(s) is angular position, J_m, B_m are mechanical constants, R_a, L_a are armature resistance and Inductance, k_1, k_b is Torque and emf constants

The mechanical constants J_m and B_m must be specified to analyze the DC servo motor coupled to the ball screw assembly. The mechanical constants J_m and B_m is calculated using the gear box relationship, $N_1$and N_2, the inertia J_L and damping B_L of the load, as given in Eq. (2)

$$J_m = J_a + J_L\left(\frac{N_1}{N_2}\right)^2; \quad B_m = B_a + B_L\left(\frac{N_1}{N_2}\right)^2 \tag{2}$$

The transfer function of linear motion system is given by

$$G(s) = \frac{k_1}{\left[P[J_m s^2 + B_m s]\left[L_a s + R_a\right] + k_1 k_b s\right]} \tag{3}$$

where P = 2π/L, L represents the lead of the screw.

The specifications of the linear motion control system [30] consisting of a DC servo motor and ball screw assembly are given in Table1. Substituting the parameter values of the linear motion system into Eq. 3, gives the transfer function as

$$G(s) = \frac{232600}{s^3 + 18.44s^2 + 30.37s} \tag{4}$$

Table 1 Parameters of linear motion control system

Parameter	Definition	Values
B_m	Equivalent viscous friction coefficient [Nms/rad]	0.02
B_L	Load damping constant [Nms/rad]	1
J_a	Motor inertial constant [Kgm^2]	0.02
B_a	Motor damping constant [Nms/rad]	0.01
J_L	Load inertial constant [Kgm^2]	1
K_t	Motor torque constant [Nm/A]	0.5
J_m	Equivalent moment of inertia [Kgm^2]	0.03
K_b	Back emf constant [Vs/rad]	0.5
L_a	Motor armature inductance [H]	0.45
R_a	Motor armature resistance [Ω]	8
N_1, N_2	Gear teeth (respectively)	25,250
L	Lead of the screw (mm)	1

3 MOGA Based Objective Function

The objective functions used for tuning PID controllers can be classified into integral and performance-index based functions. The integral functions that are used to tune PID controllers are as follows:

$$IAE = \int_0^t |e(t)|\mathrm{d}t \tag{5}$$

$$ITAE = \int_0^t t|e(t)|\mathrm{d}t \tag{6}$$

$$ISE = \int_0^t e^2(t)\mathrm{d}t \tag{7}$$

$$MSE = \frac{1}{t}\int_0^t (e(t))^2\mathrm{d}t \tag{8}$$

where e (t) is the system error, which is the difference between the set point and actual value.

The maximum overshoot, steady state error, rise time and settling time are the performance based indices functions used to tune the PID controller.

$$\text{Minimize (F1, F2, F3, F4)} \tag{9}$$

$$\text{Minimize } \left(\text{ITAE}, \ \text{T}_\text{r}, \text{T}_\text{s}, \ \text{M}_\text{p}\right) \tag{10}$$

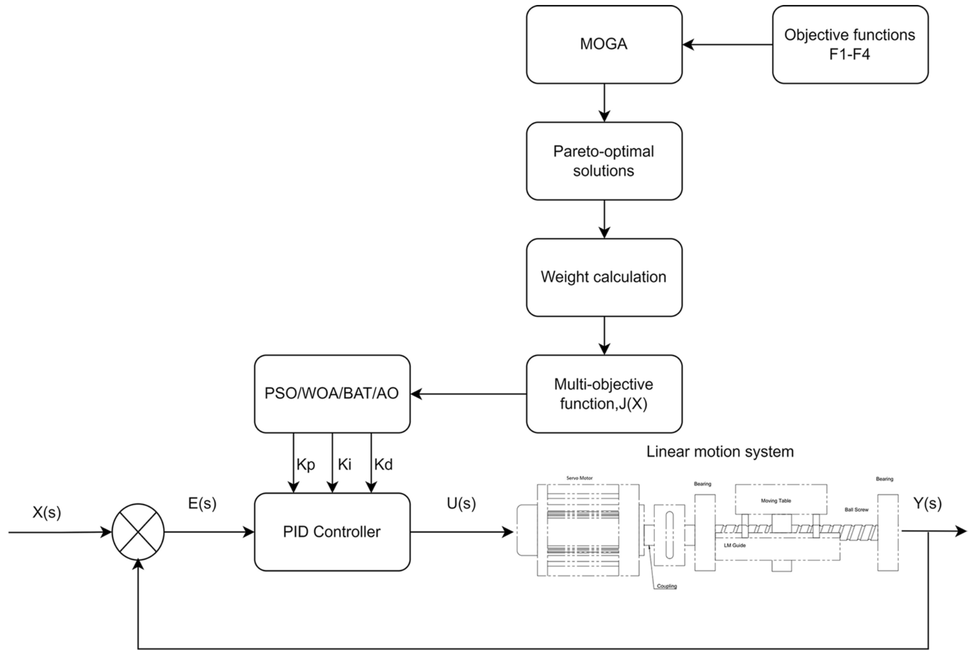


Fig. 3 Block diagram of MOGA based PID Tuning

Equation (9) refers four objective functions of MOGA ie. F1, F2, F3, F4, where function F1 represents the integral based error minimizing function (ITAE) and the function F2, F3, F4 are rise time, settling time and peak overshoot respectively.

The MOGA algorithm generates a set of pareto optimal solutions denoted by $Po = \left\{ \overrightarrow{k_{po1}}, \overrightarrow{k_{po2}}, \ldots, \overrightarrow{k_{pon}} \right\}$ based on the above objective functions for optimal tuning of the PID controller. Given the objective function $\overrightarrow{f(k)} = \left[f_1(\vec{k}), f_2(\vec{k}), \ldots, f_m(\vec{k}) \right]$, pareto front generated by MOGA is given by:

$$P_f = \left\{ \begin{matrix} f_1\left(\overrightarrow{k_{po1}}\right) & f_2\left(\overrightarrow{k_{po1}}\right) & \cdots & f_m\left(\overrightarrow{k_{po1}}\right) \\ f_1\left(\overrightarrow{k_{po2}}\right) & f_2\left(\overrightarrow{k_{po2}}\right) & \cdots & f_m\left(\overrightarrow{k_{po2}}\right) \\ f_1\left(\overrightarrow{k_{pon}}\right) & f_2\left(\overrightarrow{k_{pon}}\right) & \cdots & f_m\left(\overrightarrow{k_{pon}}\right) \end{matrix} \right\} \tag{11}$$

The pareto front generated by a MOGA is converted into a single weighted sum of objective functions given by:

$$J\left(\vec{k}\right) = \sum_{i=1}^{m} w_i f_i\left(\vec{k}\right) \tag{12}$$

where w_i are the weights of the objective functions that give the relative importance of the individual objective functions on the overall multi-objective function. The weights of the individual objective functions are calculated using the following relation:

$$w_i = \frac{1}{\mu_i \times \sum_{j=1}^{l} \frac{1}{\mu_j}} \tag{13}$$

where μ_i and μ_j are the mean values of the pareto solutions obtained using individual objective functions.

Figure 3 is the block diagram of proposed MOGA based PID tuning of linear motion system. The proposed objective function is formulated by the weighted sum of the objective function given by:

$$J(X) = w_{IT} J_{IT} + w_r J_r + w_s J_s + w_{po} J_{po} \tag{14}$$

where w_{IT}, w_r, w_s, w_{po} are the weights for ITAE, rise time, settling time, and peak overshoot calculated using pareto optimal values obtained using the MOGA algorithm.

4 Servo PID Tuning Algorithms

4.1 PSO Servo-PID Tuning

Particle Swarm Optimization (PSO) is a popular stochastic optimization approach that is based on social behavior. In PSO, the particle adjusts its movement in order to achieve its individual best position as well as the global best position achieved by any member of its neighborhood [31]. The pseudo code for PSO algorithm is given below.

Procedure PSO
Initialize Swarm Size
Intialize $W_{max,}$ W_{min},c_1,c_2
Do
***for** each particle p with position $x_{p,}$ do calculat e fitness function f(x_p)*
***if** f(x_p) is better than $pbest_p$ then*
$pbest_b \leftarrow x_p$
endif
end for
define $gbest_b$ as the best position found in the nei ghbourhood of 'p' particles
***for** each particle p do*
$V_p \leftarrow$ *calculate velocity*
$X_p \leftarrow$ *update position*
end for
***while** (The maximum number of iterations has no t been reached, or a stop requirement has not been m et)*

In each iteration, the particle position and velocity are updated using the following equations:

$$V_i^{k+1} = wV_i^{k+1} + C_1 r_{i1}^k \left(P_i^k - X_i^k\right) + C_2 r_{i2}^k \left(P_g^k - X_i^k\right) \tag{15}$$

$$X_i^{k+1} = X_i^k + V_i^{k+1} \tag{16}$$

where i is the swarm size, w is the inertia weight, r_{i1} and r_{i2} are random numbers uniformly distributed within the range of 0 to 1, and c_1 and c_2 are the cognitive and social parameters, respectively. Figure 4 is the pseudo code for the PSO algorithm. At each iteration, the PSO algorithm updates the position and velocity of each particle based on the given objective function.

4.2 WOA Based Servo PID Tuning

In the WOA optimization technique, the behavior of humpback whales attacking prey using a method of prey encircling, bubble-net strategy and search for prey is mathematically modelled. The humpback whales travel in a shrinking circle around the prey while also swimming along a spiral-shaped path [32]. The behavior of encircling the prey is modelled by the equation:

$$\vec{D} = \left|\vec{C}.\vec{X}*(t) - \vec{X}(t)\right| \tag{17}$$

$$\vec{X}(t+1) = \vec{X}*(t) - \vec{A}.\vec{D} \tag{18}$$

where X* is the updated best position vector, $\vec{X}$ is the current position vector and $\vec{A}, \vec{C}$ are coefficient vectors calculated as follows:

$$\vec{A} = 2\vec{a}.\vec{r} - \vec{a}$$

$$\vec{c} = 2.\vec{r}$$

where $\vec{r}$ is a random vector in [0,1], $\vec{a}$ is decreased linearly from 2 to 0 during the process of iterations.

The behavior of bubble net attacking method is modelled as:

$$\vec{X}(t+1) = \begin{cases} \vec{X}*(t) - \overrightarrow{A}.\vec{D} & if \;\; p < 0.5 \\ \overrightarrow{D'}.e^{bl}.\cos(2\pi l) + \vec{X}*(t) & if \;\; p \geq 0.5 \end{cases} \tag{19}$$

where p is a random number in [0,1], b is a constant representing the shape of the logarithmic spiral, l is a random number in [− 1,1].

The behavior of random prey search is modelled as follows:

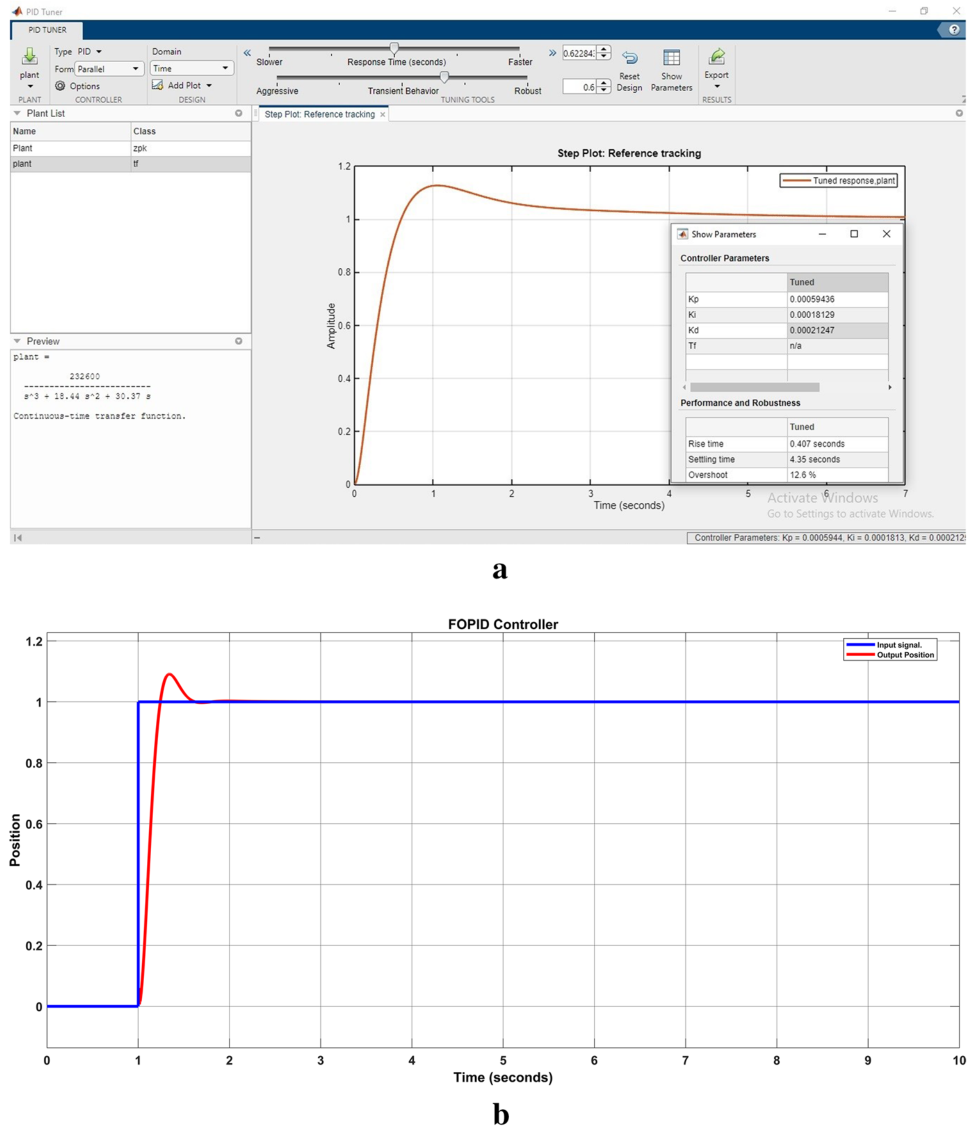
$$\vec{D} = \left|\vec{C}.\overrightarrow{X_{rand}} - \vec{X}\right| \tag{20}$$

$$\vec{X}(t+1) = \overrightarrow{X_{rand}} - \vec{A}.\vec{D} \tag{21}$$

The pseudo code for WOA algorithm is given below:

Procedure WOA
Initialize the whales population X_w
Calculate each search agent fitness (K_p ,K_i ,K_d)
X=compute the the best search agent (K_p ,K_i ,K_d)*
***while**(t<maximum iterations)*
***for** each search agent (K_p ,K_i ,K_d)*
Update a, A, C, l and p
***If1**(p<0.5)*
***If2**(|A|<1)*
Update current search age nt position (K_p ,K_i ,K_d) by the Eq.(18)
***else if2**(|A|>1)*
Select a random search agent (X_{rand})
Update the position of the curr ent search agent (K_p ,K_i ,K_d)by the Eq.(21)
end if2
***else if1**(p≥0.5)*
Update current search position val ues (K_p ,K_i ,K_d)by the Eq.(15)
end if1
end for
Modify the search values (K_p ,K_i ,K_d)which goes beyond the search space
Calculate the fitness of each search agent (K_p ,K_i ,K_d)
Update X if there is a better solution*
t=t+1
end while
*return X**

Fig. 4 PID tuning of a linear motion system using **a** Matlab PID auto-tuner **b** FOPID Controller



4.3 Bat Based Servo PID Tuning

The Bat algorithm uses echolocation characteristics of micro bats to find prey. Bats fly at a random velocity v_i at point x_i with a fixed frequency f_i, varying its wavelength l, and loudness A0 in search of prey. They may automatically adjust the wavelength (or frequency) of their generated pulses as well as the rate of pulse emission r in the range [0, 1] depending on the proximity of their target [33]. Each bat's frequency, velocity, and position are updated as follows:

$$f_i = f_{\min} + \left(f_{\max} - f_{\min}\right)\beta \tag{22}$$

$$v_i(t) = v_i(t-1) + \left(x_i(t) - x*\right) \tag{23}$$

$$x_i(t) = x_i(t-1) + v_i(t) \tag{24}$$

where β is a random number in [0, 1], x* is the current global best position obtained by comparing the fitness values of all the n bats. The pseudo code for the BAT algorithm is given below:

Procedure Bat
Initialize the bat position x_i (K_p ,K_i ,K_d) and velo city, v_i (i=1,2,3,4....n)
Initialize frequency f_i , pulse rate r_i and the loud ness A_i
***While**(t<Max iterations)*
Generate new positions (K_p, Ki, Kd) by changing frequency, and updating
Velocities and positions by equations (22) t o (24)
***If** (rand<r_i)*
Select a position (K_p ,K_i ,K_d) among the best positions
Generate the local position among the sel ected
end if
Generate a new position (K_p ,K_i ,K_d) by flying r andomly
***If** (rand <A_i and f(x_i)<f(x*))*
Accept the new positions
Increase r_i and reduce A_i
end if
*Rank the bats and find the current best position x**
end while

4.4 Aquila Based Servo PID Tuning

The Aquila optimizer is a newly designed algorithm based on the Aquila's prey-catching behavior. The Aquila catches its prey using the four methods listed below. 1) By completing a high soar with a vertical stoop, it recognizes the prey area and chooses the optimum hunting area (Enlarged exploration) 2) When a high soar locates a prey spot, the Aquila circles over it, prepares the land, and then attacks, a maneuver known as Contour flight with short glide attack(Narrowed exploration) 3) Once the prey location has been precisely detected, the Aquila descends vertically with a preliminary attack to detect the prey reaction, a method known as "low flying with slow descent attack" (Enlarged exploitation) 4) When the Aquila gets close enough to the target, it uses stochastic motions dubbed "walking and grabbing" to attack the prey on the ground [34]. These four methods can be mathematically modelled as given below:

The enlarged exploration is given by the equation:

$$X_1(t+1) = X_{best}(t) \times \left(1 - \frac{t}{T}\right) + \left(X_M(t) - X_{best}(t) * rand\right) \tag{25}$$

where X_{best} (t) is the best position obtained until the tth iteration, the term (1 − t/T) is used to control the expanded exploration, and X_M(t) is the mean position value of the current solutions at the tth iteration.

The narrowed exploration is given by the equation:

$$X_2(t+1) = X_{best}(t) \times Levy(D) + X_R(t) + (y - x) * rand \tag{26}$$

where X_2 (t + 1) is the position of the next iteration of t, D is the dimension space, X_R(t) is the random position taken in the range of [1N] at the ith iteration and Levy(D) is the levy flight distribution function which is given by:

$$\text{Levy(D)} = s \times \frac{\mu \times \sigma}{|v|^{\frac{1}{\beta}}} \tag{27}$$

where s is a constant value assigned as 0.01, u and v are random numbers between 0 and 1 and σ is calculated using the equation:

$$\sigma = \left(\frac{(1+\beta) \times \sin e\left(\frac{\pi\beta}{2}\right)}{\left(\frac{1+\beta}{2}\right) \times \beta \times 2^{\left(\frac{\beta-1}{2}\right)}} \right) \tag{28}$$

where β is a fixed constant value of 1.8.

In Eq. (26), y and x are used to configure the spiral shape in the search, which are calculated using the equation:

$$y = r \times \cos(\theta) \tag{29}$$

$$x = r \times \sin(\theta) \tag{30}$$

where,

$$r = r_1 + U \times D_1 \tag{31}$$

$$\theta = -w \times D_1 + \theta_1 \tag{32}$$

$$\theta_1 = \frac{3 \times \pi}{2} \tag{33}$$

The enlarged exploitation is modelled using the equation:

$$X_3(t+1) = \left(X_{best}(t) - X_M(t)\right) \times \alpha - rand + ((UB - LB) \times rand + LB) \times \delta \tag{34}$$

where X_3 (t + 1) is the position obtained using the third search method for the next iteration of t, X_{best} (t) is the best obtained position until ith iteration, X_M (t) is the mean value of the current position at tth iteration, rand refers to a random value in [0,1], α and δ are the exploitation adjustment parameters kept at small values.LB and UB are the lower and upper bounds of the PID tuning parameters.

The narrowed exploitation is given by the equation:

$$X_4(t+1) = QF \times X_{best}(t) - \left(G_1 \times X(t) \times rand\right) - G_2 \times Levy(D) + rand \times G_1 \tag{35}$$

Table 2 Tuning and performance parameters of PID and FOPID controller

Optimizing algorithm	Servo PID tuning parameters			Performance parameters			
	$K_p \times 10^{-3}$	$K_i \times 10^{-3}$	$K_d \times 10^{-3}$	M_p	T_r	T_s	Ess
PID auto-tuner	0.59	0.18	0.21	12.6	0.42	4.35	0.009
FOPID	0.48	0.17	0.21	9.1	0.45	1.53	0.004

where $X_4(t+1)$ is the position obtained using the fourth search method for the next iteration of t, QF refers to the quality function at tth iteration given by the equation:

$$QF(t) = t^{\frac{2\times rand()-1}{((1-T)^2)}} \tag{36}$$

G_1 is the different motions used by Aquila to track the prey given by the equation:

$$G_1 = 2 \times \text{rand}() - 1 \tag{37}$$

G_2 refers to the flight slope of Aquila having values decreasing from 2 to 0, which is given by the equation**:**

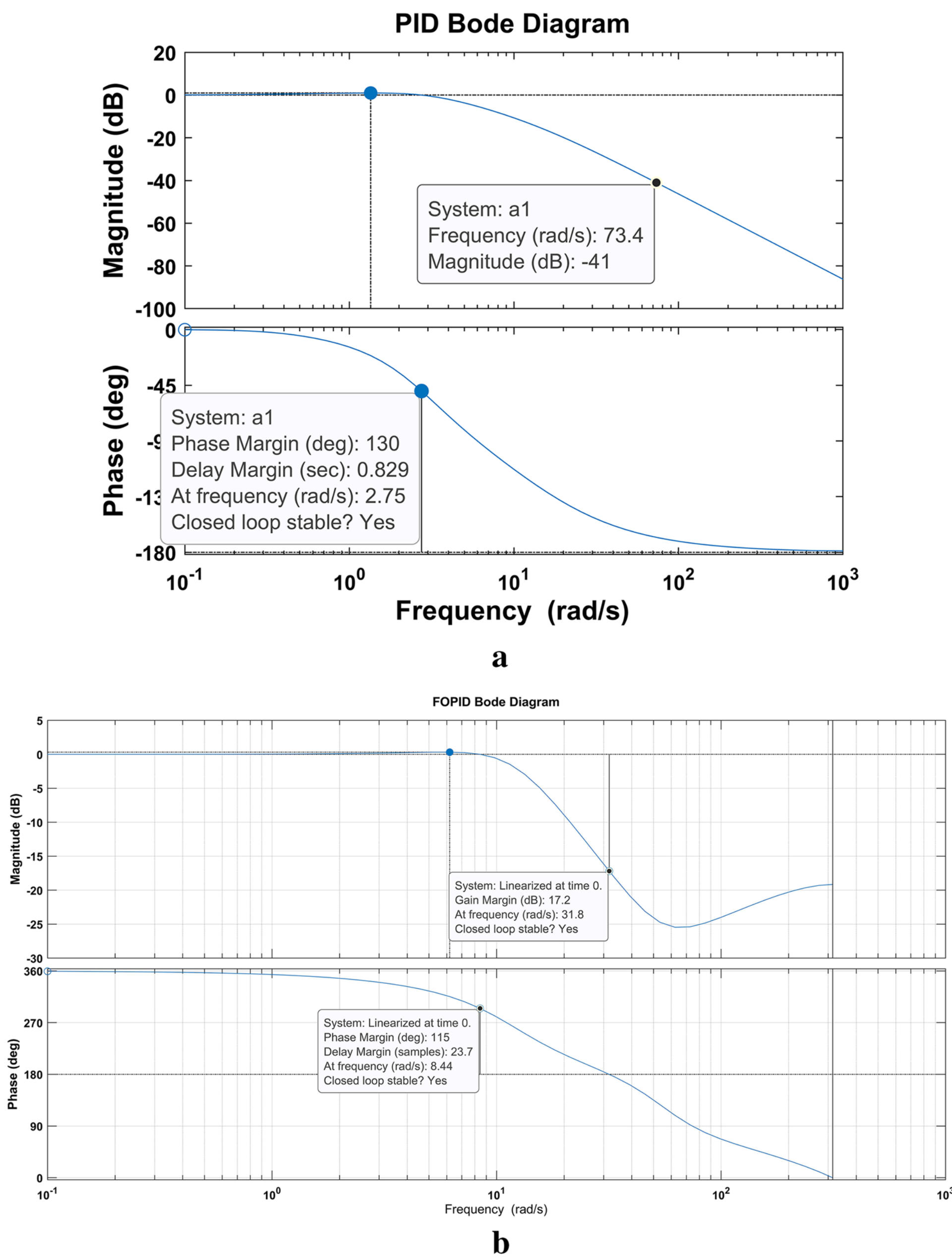


Fig. 5 Bode plot of a linear motion system tuned using **a** PID controller **b** FOPID controller

Table 3 MO-GA Algorithm parameters for tuning linear motion control system

MO-GA algorithm parameters
Population size = 50
Cross over fraction = 0.8
Cross over function = intermediate; cross over ratio = 1
Migration fraction = 0.2
Migration interval = 20
Selection criteria = Tournament; Tournament size = 2
Mutation criteria = Constraint-dependent
Pareto Fraction = 0.35
Distance measure function: @distancecrowding
MaxStallGenerations = 100

Table 4 Objective function weight calculation using pareto front sets

	Pareto front sets			
	ITAE	T_r	T_s	M_p
Mean value, μ	0.02103	0.1577	3.382	2.594
Contribution percentage	0.00342	0.0256	0.5494	0.4215
Weight value	0.87129	0.1162	0.0054	0.0071

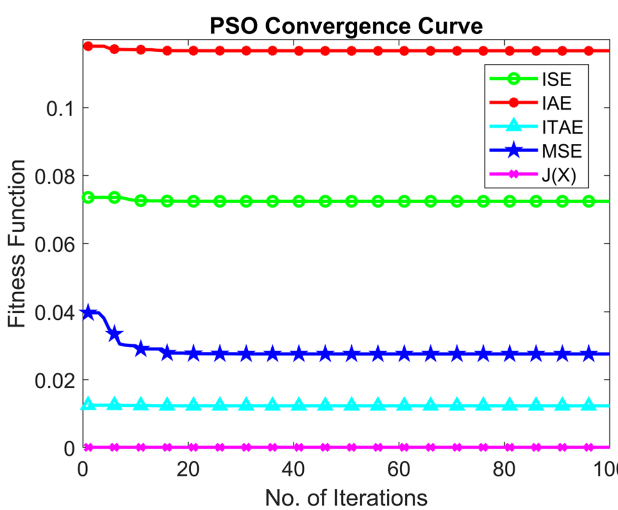


Fig. 6 PSO convergence curve for various objective functions

$$G_2 = 2 \times \left(1 - \frac{t}{T}\right) \tag{38}$$

The pseudo code for AO algorithm is given below.

Procedure AO
Initialize population X_a for Servo PID tuning
Initialize the exploitation adjustment parameters (α,δ, etc).
do
Calculate the fitness function values (K_p ,K_i ,K_d)
Generate the best result ($X_{best}(t)$) based on the fitness values.
for *(i=1,2…,N)**do***
Update the current result mean value $X_M(t)$
Update the G_1,G_2,x,y,Levy(D)
if $t \le \left(\frac{2}{3}\right) * T$ ***then***
if *rand≤0.5* ***then***
{Method 1: Enlarged exploration(X_1)}
Update the current result (K_p ,K_i ,K_d) using the Eq.(25)
if *Fitness($X_1(t+1)$)<Fitness(X(t))**then***
$X(t)=X_1(t+1)$
if *Fitness($X_1(t+1)$)<Fitness($X_{best}(t)$)**then***
$X_{best}(t)=X_1(t+1)$
end if
end if
else
{Method2: Narrowed Exploration(X_2)}
Update the current result (K_p ,K_i ,K_d) using the Eq.(26)
if *Fitness($X_2(t+1)$)<Fitness(X(t))**then***
$X(t)=X_2(t+1)$
if *Fitness($X_2(t+1)$)<Fitness($X_{best}(t)$)**then***
$X_{best}(t)=X_2(t+1)$
end if
end if
end if
else
if *rand≤0.5 then*
{Method3: Enlarged exploitation(X_3)}
Update the current result (K_p, Ki, Kd) using the Eq.(34)
if *Fitness($X_3(t+1)$)<Fitness(X(t))**then***
$X(t)=X_3(t+1)$
if *Fitness($X_3(t+1)$)<Fitness($X_{best}(t)$)**then***
$X_{best}(t)=X_3(t+1)$
end if
end if
else
{Method4: Narrowed Exploitation(X_4)}
Update the current result (K_p ,K_i , K_d) using the Eq.(35)
if *Fitness($X_4(t+1)$)<Fitness(X(t))**then***
$X(t)=X_4(t+1)$
if *Fitness($X_4(t+1)$)<Fitness($X_{best}(t)$)**then***
$X_{best}(t)=X_4(t+1)$
end if
end if
end if
end if
end for
while(*(The maximum number of iterations has not been reached, or a stop requirement has not been met)*
return *The best result (X_{best})*

5 Results and Discussion

The linear motion system model given in Eq. (3) is initially tuned using the PID auto tuner function available in matlab. The PID auto-tuner function which includes the required robustness in the model provides high peak overshoot and settling time as shown in Fig. 4a and Table 2. The model is then tuned using the FOPID controller using the ninteger tool box in matlab. The five parameters are tuned to enhance the performance against uncertainties in system model, high frequency noise and load disturbances. The five parameters are tuned as follows: $k_p = 0.00048$, $K_i = 0.00017$, $K_D = 0.00021$, $\lambda = -0.5$, $\mu = 0.05$. The FOPID algorithm gives better performance parameters than the classical PID algorithm as shown in Fig. 8b and Table 2. The closed loop stability of the proposed model is verified using the bode plot for PID and FOPID controllers as shown in Fig. 5a and b.

This transfer function given in Eq. (3) is used for optimal tuning of the Servo PID controller used in linear motion systems using a heuristic approach. In this work, a new objective function is presented based on the ITAE, rise time, settling time, and maximum overshoot. The weights for this proposed objective function are calculated using pareto optimal sets of the MOGA algorithm. The parameters used for tuning a linear motion system using the MOGA algorithm are given in Table 3. The MOGA algorithm generates forty-eight data sets of pareto optimal solutions of the PID parameters and their corresponding pareto front sets. The mean, contribution percentage, and weights of the pareto front sets are calculated as given in Table 4. The multi-objective function (J(X)) is formulated by combining four objective functions (ITAE, t_r, t_s, M_p) using the weighted sum function. The novel multi-objective function used for optimal servo PID tuning of the linear motion control system is given in Eq. (39).

$$J(X) = 0.8713 * ITAE + 0.1162 * t_r + 0.0054 * t_s + 0.0071 * M_p \tag{39}$$

The new multi-objective function is tested using the PSO algorithm and its performance parameters are compared with conventional PID error minimizing objective functions. Figure 6. shows the convergence curves of various objective functions. The proposed function shows better convergence compared to the other conventional objective functions. The new cost function outperforms the error minimizing functions such as IAE, ISE, ITAE, and MSE in terms of peak overshoot, settling time, and steady state error, as shown in Table 5 and Fig. 7.The proposed multi-objective function is also tested using the most popular recently developed heuristic algorithms such as WOA, BAT, and AO. The heuristic algorithm parameters initialized for tuning the linear

Table 5 PSO based servo PID tuning of linear motion control system

Objective functions	Servo PID tuning parameters			Performance parameters			
	$K_p \times 10^{-3}$	$K_i \times 10^{-3}$	$K_d \times 10^{-3}$	M_p	T_r	T_s	Ess
IAE	0.88	0.75	0.95	5.61	0.13	3.85	0.007
ISE	0.81	0.45	0.96	4.86	0.12	4.79	0.002
ITAE	0.85	0.45	0.78	4.34	0.15	4.11	0.004
MSE	0.86	0.66	0.96	5.42	0.13	4.09	0.007
J(X)	0.78	0.17	0.53	0.09	0.22	0.35	5.92E−04

Table 6 Heuristic Algorithm parameters for tuning linear motion control system

PSO	WOA	BAT	AO
No. of population = 50 Maximum Iteration = 100 $w_{Max} = 0.5$; $w_{Min} = 0.2$; $c_1 = 0.2$; $c_2 = 0.2$;	Search agents = 50 Maximum Iteration = 50 P random number in [0,1] l random number in [−1,1] p random number in [0,1]	No. of bats = 50 Maximum Iteration = 50 $F_{max} = 1$ $F_{min} = 0$ Loudness, $\alpha = 0.5$ Pulse rate, $r_i = 0.001$ Emission rate update, $\delta = 0.5$	No. of Solution = 50 Maximum Iteration = 50 $\alpha = 0.9$ $\delta = 0.9$

Table 7 WOA based servo PID tuning of linear motion control system

Objective functions	Servo PID tuning parameters			Performance parameters			
	$K_p \times 10^{-3}$	$K_i \times 10^{-3}$	$K_d \times 10^{-3}$	M_p	T_r	T_s	Ess
ITAE	2.59	9.98	9.98	53.27	0.02	0.419	0.004
J(X)	0.74	0.1	0.73	0.423	0.17	1.084	5.18E−04

motion control system are shown in Table 6. The PID factors of the linear motion control system are tuned by the WOA technique using the most popular single objective function ITAE, and a proposed multi-objective function (J(X)). Table 7 shows the performance parameters for the WOA based servo PID tuning using two objective functions. The multi-objective function shows a large reduction in peak overshoot and steady state error with rise and settling time close to the single objective function as shown in Fig. 8.

The BAT algorithm is used to tune the PID values of a linear motion system using a single objective function, ITAE and the proposed objective function, J(X). This algorithm shows a large improvement in peak overshoot and steady state error with respect to multi-objective function as shown in Table 8 and Fig. 9.

Finally, the proposed objective function (J(X)) is tested using the recently developed Aquila optimizer algorithm for a linear motion system, and the results are compared with the function ITAE. The function J(X) improves on the ITAE in peak overshoot, settling time and steady state error, and has a rise time similar to the ITAE as shown in Table 9 and Fig. 10.

Figure 11 and Table 10 shows the comparison results of step output of a linear motion system tuned by the optimization algorithm using error minimizing objective function ITAE. The PSO and AO algorithm provides better result in terms of peak overshoot percentage, while WOA and BAT shows improvement in terms of settling time.

Figure 12 and Table 11 shows the comparison results of step output of a linear motion system tuned by the optimization algorithm using proposed multi-objective function J(X). The comparison of Table 10 and 11 reveal that the new function gives good optimal performance than single objective function and classical PID and FOPID controllers. The hardware validation of the proposed methodology can be done using the hardware set up shown in Fig. 13. It consists of a real time compact RIO (cRIO) FPGA controller, a NI-9502 servo drive module, a kollmorgen servo motor, and a Bosch Rexroth linear motion system. This hardware setup can be programmed in a LabVIEW environment using the LabVIEW soft motion module 18.0.

The PID interactive tuning panel of the servo position control program is tuned using the parameters predicted using the conventional PID controller and the novel cost function based soft tuning algorithms, and the validated results of the performance parameters are shown in Table 12.

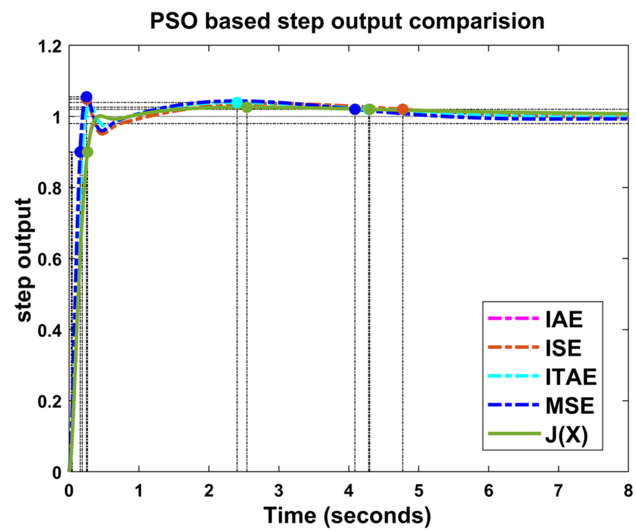


Fig. 7 PSO step output of linear motion control system for various objective functions

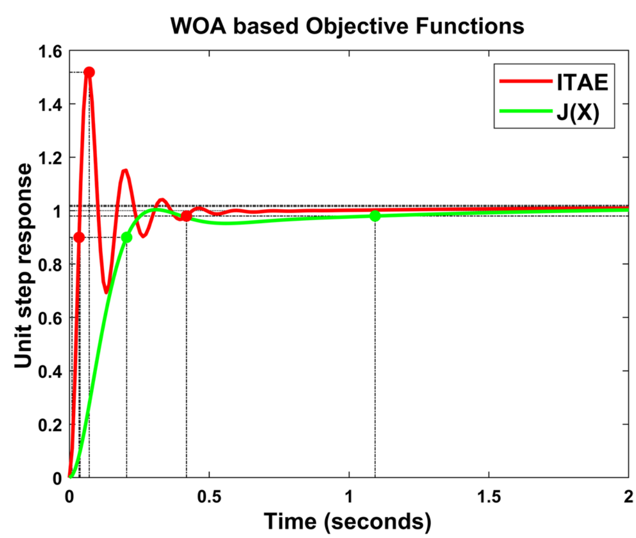


Fig. 8 WOA based step output of linear motion control system for objective functions ITAE and J(X)

Table 8 BAT based servo PID tuning of linear motion control system

Objective functions	Servo PID tuning parameters			Performance parameters			
	$K_p \times 10^{-3}$	$K_i \times 10^{-3}$	$K_d \times 10^{-3}$	M_p	T_r	T_s	Ess
ITAE	3.63	8.79	8.79	51.32	0.03	0.442	0.0029
J(X)	0.92	0.1	0.92	4.881	0.13	0.989	7.98E−04

Table 9 AO based servo PID tuning of linear motion control system

Objective functions	Servo PID tuning parameters			Performance parameters			
	$K_p \times 10^{-3}$	$K_i \times 10^{-3}$	$K_d \times 10^{-3}$	M_p	T_r	T_s	Ess
ITAE	0.87	0.65	0.89	4.79	0.13	3.97	0.006
J(X)	0.88	0.16	0.71	1.73	0.16	0.81	2.10E−03

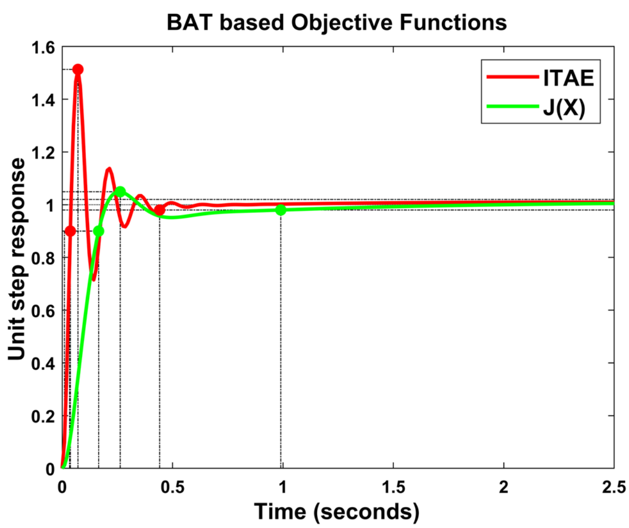


Fig. 9 BAT algorithm based step output of linear motion control system for objective functions ITAE and J(X)

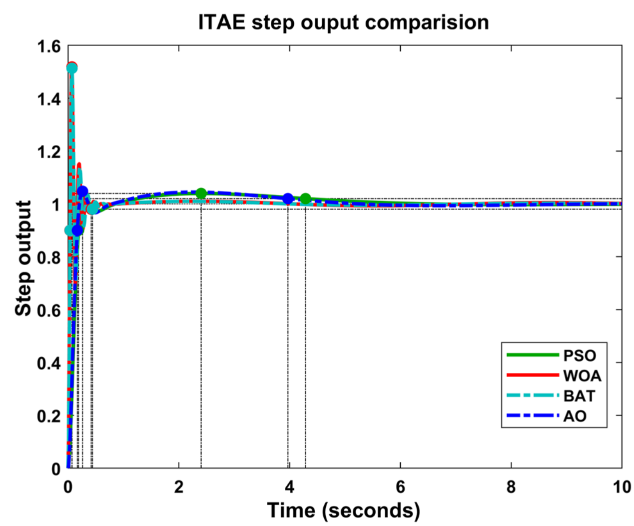


Fig. 11 Performance of optimization algorithm for objective function ITAE

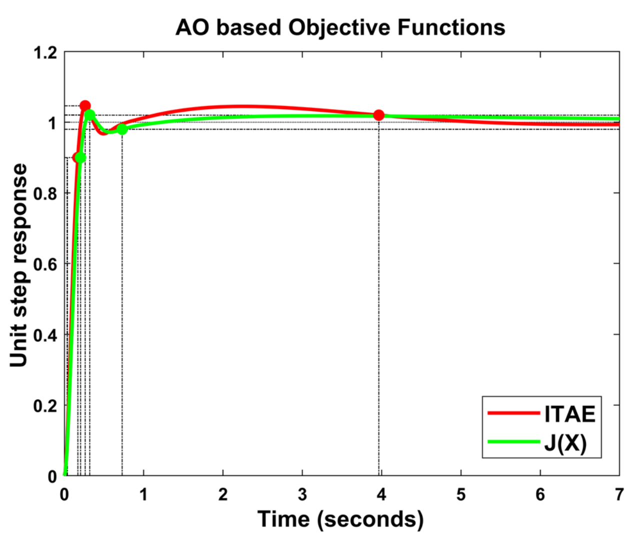


Fig. 10 AO based step output of linear motion control system for objective functions ITAE and J(X)

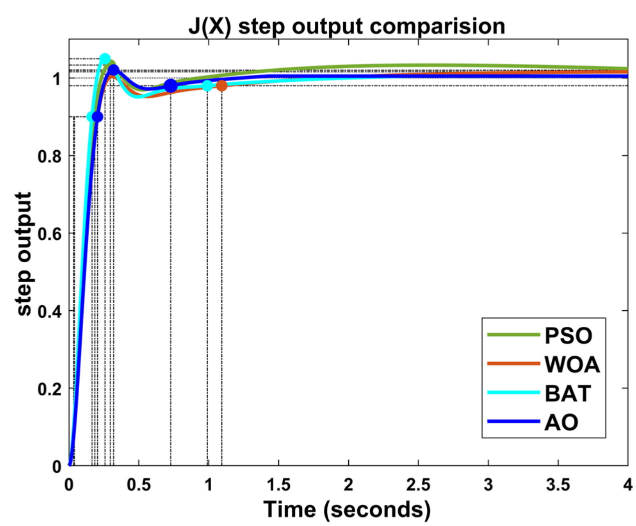


Fig. 12 Effectiveness of optimization algorithm for multi-objective function, J(X)

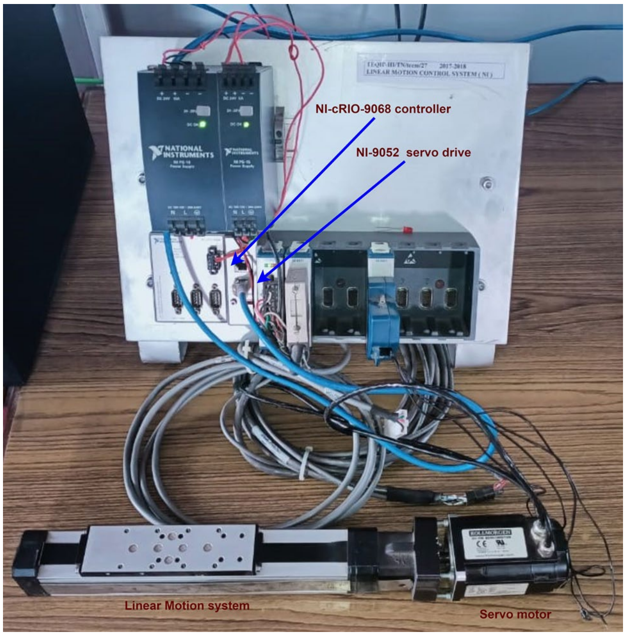


Fig. 13 Hardware setup for validating propose methodology

The unit step plot obtained for various tuning algorithms is shown in the Fig. 14a–f. From Fig. 14 and Table 12, it is concluded that the proposed PID tuning parameters using multi-objective based optimization algorithms provide better performance results for linear motion systems than the conventional PID control tuning algorithms. Table 13 shows the error comparison results of the simulation and the validation done for the linear motion system using the matlab and the LabVIEW tools.

A small variation in validation results is observed compared to the simulation results due to the precision of the pid tuning parameters. Since the matlab model accepts the tuning parameters with more precision than the LabVIEW,

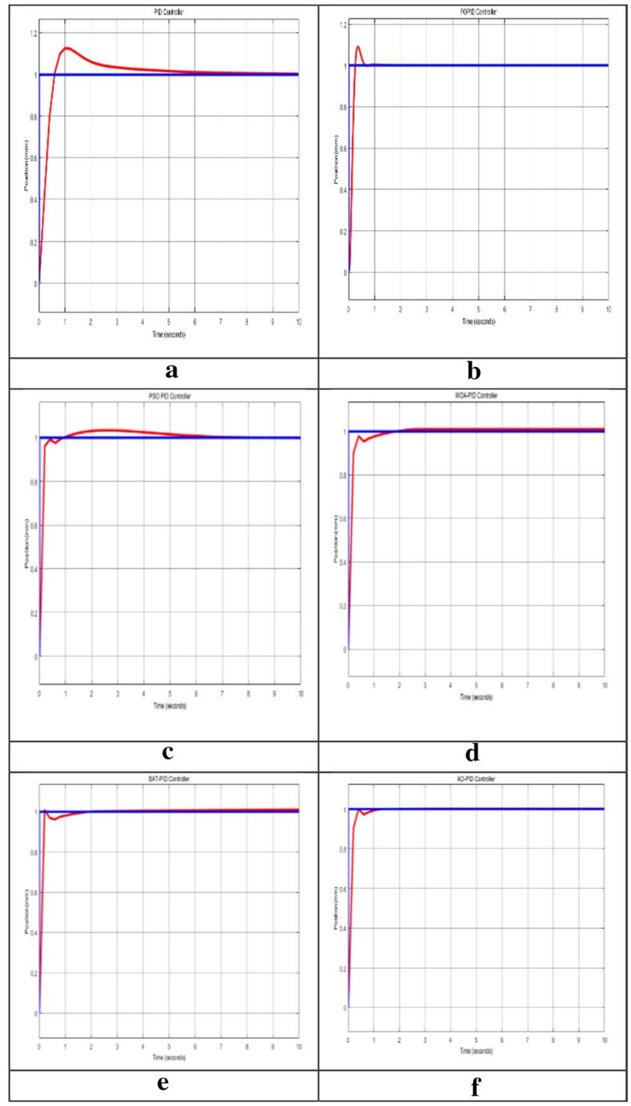


Fig. 14 Validation plots of the linear motion system for **a** Auto tuned PID controller **b** FOPID controller **c** PSO tuned PID **d** WOA tuned PID **e** BAT tuned PID **f** AO tuned PID

Table 10 Performance of optimization algorithm for servo PID tuning of linear motion control system using ITAE

Optimizing algorithm	Servo PID tuning parameters			Performance parameters			
	$K_p \times 10^{-3}$	$K_i \times 10^{-3}$	$K_d \times 10^{-3}$	M_p	T_r	T_s	Ess
PSO	0.85	0.45	0.78	4.34	0.15	4.11	0.004
WOA	2.59	9.98	9.98	53.3	0.02	0.42	0.004
BAT	3.63	8.79	8.79	51.3	0.03	0.44	0.003
AO	0.87	0.65	0.89	4.79	0.13	3.97	0.006

Table 11 Performance of optimization algorithm for servo PID tuning of linear motion control system using multi-objective function J(X)

Optimizing algorithm	Servo PID tuning parameters			Performance parameters			
	$K_p \times 10^{-3}$	$K_i \times 10^{-3}$	$K_d \times 10^{-3}$	M_p	T_r	T_s	Ess
PID auto-tuner	0.59	0.18	0.21	12.6	0.41	4.35	0.009
FOPID	0.48	0.17	0.21	9.1	0.45	1.53	0.004
PSO	0.88	0.38	0.80	1.29	0.19	0.28	1.70E−03
WOA	0.74	0.1	0.73	0.42	0.17	1.08	5.18E−04
BAT	0.92	0.1	0.92	4.88	0.13	0.99	7.98E−04
AO	0.88	0.16	0.71	1.73	0.16	0.81	2.10E−03

the results obtained in simulation are more accurate than the validation. The error difference between simulation and validation for the PID auto tuner is found to be less, and steady state error is found to be better in validation.

The error difference in the FOPID controller is also found to be less in all performance parameters. The PSO algorithm provides higher peak overshoot and settling time than simulation and provides better rise time and steady state error in validation. The WOA algorithm provides better results, except for a small increase in peak overshoot. The performance results of the BAT algorithm are superior in all parameters. The error difference is less in the AO algorithm for peak overshoot and rise time and better in validation for settling time and steady state error. The validation results show that the proposed heuristic algorithm based tuning provides better results than conventional PID tuning.

6 Conclusion

The optimal servo PID control tuning is necessary in order to maintain the positional accuracy in linear motion control systems. The error minimizing functions such as IAE, ITAE, ITSE, ISE, and MSE used by the heuristic algorithms do not satisfy all the performance needs of a linear motion system. A new multi-objective function is presented based on the optimal choice of weights obtained using MOGA techniques for four conflicting objective functions (ITAE, rise time, settling time, maximum overshoot). The proposed function is tested for linear motion control systems using familiar heuristic algorithms such as PSO, WAO, BAT, and the recently developed Aquila optimizer, and the results are compared with the commonly used objective function ITAE. The simulation results show that the proposed objective function shows better performance results in terms of peak overshoot and steady state error compared to the ITAE and also produces results similar to the ITAE in terms of rise and settling time. The proposed algorithm validated in the LabVIEW environment also shows that the heuristic algorithms provide better performance results than the conventional PID controllers.

Table 12 Comparison of the performance parameters with validation results

Optimizing algorithm	Simulation results in matlab environment (A)				Validation results in lab view environment(B)				Error (E = B − A)			
	M_p	T_r	T_s	Ess	M_p	T_r	T_s	Ess	M_p	T_r	T_s	Ess
PID auto-tuner	12.6	0.407	4.35	0.0088	12.761	0.4097	4.363	0.0032	0.161	0.0027	0.013	−0.0056
FOPID	9.0952	0.4461	1.5262	0.0042	9.65	0.4686	1.841	0.0052	0.555	0.0225	0.3148	0.001
PSO	1.2853	0.1871	0.2828	1.70E−03	3.3274	0.1488	4.41	6.96E−4	2.042	−0.0383	4.1272	−0.001
WOA	0.4233	0.1669	1.0836	5.18E−04	1.5946	0.1664	1.093	0.0103	1.171	−0.0005	0.0094	0.0098
BAT	4.8812	0.1316	0.9886	7.98E−04	4.946	0.1311	0.99	0.0014	0.065	−0.0005	0.0014	0.0006
AO	1.7302	0.163	0.8108	2.10E−03	2.065	0.1652	0.7281	2.61E−4	0.335	0.0022	−0.083	−0.0018

Table 13 Error analysis results of simulation in matlab and validation in labview tool

Optimizing algorithm	Servo PID tuning Parameters			Performance parameters				Validation results			
	K_p	K_i	K_d	M_p	T_r	T_s	Ess	M_p	T_r	T_s	Ess
PID auto-tuner	0.00059	0.00018	0.00021	12.6	0.407	4.35	0.0088	12.761	0.4097	4.363	0.0032
FOPID	0.00048	0.00017	0.00021	9.0952	0.4461	1.5262	0.0042	9.65	0.4686	1.841	0.0052
PSO	0.00088	0.00038	0.00080	1.2853	0.1871	0.2828	1.70E−03	3.3274	0.1488	4.41	6.96E−4
WOA	0.00074	0.0001	0.00073	0.4233	0.1669	1.0836	5.18E−04	1.5946	0.1664	1.093	0.0103
BAT	0.00092	0.0001	0.00092	4.8812	0.1316	0.9886	7.98E−04	4.946	0.1311	0.99	0.0014
AO	0.00088	0.00016	0.00071	1.7302	0.163	0.8108	2.10E−03	2.065	0.1652	0.7281	2.61E−4

References

1. Ramesh H, Xavier SAE, Kumar SB (2017) Multiple model filter based position tracking in CNC machines. In: 2017 fourth international conference on signal processing, communication and networking (ICSCN), Chennai, 2017, pp 1–6, https://doi.org/10.1109/ICSCN.2017.8085681
2. Astrom KJ, Hagglund T (2006) Advanced PID control. In: Åström KJ, Hägglund T (eds) ISA-the instrumentation, systems and automation society. **ISBN: 978155617-9426**
3. Shi H, Ma C, Yang J, Zhao L, Mei X, Gong G (2015) Investigation into effect of thermal expansion on thermally induced error of ball screw feed drive system of precision machine tools. Int J Mach Tools Manuf 97:60–71 (**ISSN 0890-6955**)
4. Liu K, Chen F, Zhu T, Wu Y, Sun M (2016) Compensation for spindles axial thermal growth based on temperature variation on vertical machine tools. Adv Mech Eng. https://doi.org/10.1177/1687814016657733
5. Li Y, Zhang J, Su D, Zhou C, Zhao W (2018) Experiment-based thermal behavior research about the feed drive system with linear scale. Adv Mech Eng. https://doi.org/10.1177/168781401881239
6. Hu F, Chen X, Cai N, Lin YJ, Zhang F, Wang H (2018) Error analysis and compensation of an optical linear encoder. IET Sci Meas Technol 12(4):561–566
7. Astrom K et al (2006) "Advanced PID control", "controller design". In: The ISA Society
8. Astrom K, Hgglund T (1995) PID controller: theory, design and tuning. Instrument of Society of America, Pittsburgh
9. Jayachitra A, Vinodha R (2014) "Genetic algorithm based PID controller tuning approach for continuous stirred tank reactor. Hindawi Publ Corp Adv Artif Intell. https://doi.org/10.1155/2014/791230
10. Krohling R-A, Rey J-P (2001) Design of optimal disturbance rejection PID controllers using genetic algorithms. IEEE Trans Evolut Comput 5(1):78–82
11. Kitsios I, Pimenides TG (2001) A genetic algorithm for designing H∞ structured specified controllers. In: Proceedings of the 2001 IEEE international conference on control applications, pp 1196–1201. https://doi.org/10.1109/CCA.2001.974035
12. Mirzal A, Yoshii S, Furukawa M (2006) PID parameters optimization by using genetic algorithm: a study on time-delay systems. In: ISTECS, 2006 Journal, vol 8, pp 34–43
13. Saad MS, Jamaluddin H, Darus IZ (2012) Implementation of PID controller tuning using differential evolution and genetic algorithms. Int J Innov Comput Inf Control 8(11):7761–7779
14. Krohling RA, Rey JP (2001) Design of optimal disturbance rejection PID controllers using genetic algorithms. IEEE Trans Evolut Comput 5(1):78–82
15. Ahmed I, Ikram J, Yousuf M, Badar R, Bukhari SSH, Ro J-S (2021) Performance improvement of multi-rotor axial flux vernier permanent magnet machine by permanent magnet shaping. IEEE Access 9:143188–143197
16. Bilal M, Ikram J, Fida A, Bukhari SSH, Haider N, Ro J-S (2021) Performance improvement of dual stator axial flux spoke type permanent magnet vernier machine. IEEE Access 9:64179–64188
17. Kumar AA, Kumar SG (2018) Application of whale optimization algorithm for tuning of a PID controller for a drilling machine. In: ICAARS 2018, Dec 2018, Coimbatore, India.hal-0231442
18. Alharbi WN, Gomm B (2017) Genetic algorithm optimisation of PID controllers for a multivariable process. IJES 5(1):77–96. https://doi.org/10.39991/ijes.v5il.6692
19. Zahir AAM, Alhady SSN, Othman WAFW, Wahab AAA, Ahmad MF (2020) Objective functions modification of GA optimized PID controller for brushed DC motor. Int J Electr Comput Eng

(IJECE) 10(3):2426–2433. https://doi.org/10.11591/ijece.v10i3.pp2426-2433 (**ISSN: 2088-8708**)
20. Latha K, Rajinikanth V, Surekha PM (2013) PSO-based PID controller design for a class of stable and unstable systems. Int Sch Res Not 2013:11. https://doi.org/10.1155/2013/543607
21. Jaen-Cuellar AY, de J. Romero-Troncoso R, Morales-Velazquez L, Osornio-Rios RA (2013) PID-controller tuning optimization with genetic algorithms in servo systems. Int J Adv Robot Syst 10:324
22. Popov A, Farag A, Werner H (2005) Tuning of a PID controller using a Multi-objective optimization technique applied to a neutralization plant. In: Proceedings of the 44th IEEE conference on decision and control and European control conference 2005, Seville, Spain, December 12–15,2005
23. Ali S, Ikram J, Devereux CP, Bukhari SSH, Khan SA, Khan N, Ro J-S (2021) Reduction of cogging torque in AFPM machine using elliptical-trapezoidal-shaped permanent magnet. Appl Comput Electromagn Soc (ACES) J 36(8):1090–1098
24. Yousuf M, Khan F, Ikram J, Badar R, Bukhari SSH, Ro J-S (2021) Reduction of torque ripples in multi-stack slotless axial flux machine by using right angled trapezoidal permanent magnet. IEEE Access 9:22760–22773
25. Karahan O, Ataslar-ayyildiz B (2019) Application of multi-objective controller to optimal tuning of PID parameters for different process systems using cuckoo search algorithm. Eskisehir Tech Univ J Sci Technol A Appl Sci Eng 20(1):1–16. https://doi.org/10.18038/aubtda.476952
26. Hassan MHA (2018) Genetic algorithm optimization of PID controller for brushed DC motor. In: M Zahir AA, Alhady SS, Othman WA, Ahmad MF (eds) Intelligent manufacturing & mechatronics, lecture notes in mechanical engineering. Springer Nature, Singapore. https://doi.org/10.1007/978-981-10-8788-2_38
27. Aly A (2011) PID parameters optimization using genetic algorithm technique for electrohydraulic servo control system. Intell Control Autom 2:69–76. https://doi.org/10.4236/ica.2011.22008
28. Ruiz-Rojas ED, Vazquez-Gonzalez JL, Alejos-Palomares R, Escudero-Uribe AZ, Mendoza-Vázquez JR (2008) Mathematical model of a linear electric actuator with prosthesis applications. In: 18th international conference on electronics, communications and computers (conielecomp 2008), Puebla, pp 182–186
29. Nise NS (2019) Control systems engineering, 8th edn. Wiley, Hoboken
30. Aloo LA, Kihato PK, Kamau SI (2016) DC servomotor-based antenna positioning control system design using hybrid PID-LQR controller. Eur Int J Sci Technol 5(2):17–31
31. Sahib MA, Ahmed BS (2016) A new multi-objective performance criterion used in PID tuning optimization algorithms. J Adv Res 7(1):125–134
32. Mirjalili S, Lewis A (2016) The whale optimization algorithm. Adv Eng Softw 95:51–67
33. Yang X-S, Gandomi AH (2012) Bat algorithm: a novel approach for global engineering optimization. Eng Comput Int J Comput Aided Eng Softw 29(5):464–483. https://doi.org/10.1108/02644401211235834
34. Abualigah L, Yousri D, Elaziz MA, Ewees AA, Al-qaness MAA, Gandomi AH (2021) Aquila optimizer: a novel meta-heuristic optimization algorithm. Comput Ind Eng. https://doi.org/10.1016/j.cie.2021.107250

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

H. Ramesh he received a bachelor's degree in Electrical and Electronics Engineering from M.S University in 1997 and M.E degree in Applied Electronics from Anna University, Chennai in 2012. He had worked as an electrical Engineer both in India and Saudi Arabia for ten years. Currently, he is an Assistant Professor in the Department of Mechatronics Engineering at Thiagarajar College of Engineering, Madurai. His research interests include electrical Drives, motion control Systems, and industrial Drives. He had published national and international research papers in motion Control applications. Currently he is pursuing his research in industrial servo drives.

S. Arockia Edwin Xavier he received the bachelor's degree in electrical and electronics engineering, the master's degreein power electronics and drives with distinction, and the Ph.D. degree in 1998, 2005, and 2012,respectively, from Anna University, Chennai, India. He has more than 26 papers published inreputed international journals. He is currently working as an Associate Professor inThiagarajar College of Engineering, Madurai, India. His research focuses on applications of power electronicsand electrical drives.