www.dergipark.gov.tr
ISSN:2148-3736

El-Cezerî Journal of Science and Engineering
Vol: 10, No: 2, 2023 (326-348)
DOI: 10.31202/ecjse.1214722

ECJSE

Research Paper

Satellite Tracking Control System Using Optimal Variable Coefficients Controllers Based on Evolutionary Optimization Techniques

Mohamed El-Sayed M. SAKR[1a], Mohamed A. Moustafa HASSAN[1b]

[1]Department of Electrical Power, Faculty of Engineering, Cairo University Giza, Egypt.
[a]mohamed.201920007@eng-st.cu.edu.eg

Received: 12.12.2022 **Accepted:** 27.04.2023

Abstract: Satellite tracking control system is mechanism that redirects the parabolic antenna to the chosen satellite automatically. It perfectly tracks the satellite as it spins across the sky in its orbit. To maintain a continuous communication signal throughout multiple satellite tracking missions, the tracking process must be fast and smooth, with minimal deviations from the target position. Various controller models have been presented over time to address the problem of antenna positioning in satellite systems and to track moveable targets using servomechanism. The purpose of this study is to describe and debate a satellite tracking control system based on a DC servo motor. Particle Swarm Optimization (PSO), Gravitational Search Algorithm with Particle Swarm Optimization (GSA-PSO) and Eagle Strategy with Particle Swarm Optimization (ES-PSO) techniques are proposed for optimal tuning of Proportional-Integral-Derivative (PID), Fractional Order PID (FOPID), Variable Coefficient PID (V-PID) and Variable Coefficient Fractional Order PID (V-FOPID) controllers that applied in satellite tracking control system. For optimal controllers based on optimization techniques, the dynamic performance indices based objective functions are used to compute the performance index. Furthermore, Self-Tuning Fuzzy FOPID (STF-FOPID) is proposed for satellite tracking control system. The system's response is analyzed, and the outcomes of various control strategies are measured and compared to others. The obtained results implies that V-FOPID controller tuned using ES-PSO can precisely trace the desired position with the fastest settling time and free overshoot when compared to other control strategies.

Keywords: Satellite Tracking, Position Control, Nonlinear Controller, Evolutionary Optimization Techniques.

1. Introduction

As a result of developments in satellite technology, satellites have many applications in the current world in the following areas: meteorology, weather forecasting, communications, radio and TV broadcast, navigation, military and space exploration [1]. Receiving and transmitting systems are mounted on a fixed station or mobile station such as a ship, train, car or aircraft. For multiple missions satellite ground station, in order to ensure continuous communication signals, antenna system must be steered in both the azimuth and elevation angles to trace the target satellite. An earth station's tracking system presented in Figure 1, is necessary to perform some jobs involving satellite acquisition, manual tracking, and automatic tracking. For several years DC servo motor-based controllers have been applied in closed loop control systems to position satellite dishes [1]. Various control models have been presented over time to address the problem of antenna positioning in satellite systems and to track moveable targets using servomechanism [2,3]. The suggested controller's goal is to ensure that the system meets the appropriate requirements regarding overshoot, rise time, settling time and steady-state error while retaining the system's high stability, also, at the same time, providing the system with capacity to reject any disturbance and noise [4]. Optimal controllers based on evolutionary optimization approaches are suggested in order to design and

How to cite this article
Sakr M.E.S.M., Hassan M.A.M., "Satellite Tracking Control System Using Optimal Variable Coefficients Controllers Based on Evolutionary Optimization Techniques," *El-Cezerî Journal of Science and Engineering*, 2023, 10 (2), 326-348.
ORCID: [a]0000-0002-4996-7218; [b]0000-0001-6116-6707

perform a two degree of freedom (2DOF) control system to stabilize the azimuth and the elevation angles of the satellite tracking control system.

In this paper, Particle Swarm Optimization (PSO) [5], Gravitational Search Algorithm with Particle Swarm Optimization (GSA-PSO) [6], and Eagle Strategy with Particle Swarm Optimization (ES-PSO) [7], are proposed for optimal tuning of Proportional-Integral-Derivative (PID), Fractional Order PID (FOPID), Variable Coefficient PID (V-PID) and Variable Coefficient Fractional Order PID (V-FOPID) controllers in satellite tracking control system. Furthermore, Self-Tuning Fuzzy FOPID (STF-FOPID) utilized for satellite tracking control system."

2. Satellite Tracking Control System

For multiple missions satellite ground station, the same earth station keeps track of Low Earth Orbit (LEO), Medium Earth Orbit (MEO), High Earth Orbit (HEO) and Geostationary Orbit (GEO) by modifying the azimuth and elevation angles and employing the appropriate sending/receiving frequency bandwidth. It's significant to note that earth station movements to follow satellite applicable to LEO, MEO, and HEO satellites. In the case of GEO satellites, the control methods are frequently utilized only once in order to automatically direct antenna in state performing the process manually.

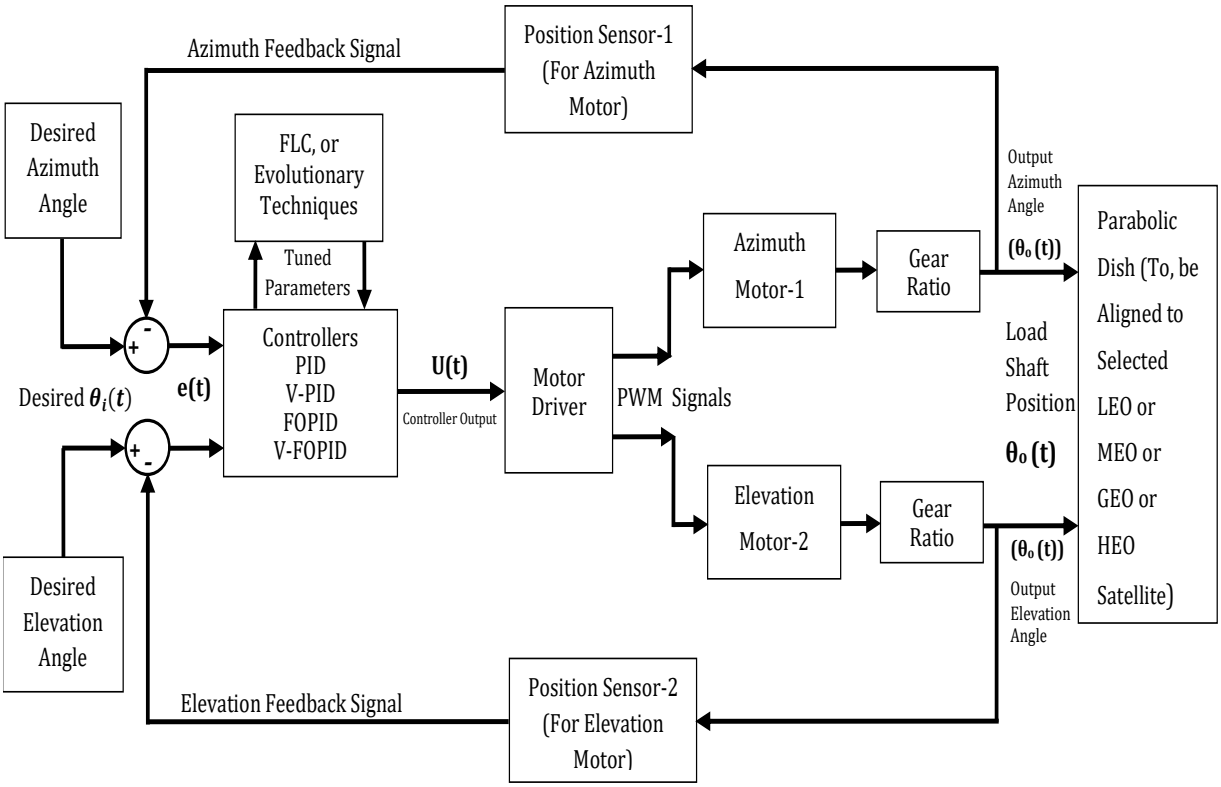

Figure 1. Ground station's tracking control system

Several control methods have been suggested to achieve proper control of the azimuth/elevation angles. The proposed control strategy greatly enhances the system's response by eradicating overshoot, steady-state error as well diminishing rising time and setting time. The tracking control system of earth station depicted in Figure 1 is used to trace desired satellite across both azimuth and elevation angles. The summer has two inputs: one for desired azimuth/elevation positions and one for the current position of the azimuth/elevation motors which is measured by feedback sensors. The position error signal is the difference between these two inputs, and it is delivered to the controller. The error signal is received by the controller, and the corresponding output signal is generated. As a

result, depending on the sign of the error signal, the controller output is given to the motor driver, which develops a corresponding output to rotate the proper motor in either direction. When the intended position is reached, the error signal drops to zero and then the motors stop.

3. Satellite Earth Station Tracking System

Satellite earth station tracking system includes several parts such as: armature-controlled DC servo motor system, mathematical model of satellite tracking system and data of satellite tracking system.

3.1. DC Servo Motor System

The DC servo motor is a torque transducer which converts electrical power into mechanical power. Servo motors are automatic devices which control position or speed in closed-loop control systems. The three more prevalent speed control approaches for DC servo motors are field resistance control, armature voltage control and armature resistance control. We will concentrate on the armature voltage control method since servo motors are less susceptible to variations in field current. Figure 2 demonstrates typical model of servo motor system with gear system [9,10,11].

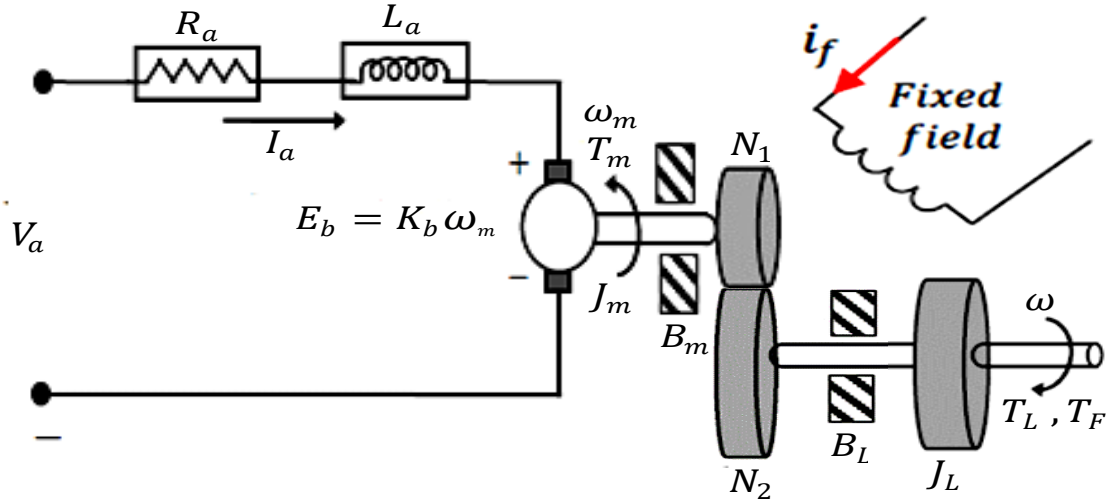

Figure 2. Typical model of DC servo motor system with gear system

3.1.1. Modeling of DC Servo Motor System

Applying Kirchhoff's voltage law to armature loop of an armature voltage control DC motor which is displayed in Figure 2:

$$V_a(\mathrm{t}) = R_a I_a(t) + L_a \frac{d\, I_a(t)}{dt} + E_b(t) \tag{1}$$

E_b: back electromotive force computed using Eqs. (2):

$$E_b = K_b \frac{d\theta}{dt} = K_b \omega(\mathrm{t}) \tag{2}$$

Motor torque equation presented by Eq. (3):

$$T_m = K_t\ I_a(t) \tag{3}$$

Based on the concept that the produced torque must be equal and opposing torques owing to friction, inertia and load at any instant of time, Eq. (4) is determined.

$$T_m(t) = T_F(t) + B_{eq}\omega(\mathrm{t}) + T_L(t) + J_{eq}\frac{d\omega(\mathrm{t})}{dt} \tag{4}$$

where:

$$J_{eq} = J_m + \left(\frac{N_1}{N_2}\right)^2 J_L \tag{5}$$

$$B_{eq} = B_m + \left(\frac{N_1}{N_2}\right)^2 B_L \tag{6}$$

Let $T_d(t)$ as in Eq. (7), to get Eq. (8):

$$T_d(t) = T_F(t) + T_L(t) \tag{7}$$

$$T_m(t) = T_d(t) + B_{eq}\omega(\mathrm{t}) + J_{eq}\frac{d\omega(\mathrm{t})}{dt} \tag{8}$$

When Laplace transformation is performed to both sides of the fundamental equations of the DC servo motor model Eqs. (1), (2), (3) and (8), the following results are obtained:

$$V_a(\mathrm{s}) - E_b(\mathrm{s}) = (L_a S + R_a) I_a(s) \tag{9}$$

$$E_b = K_b\omega(\mathrm{s}) \tag{10}$$

$$T_m = K_t I_a(s) \tag{11}$$

$$T_m - T_d(\mathrm{s}) = (J_{eq}\,\mathrm{s} + B_{eq}\,\mathrm{s})\,\omega(\mathrm{s}) \tag{12}$$

$$T_m - T_d(\mathrm{s}) = (J_{eq}\,s^2 + B_{eq}\,\mathrm{s})\,\theta(\mathrm{s}) \tag{13}$$

$$K_t I_a(s) - T_d(\mathrm{s}) = (J_{eq}\,s^2 + B_{eq}\,\mathrm{s})\,\theta(\mathrm{s}) \tag{14}$$

$$K_t\left(\frac{V_a(s) - E_b(s)}{L_a\,\mathrm{s} + R_a}\right) - T_d(\mathrm{s}) = (J_{eq}\,s^2 + B_{eq}\,\mathrm{s})\,\theta(\mathrm{s}) \tag{15}$$

3.1.2. Block Diagram of DC Servo System

Figure 3 depicts model of open loop DC servo motor system.

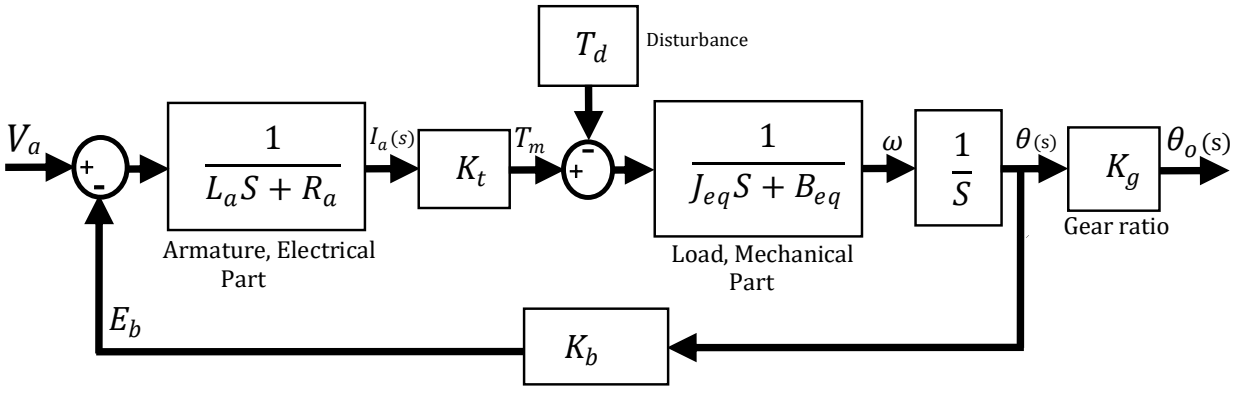

Figure 3. Block diagram model of DC servo motor system with load

3.1.3. Modeling of Satellite Tracking Control System

Schematic diagram of a system for adjusting the azimuth (or elevation) position of a satellite tracking antenna is shown in Fig. 4. The overall system is made up of five subsystems: controller, power

amplifier, motor and load, gear system and feedback position sensor. Each subsystem has a unique transfer function [12].

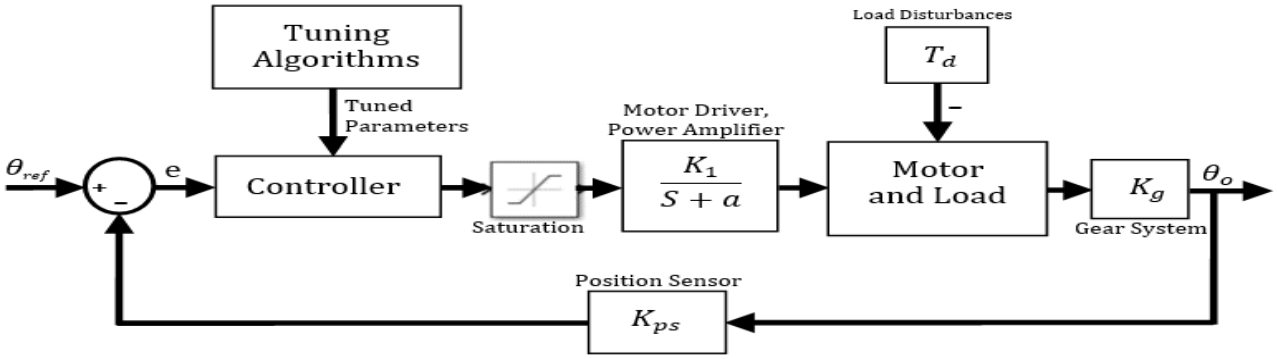

Fig. 4. Block diagram for controlling the azimuth or elevation angles

3.1.4. Data of Tracking Satellite System

Table 1 displays satellite earth station tracking system's parameters and variables. The AKM82T servo motor from Kollmorgen has been selected [13].

Table 1. Parameters and variables of tracking satellite system

Symbol	Description	Value
V_a	Motor armature input voltage	0 to 640 VDC
R_a	Motor armature resistance	0.092 ohm
L_a	Motor armature inductance	2.73 mH
I_a	Motor armature current	
K_b	Back EMF constant of motor	1.6 Vs/rad
E_b	Back electromotive force	V
K_t	Torque constant of motor	1.6 Nm/A
J_m	Moment of inertia of motor	1.72 kg.m2
J_L	Moment of inertia of load	827 kg.m2
B_m	Viscous friction coefficient of motor	0.35 Nms. /rad
B_L	Viscous friction coefficient of load	1.5 Nms. /rad
J_{eq}	Equivalent moment of inertia	1.72 Nms. /rad
B_{eq}	Equivalent viscous coefficient	0.35 Nms. /rad
T_L	Torque due to load static frication	2.3 Nm.
T_F	Torque due to motor static frication	1.7 Nm.
K_1	Power amplifier Gain	100
a	Power amplifier pole.	100
K_g	Gearbox ratio	0.002777
N_1	Gear teeth1	5
N_2	Gear teeth2	1800
K_{ps}	Position sensor gain	1
ω_m , ω	Motor/load shaft angular velocity	
θ	Motor shaft angular position	
θ_o	Load shaft angular position	

4. Control Strategies

This section discusses numerous control techniques used in satellite position control issues, such as PID, V-PID, FOPID, STF-FOPID and V-FOPID.

4.1. Conventional PID Controller

PID controllers were applied in numerous control systems due to its easy design, dependable performance and relatively inexpensive cost [14]. The composition of a PID controller is depicted in Figure 5, which contains proportional (K_P), integral (K_I) and derivative (K_D) gains. The objective of proportional control is to reduce rising and settling times, whereas the role of integral control is to erase the system's steady-state error. The derivative control is frequently applied to enhance closed-loop system's transient response. The PID controller's expressions are listed below [15]:

Eq. (16) presents standard PID controller's output U(t):

$$U(t) = K_P e(t) + K_I \int_0^t e(t) + K_D \left(\frac{N \frac{de(t)}{dt}}{\frac{de(t)}{dt} + N} \right) \tag{16}$$

where N is derivative filter coefficients used as tunning parameters to improve the system performance and increased control flexibility in construction of PID controllers.

Eq. (17) shows transfer function of PID controller:

$$G_{PID}(S) = K_P + K_I \frac{1}{S} + K_D \left(\frac{NS}{S+N} \right) \tag{17}$$

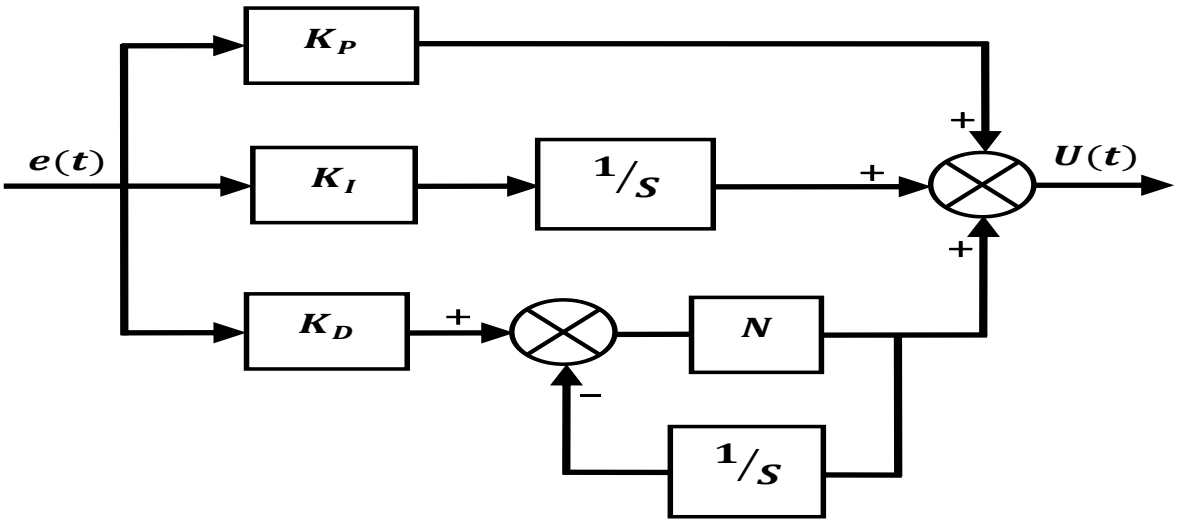

Figure 5. Structure of PID controller

4.2. Fractional Order PID Controller

In 1999, Igor Podlubny developed Fractional Order PID controller labeled by $(PI^{\lambda}D^{\mu})$ [16]. The FOPID controller is designed based on fractional order calculus. In a comparison to PID controller, FOPID controller depicted in Figure 6 is s differentiated by two extra control parameters in which the

orders of the integral part λ and derivative part μ are non-integer. With addition of two degrees of freedom, FOPID gives increased control flexibility in construction of PID controllers and allows for better adaptation of the control system's dynamics [17]. The controller's integral and differential links are drastically impacted by λ and μ . The λ and μ ranges are determined by depending on the system's order. λ significantly effects on system's steady state precision and settling time, whereas μ drastically influences closed loop system's overshoot and stability.

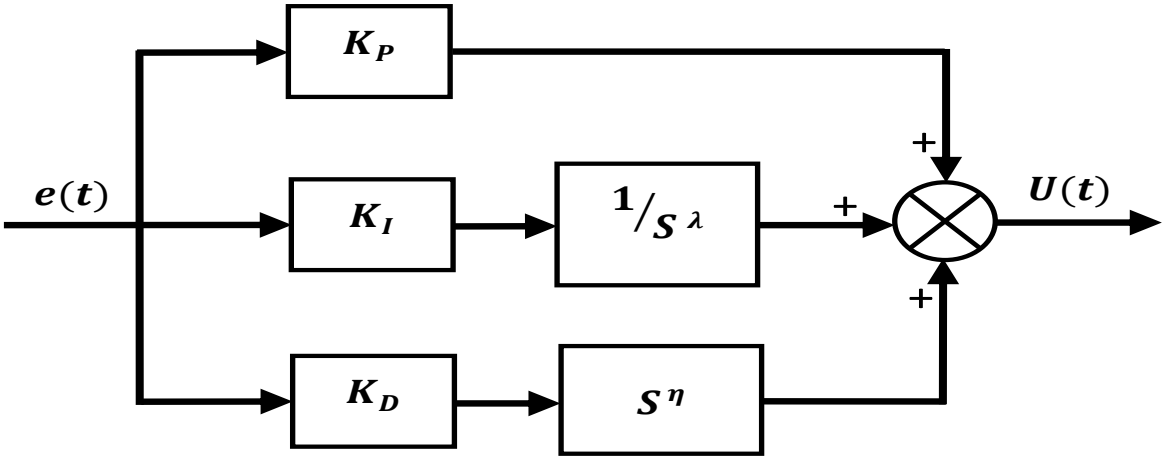

Figure 6. Structure of FOPID controller

The FOPID controller's expressions are listed below:

Output of $(PI^{\lambda}D^{\mu})$ controller is provided by Eq. (18):

$$U(t) = K_P e(t) + K_I D_t^{-\lambda} e(t) + K_D D_t^{\mu} e(t) \tag{18}$$

FOPID's transfer function is also denoted by Eq. (19):

$$G_{FOPID}(S) = K_P + K_I s^{-\lambda} + K_D s^{\mu} \tag{19}$$

4.3. Self-Tuning Fuzzy FOPID Controller

Self-Tuning Fuzzy FOPID (STF-FOPID) controller represented in Figure 7 is an enhanced version of FOPID that is an amalgamation of fuzzy control theory and the FOPID controller. In comparison to FOPID, STF-FOPID is a more robust controller that does not vary its response to disturbance release and has a less settling time and lower overshoot. STF-FOPID is an adaptive controller that can automatically tune FOPID controller parameters online as well as deal with unknown dynamics and non-linearity in system to achieve required response [15].

In the proposed control approach, the ‘Mamdani-type’ fuzzy logic control utilizes the error (e) and the rate of the change of error (de/dt) as inputs and compute three outputs K_P, K_I and K_D. The dual inputs and three outputs are well-defined using fuzzy sets, and each fuzzy set defined by seven fuzzy subsets {NB, NM, NS, ZO, PS, PM, PB}. The seven subsets are defined by generalized bell membership function and referring to negative big, negative medium, negative small, zero, positive small, positive medium, positive big, respectively. Further, λ and μ values adjusted manually. Fuzzy rules are employed to fire outputs are displayed in Table 2, Table 3 and Table 4 [18,19].

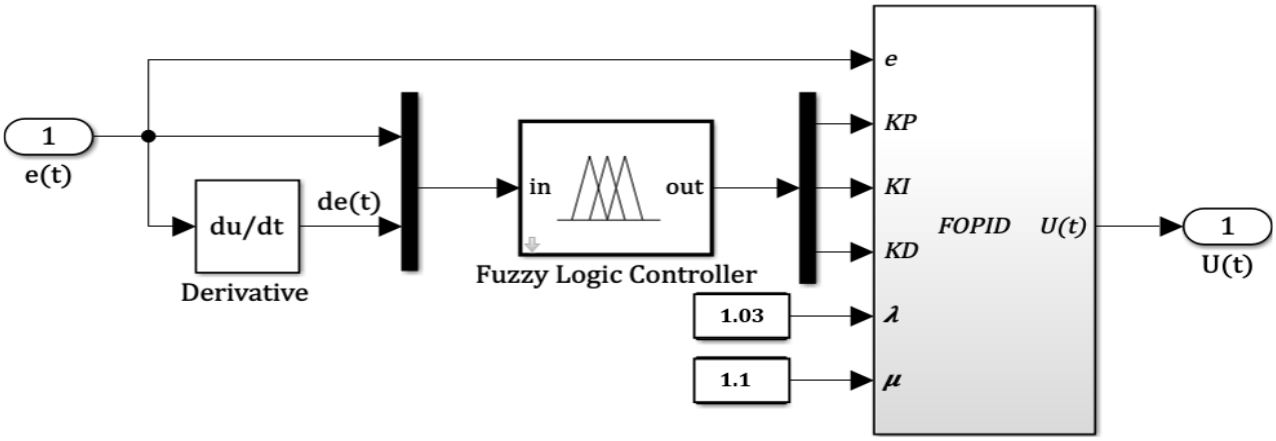

Figure 7. Structure of (STF-FOPID) controller

Table 2. Fuzzy rules of K_P

e	de	NB	NM	NS	ZO	PS	PM	PB
NB		PB	PB	PM	PM	PS	ZO	ZO
NM		PB	PB	PM	PS	PS	ZO	NS
NS		PM	PM	PM	PS	ZO	NS	NS
ZO		PM	PM	PS	ZO	NS	NM	NM
PS		PS	PS	ZO	NS	NS	NM	NM
PM		PS	ZO	NS	NM	NM	NM	NB
PB		ZO	ZO	NM	NM	NM	NB	NB

Table 3. Fuzzy rules of K_I

e	de	NB	NM	NS	ZO	PS	PM	PB
NB		NB	NB	NM	NM	NS	ZO	ZO
NM		NB	NB	NM	NS	NS	ZO	ZO
NS		NB	NM	NS	NS	ZO	PS	PS
ZO		NM	NM	NS	ZO	PS	PM	PM
PS		NM	NS	ZO	PS	PS	PM	PB
PM		ZO	ZO	PS	PS	PM	PB	PB
PB		ZO	ZO	PS	PM	PM	PB	PB

Table 4. Fuzzy rules of K_D

e	de	NB	NM	NS	ZO	PS	PM	PB
NB		PS	NS	NB	NB	NB	NM	PS
NM		PS	NS	NB	NM	NM	NS	ZO
NS		ZO	NS	NM	NM	NS	NS	ZO
ZO		ZO	NS	NS	NS	NS	NS	ZO
PS		ZO	ZO	ZO	ZO	ZO	ZO	ZO
PM		PB	NS	PS	PS	PS	PS	PB
PB		PB	PM	PM	PM	PS	PS	PB

4.4. Variable Coefficient PID Controller

Variable coefficient PID controller (V-PID), known as non-linear PID controller, is distinguished by variable coefficient gains denoted by K_P', K_I' and K_D' . These gains are determined by current error values. As a result, the values of K_P', K_I' and K_D' gains vary as a function of system error [20].

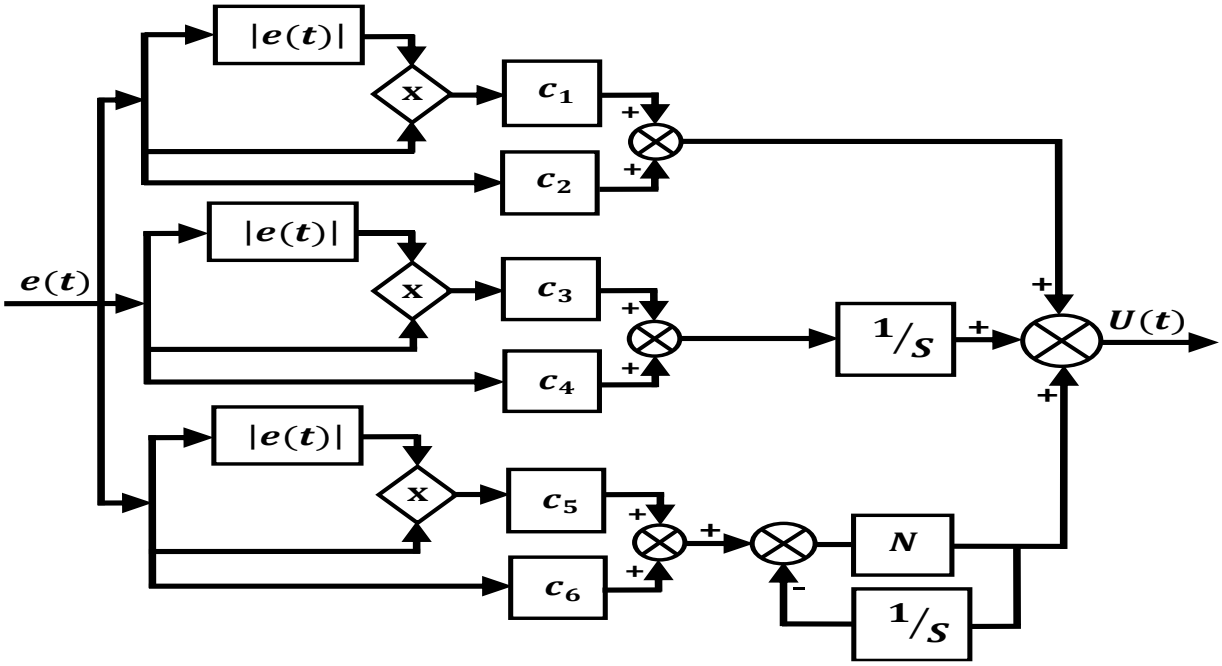

Figure 8. Structure of V-PID controller

The V-PID controller demonstrated in Figure 8 enhances the transient state while maintaining the steady-state response unaffected. This can can be performed by replacing constant gains of classical PID controller with variables gains. The expressions of V-PID are listed below [15]:

Eq. (20) defines the output of the variable PID controller:

$$U(t) = K_P' e(t) + K_I' \int_0^t e(t) + K_D' \left(\frac{N \frac{de(t)}{dt}}{\frac{de(t)}{dt} + N} \right) \tag{20}$$

Eq. (21) also gives the V-PID's transfer function:

$$G_{V-PID}(S) = K_P' + K_I' \frac{1}{S} + K_D' \left(\frac{NS}{S+N} \right) \tag{21}$$

Where N is derivative filter coefficients used as tunning parameters to improve system performance. Eqs. (22), (23) and (24) define variable coefficients K_P', K_I' and K_D' of V-PID controller as follow:

$$K_P' = c_1|e(t)| + c_2 \tag{22}$$
$$K_I' = c_3|e(t)| + c_4 \tag{23}$$
$$K_D' = c_5|e(t)| + c_6 \tag{24}$$

where $|e(t)|$ system's absolute error, also c_1 through c_6 and N are new tuning parameters.

4.5. Variable Coefficient FOPID Controller

The FOPID controller described in Eqs. (18) and (19) is distinguished by constant coefficient that are proportional gain K_P, integral gain K_I and derivative gain K_D. Regardless of system error, these parameters maintain the same value throughout the whole operation. As a consequence, it is futile to attempt to separate transient and steady-state responses independently. The V-FOPID is demonstrated by Eqs. (25) and (26) is differentiated by variable coefficient gains symbolized by K_P', K_I' and K_D' is commonly known as a non-linear FOPID controller. The V-FOPID controller is capable of improving the transient state while preserving the steady-state response unaltered. These gains values are dependent on the current values of the system's exact error. As a result, the controller's gains vary in relation to the system's exact error.

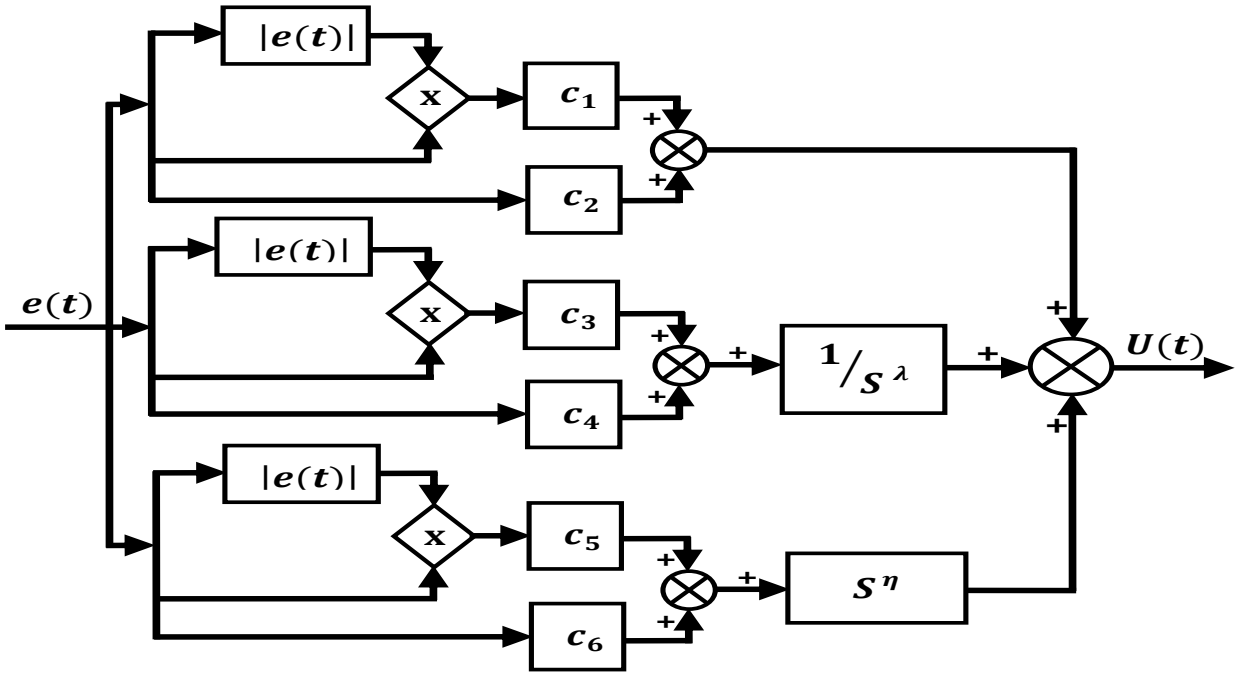

Figure 9. Structure of (V-FOPID) controller

These variable coefficients K_P', K_I' and K_D' are defined as in the Eqs. (22), (23) and (24). This novel V-FOPID technique allows to enhance system's responses in both transient and steady-state conditions. In addition to providing far more freedom in constructing FOPID controllers, it also allows to fine-tune dynamics of control system [20]. Expressions of the V-FOPID controller shown in Figure 9 are stated below [15]:

The V-FOPID controller's output is described by Eq. (25).

$$U(t) = K_P' e(t) + K_I' D_t^{-\lambda} e(t) + K_D' D_t^{\mu} e(t) \tag{25}$$

The V-FOPID controller's transfer function is represented by Eq. (26) as follows:

$$G_{V-FOPID}(S) = K_P' + K_I' s^{-\lambda} + K_D' s^{\mu} \tag{26}$$

5. Evolutionary Optimization Techniques

Many evolutionary methods have been reported by various researchers for optimizing PID controller parameters in a wide range of applications. In this research, Particle Swarm Optimization (PSO) [5],

Gravitational Search Algorithm with Particle Swarm Optimization (GSA-PSO) [6], and Eagle Strategy with Particle Swarm Optimization (ES-PSO) [7] are considered for optimal tuning of PID, FOPID and V-FOPID controllers in satellite control system. Performance index is calculated using dynamic performance indices based objective functions [8]. The fundamentals of these suggested performance index and evolutionary algorithms are demonstrated as follows:

5.1. Formulation of Objective Function

The most essential step in optimum controller design is selecting the most appropriate objective function. Time domain objective functions are divided into two main categories: Integral based objective functions and dynamic performance indices based objective functions [8].For controller optimal design this study employed dynamic performance indices based objective function (J). Also, it known as multi objective function given by Eq. (27), [15].

$$J = \int_0^\infty \delta_1 |e(t)|\, dt + \delta_2 * OS\% + \delta_3 * (t_s - t_r) \tag{27}$$

$$\text{where} \quad 0.1 \leq \delta_1, \delta_2, \delta_3 \leq 0.5 \tag{28}$$

$\delta_1,\ \delta_2, \delta_3$ are weighting factors used to set the significance of performance criteria to others.

Let:

$$\delta_1 = 0.3, \quad \delta_2 = 0.5, \quad \delta_3 = 0.2 \tag{29}$$

where:

$e(t)$: Error signal in time domain $O_.$: Overshoot percentage

t_r : Rise time t_s : Settling time

5.2. Particle Swarm Optimization (PSO)

Particle Swarm Optimization (PSO) is a modern meta-heuristic population-based stochastic optimization technique that was initially developed by Kennedy and Eberhart in 1995 [5]. It utilizes evolutionary search to find near optimum or optimum solutions. PSO is evolutionary-based search process in which particles position and velocity are initialized in a uniform random manner throughout the search space. During every iteration, each particle changes its position (X) based on both its own experience $(Pbest)$, and experience of global neighborhood particles $(Gbest)$. In PSO, each particle is linked to and can learn from every other particle in the swarm [21]. Velocity and position of each particle in the swarm updated using Eqs. (30) and (31) as follows:

$$V_i^{(Iter+1)} = \alpha * V_i^{(Iter)} + C_1 r_1 \left(Pbest_i^{(Iter)} - X_i^{(Iter)}\right) + C_2 r_2 \left(Gbest_i^{(Iter)} - X_i^{(Iter)}\right) \tag{30}$$

$$X_i^{(Iter+1)} = X_i^{(Iter)} + V_i^{(Iter+1)} \tag{31}$$

where:

n : Number of particles in swarm (i =1: n).

$Iter$: Iteration number.

$V_i^{(Iter)}$: Velocity of particle i at current iteration.

$X_i^{(Iter)}$: Position of particle i at current iteration.

$Pbest_i^{(Iter)}$: Personal best position of particle i at current iteration.

$Gbest_i^{(Iter)}$: Global best position in the swarm at current iteration.

α : Inertia weight factor within the range [0,1].

C_1 , C_2 : Acceleration coefficients usually are equal 2.

r_1, r_2 : Random numbers within the range [0,1].
$V_i^{(Iter+1)}$: Updated velocity of particle (i) at next iteration.
$X_i^{(Iter+1)}$: Updated position of particle (i) at next iteration.

5.3. Gravitational Search Algorithm with Particle Swarm Optimization

A novel hybrid population-based algorithm is established by combining Particle Swarm Optimization (PSO) with Gravitational Search Algorithm (GSA) which is called Gravitational Search Algorithm with Particle Swarm Optimization (GSA-PSO) [6]. The primary concept behind GSA-PSO is to combine the PSO's global search $Gbest$ with the GSA's local search capability. Agents near optimal solution attempt to attract additional agents exploring the search space. When all agents do seem to be relatively close to the optimal solution, they start moving extremely slowly. In this scenario, $Gbest$ greatly helps them to explore searching area globally. The GSA-PSO uses memory $Gbest$ to keep the best solution identified thus far, making it accessible at all times for every agent, [6]. The abilities of global search and local search are adjusted by altering weighting factors C_1' and C_2'. To combine these two algorithms together, Eq. (40) is proposed [6,15].

5.3.1. Working Mechanism of GSA-PSO

GSA-PSO algorithm described in the steps listed below [6,15]:

Step 1. Load System Parameters

Import objective function, parameters of antenna system, initial condition and parameters limits.

Step 2. Generate Initial Population

Consider a system containing ‘n’ agents. Initially, all agents in the search space are randomly generated. Each agent is viewed as a promising solution.

Step 3. Evaluate Fitness of Each Agent and Update Best Fitness at Current Iteration

while $(Iter < MaxIter), do$
Evaluate: $fitness_i^{(Iter)} = fitness(X_i^{(iter)})$, and
$fitness(Gbest_i^{(Iter)}) = min\left(fitness(X_i^{(Iter)})\right)$

Step 4. Update GSA-PSO Algorithm Parameters

After each iteration, the gravitational constant, best fitness, worst fitness and inertia mass must be updated as follows: gravitational constant $G^{(Iter)}$ is a function of initial value of gravitational constant Go, descending coefficient σ, and iteration number $Iter$. $G^{(Iter)}$is calculated by Eq. (32):

$$G^{(Iter)} = G_0 * e^{(-\sigma * Iter/MaxIter)} \tag{32}$$

Inertia mass of each agent M_i, is determined by Eq. (34):

$$m_i^{(Iter)} = \frac{fitness_i^{(Iter)} - worst^{(Iter)}}{best^{(Iter)} - worst^{(Iter)}} \tag{33}$$

$$M_i^{(Iter)} = \frac{m_i^{(Iter)}}{\sum_{j=1}^{N} m_j^{(Iter)}} \tag{34}$$

where:

$fitness_i^{(Iter)}$: Fitness value for agent i at current iteration.
$worst^{(Iter)}$: Worst fitness value at current population at current iteration.
$best^{(Iter)}$: Best fitness value at current population.
$m_i^{(Iter)}$: Mass of agent i at current iteration.
$M_i^{(Iter)}$: The agent's inertial mass at current iteration.

For minimization problem, best fitness value $best^{(Iter)}$ and worst fitness value $worst^{(Iter)}$ fitness values defined as in Eqs. (35) and (36):

$$best^{(Iter)} = min\ fitness_i^{(Iter)} \quad , i \in [1, \mathrm{n}] \tag{35}$$
$$worst^{(Iter)} = max\ fitness_i^{(Iter)} \quad , i \in [1, \mathrm{n}] \tag{36}$$

The global best position of agents $Gbest_i^{(Iter)}$ is updated using:

$$\boldsymbol{If} \quad fitness(Gbest_i^{Iter+1}) < fitness(Gbest_i^{Iter})$$
$$Gbest_i = Gbest_i^{Iter+1}, \quad \boldsymbol{else} \quad Gbest_i = Gbest_i^{Iter}$$

Step 5. Calculate Gravitational Force and Total Force

Gravitational force, F_{ij} , acting on agent i from agent j at current iteration is computed by Eq. (37).

$$F_{ij}^{Iter} = G^{(Iter)} \frac{M_{pi}^{(Iter)} * M_{aj}^{(Iter)}}{R_{ij}^{(Iter)} + \varepsilon} \left(X_j^{(Iter)} - X_i^{(Iter)} \right) \tag{37}$$

Total Force F_i^{Iter} acting on agent i among all agents is calculated with Eq. (38).

$$F_i^{(Iter)} = \sum_{j=1, j \neq i}^{n} rand_j\ F_{ij}^{(Iter)} \tag{38}$$

where:

R_{ij}^{Iter} : Euclidian distance between two agents i and j.
X_i^{Iter} : Position of agent i at current iteration.
X_j^{Iter} : Position of agent j at current iteration.
$\mathrm{rand_j}$: Random number in the interval [0,1].
ε : Small constant number.
n : Population size.

Step 6. Calculate the Acceleration of Agents

Acceleration of each agent $ac_i^{(Iter)}$ is computed by Eq. (39).

$$ac_i^{(Iter)} = \frac{F_i^{(Iter)}}{M_{ii}^{(Iter)}} \tag{39}$$

$ac_i^{(Iter)}$: Acceleration of the agent i at current iteration, where i =1: n.

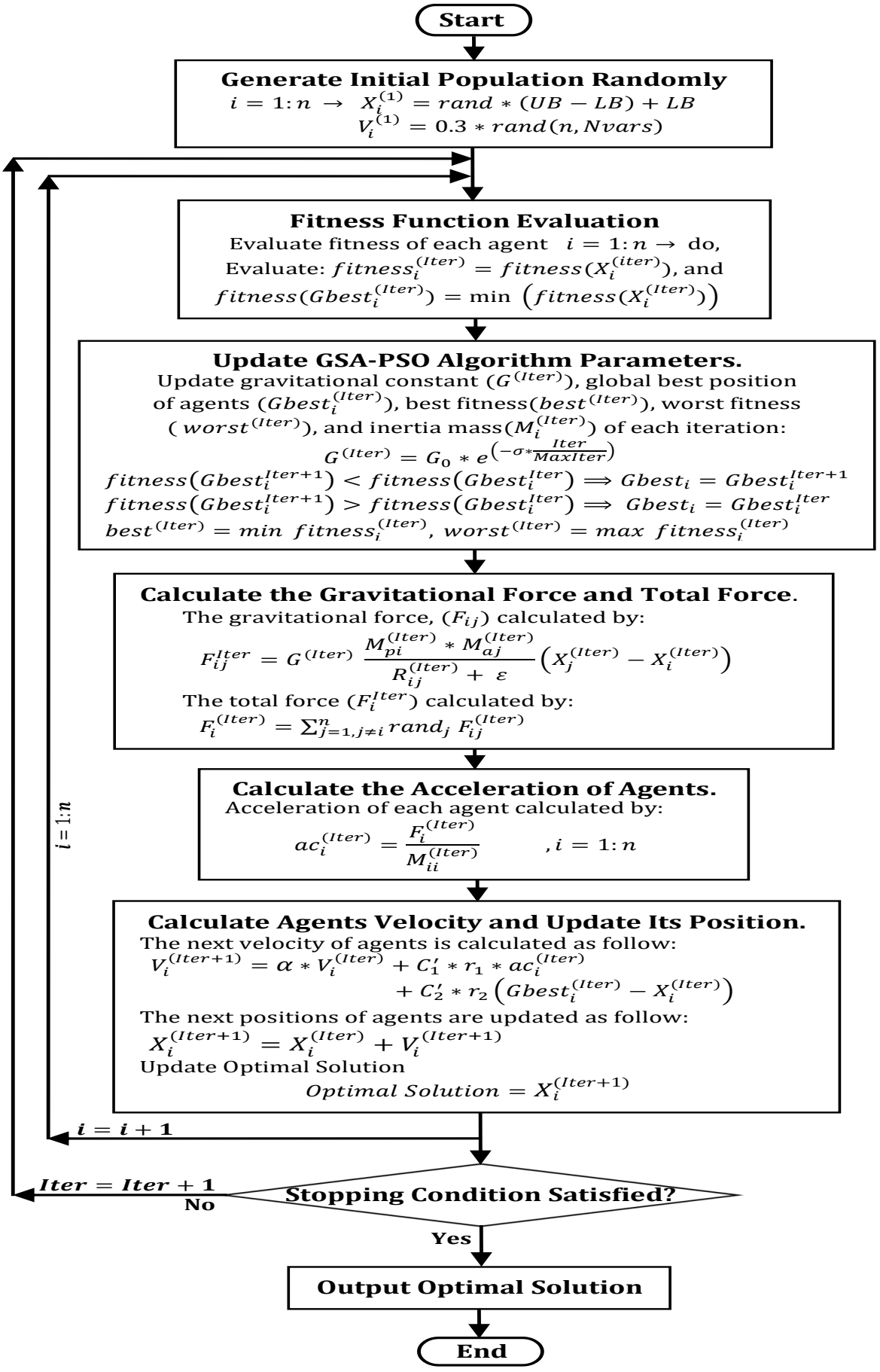

Figure 10. Flowchart of the GSA-PSO algorithm

Step 7. (Calculate Agent Velocity and Update Its Position)

Next velocity of each agent is determined by Eq. (40):

$$V_i^{(Iter+1)} = \alpha * V_i^{(Iter)} + C_1' * r_1 * ac_i^{(Iter)} + C_2' * r_2\left(Gbest_i^{(Iter)} - X_i^{(Iter)}\right) \quad (40)$$

Next position of every agent is updated using Eq. (41):

$$X_i^{(Iter+1)} = X_i^{(Iter)} + V_i^{(Iter+1)} \quad (41)$$

where:

$Gbest_i^{(Iter)}$: Global best solution at current iteration.
A : Weighting function takes as rand (0,1).
C_1', C_2' : Weighting factors.

Step 8. (Stopping criterion)

Check stopping criterion:

If $(Iter = MaxIter)$ go to ***Step*** 9.
else go to ***Step 3.***

Step 9. Output Optimal Solution

Ultimately, if GSA-PSO fulfills end criterion, it is terminated, and the most effective agents printed. The flowchart of GSA-PSO algorithm which summarizes strategy steps is depicted in Figure 10, [15].

5.3.2. Eagle Strategy with Particle Swarm Optimization

Eagle Strategy with Particle Swarm Optimization(ES-PSO) is a metaheuristic technique that searches iteratively for the optimal solution. ES-PSO is a two-stage strategy comprised of global search stage and local search stage [7]. Initially, ES-PSO explores the search space globally using Lévy flight walks; once a promising solution is found, it switches to the local search stage to execute an intensive local search using the PSO algorithm. The global search and local search stages immediately start to iterate until the criterion is met. In fact, several algorithms can be applied at the global search and local search stages. ES-PSO combines advantages of these various algorithms to achieve superior outcomes. Eq. (42) defines Lévy distribution [22].

$$L(\gamma) = \frac{\Gamma(\lambda)\sin(\pi\lambda/2)}{\lambda\ \gamma^{1+\lambda}} \quad (42)$$

where:

$L(\gamma)$: Lévy distribution function. $\Gamma(\lambda)$: Standard gamma function.
λ : Gamma function parameter. γ : Step length.
$\lambda = 2$: Lévy walks are transformed into the cauchy distribution.
$\lambda = 3$: Lévy walks transform into brownian motion.

5.3.3. Working Mechanism of ES-PSO

The suggested ES-PSO method's steps are as follow [15]:

Step 1. Load System Parameters

Import objective function, parameters of antenna system, initial condition and parameters limits.

Step 2. Generate Initial Population Randomly

Initial particles are generated randomly, then their initial velocities are computed.

Step 3. Global Search Stage

$\boldsymbol{while} \quad (Iter < MaxIter), do$

Perform random global search via Lévy Flight by Eq. (43).

$$X_i^{(Iter+1)} = X_i^{(Iter)} + \beta L(\gamma, \lambda) \tag{43}$$

Set $\lambda = 1.5,\ \beta = 1,\ \gamma = 5$, then evaluate fitness and update best fitness at global search stage:

$$\boldsymbol{If} \quad fitness\left(X_i^{(Iter+1)}\right) < fitness\left(X_i^{(Iter)}\right)$$
$$Best\ global\ finess = fitness\left(X_i^{(Iter+1)}\right)$$
$$Gbest_i = X_i^{(Iter+1)}$$
$$\boldsymbol{else}, \quad Best\ global\ \ finess = fitness\left(X_i^{(Iter)}\right)$$
$$Gbest_i = X_i^{(Iter)}$$

Update Best Position in Global Search Stage as follow:

$$Best\ global\ position = Gbest_i$$

Once the promising solution is discovered, then go to ***Step 4.***

Step 4. Switch Between Global and Local Search Stages

The global and local search stages are controlled using switching parameter p as follow:

$\boldsymbol{If} \quad p < rand$ (In this study p is set to 0.2)

$Switch\ to\ local\ search\ stage\ (go\ to\ Step5.)$

$\boldsymbol{else},$

$Switch\ to\ global\ search\ stage\ (go\ to\ Step7.)$

Step 5. Intensive Local Search Stage

Preform intensive local search around promising solution.

Calculate new velocity and position of each particle via Eqs. (44) and (46) as follows:

$$V_i^{(Iter+1)} = \alpha^{(Iter)} * V_i^{(Iter)} + C_1 r_1 (Pbest_i^{Iter} - X_i^{Iter}) + C_2 r_2 (Gbest_i^{Iter} - X_i^{Iter}) \tag{44}$$

where:

$$\alpha^{(Iter)} = \alpha_{max} * \frac{Iter(\alpha_{max} - \alpha_{min})}{MaxIter} \tag{45}$$

The new position of each particle is updated by Eq. (46).

$$X_i^{(Iter+1)} = X_i^{(Iter)} + V_i^{(Iter+1)} \tag{46}$$

Then, evaluate new fitness and update the best fitness in local search stage as follows:

$$\begin{aligned} \textbf{\textit{If}} \quad & fitness\left(X_i^{(Iter+1)}\right) < fitness\left(Pbest_i^{(Iter)}\right) \\ & Best\ local\ fitness = fitness\left(X_i^{(Iter+1)}\right) \\ & Pbest_i = X_i^{(Iter+1)} \\ \textbf{\textit{else}}, & \\ & Best\ local\ \ fitness = fitness\left(Pbest_i^{(Iter)}\right) \\ & Pbest_i = X_i^{(Iter)} \end{aligned}$$

Update Best Position in Local Search Stage:

$$Best\ local\ position = Pbest_i$$

Step 6. Update Global Best Fitness and Global Best Position in the Overall Strategy

Global best fitness and global best position in ES-PSO strategy updated through the following:

$$\begin{aligned} \textbf{\textit{If}} \quad & Best\ local\ fitness < \ Best\ global\ fitness \\ & Global\ fitness = Best\ local\ fitness \\ & Global\ position = Best\ local\ position \\ \textbf{\textit{else}}, & \\ & Global\ fitness = Best\ global\ fitness \\ & Global\ position = Best\ global\ position \end{aligned}$$

Update optimal solution in the strategy:

$$Optimal\ Solution = Global\ position$$

Step 7. (Update Iteration).

$$Iter = Iter + 1$$

Step 8. (Stopping Criterion)

Check stopping criterion.

If $(Iter = MaxIter)$ go to ***Step9.***
else go to ***Step 3.***

Step9. (Output Optimal Solution)

Finally, if the ES-PSO strategy fulfills an end criterion it will be stopped and the most effective particles will be printed.

The flowchart of the ES-PSO methodology, which summarizes the strategy steps are represented in Figure 11, [15].

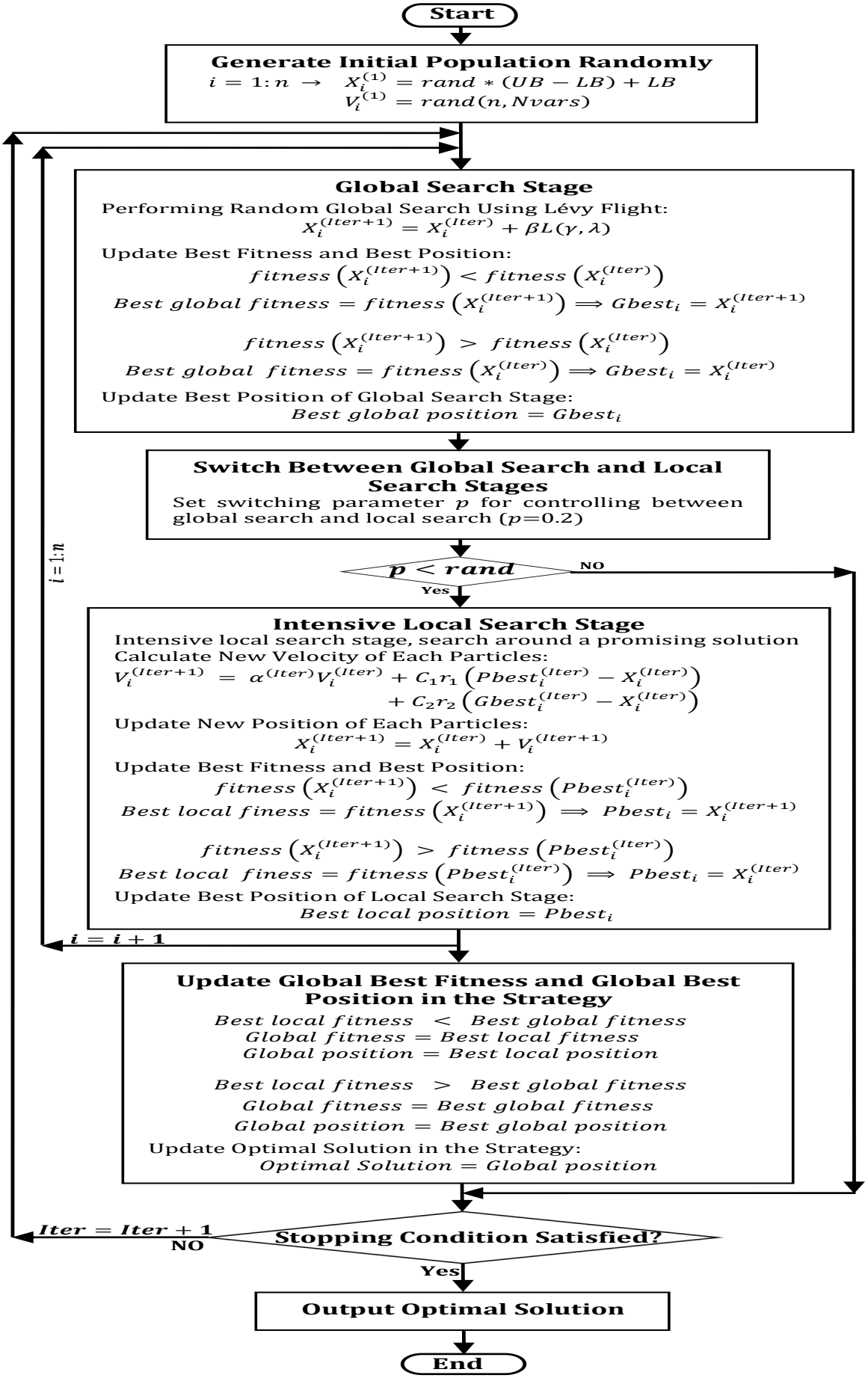

Figure 11. Flowchart of the ES-PSO technique

6. Results and Discussions

The results are performed systematically in MATLAB/Simulink environment. In this research PSO, GSA-PSO and ES-PSO are proposed for optimal tuning of PID, V-PID, FOPID and V-FOPID controllers. The number of population size and maximum iterations are chosen as 25 and 100, respectively for all optimization algorithms. Also, STF-FOPID controller is proposed for satellite control system. Settling time (t_s), rise time (t_r) and overshoot percentage ($OS\%$) are measured by MATLAB/Simulink for 10 sec of simulation running. Additionally, plots of system response and controller output signal is created.

6.1. PID Controller

Table 5. Results obtained from satellite tracking system with optimal PID controller

PID controller with objective function: $\boldsymbol{J = \int_0^\infty \omega_1 \lvert e(t) \rvert\, dt + \omega_2 * OS\% + \omega_3 * (t_s - t_r)}$							
Algorithms	$\boldsymbol{K_P}$	$\boldsymbol{K_I}$	$\boldsymbol{K_D}$	$\boldsymbol{N}$	$\boldsymbol{OS}$ (%)	$\boldsymbol{t_r}$ $\boldsymbol{(s)}$	$\boldsymbol{t_s}$ $\boldsymbol{(s)}$
PSO	1000	1000	293.702	77.772	0	1.3152	3.6598
GSA-PSO	1000	1000	403.192	54.316	0	1.0972	3.1816

6.2. Variable Coefficient PID Controller

Table 6. Results obtained from satellite tracking system with optimal V-PID controller

V-PID controller with: $\boldsymbol{J = \int_0^\infty \omega_1 \lvert e(t) \rvert\, dt + \omega_2 * OS\% + \omega_3 * (t_s - t_r)}$										
Algorithms	$\boldsymbol{C_1}$	$\boldsymbol{C_2}$	$\boldsymbol{C_3}$	$\boldsymbol{C_4}$	$\boldsymbol{C_5}$	$\boldsymbol{C_6}$	$\boldsymbol{N}$	$\boldsymbol{OS}$(%)	$\boldsymbol{t_r}$ $\boldsymbol{(s)}$	$\boldsymbol{t_s}$ $\boldsymbol{(s)}$
GSA-PSO	1000	1000	1000	1000	1	352.7	57.9	0	0.280	1.805
ES-PSO	1	821.5	1000	1000	1	285.2	50.1	1.70	0.342	1.328

6.3. Fractional Order PID Controller

Table 7. Results obtained from satellite tracking system with optimal FOPID controller

FOPID controller with: $\boldsymbol{J = \int_0^\infty \omega_1 \lvert e(t) \rvert\, dt + \omega_2 * OS\% + \omega_3 * (t_s - t_r)}$								
Algorithms	$\boldsymbol{K_P}$	$\boldsymbol{K_I}$	$\boldsymbol{K_D}$	$\boldsymbol{\mu}$	$\boldsymbol{\lambda}$	$\boldsymbol{OS}$ (%)	$\boldsymbol{t_r}$ $\boldsymbol{(s)}$	$\boldsymbol{t_s}$ $\boldsymbol{(s)}$
ES-PSO	652.51	1936.37	453.764	0.81527	0.909717	0	0.3032	1.1207

6.4. Self-Tuning Fuzzy Fractional Order PID Controller

Table 8. Results obtained from satellite tracking system with STF-FOPID controller

Self-Tuning Fuzzy Fractional Order PID Controller									
e	de	$\boldsymbol{K_P}$	$\boldsymbol{K_I}$	$\boldsymbol{K_D}$	$\boldsymbol{\mu}$	$\boldsymbol{\lambda}$	$\boldsymbol{OS}$ (%)	$\boldsymbol{t_r}$ $\boldsymbol{(s)}$	$\boldsymbol{t_s}$ $\boldsymbol{(s)}$
[-15 15]	**[-25 25]**	**[0 1900]**	**[0 2350]**	**[0 900]**	1.1	1.035	0.4246	0.4901	0.8876

6.5. Variable Coefficient Fractional Order PID Controller

Table 9. Results obtained from satellite tracking system with optimal V-FOPID controller

V-FOPID controller with: $\boldsymbol{J = \int_0^\infty \omega_1 \lvert e(t) \rvert\, dt + \omega_2 * OS\% + \omega_3 * (t_s - t_r)}$											
Algorithms	$\boldsymbol{C_1}$	$\boldsymbol{C_2}$	$\boldsymbol{C_3}$	$\boldsymbol{C_4}$	$\boldsymbol{C_5}$	$\boldsymbol{C_6}$	$\boldsymbol{\mu}$	$\boldsymbol{\lambda}$	$\boldsymbol{OS}$ (%)	$\boldsymbol{t_r}$ $\boldsymbol{(s)}$	$\boldsymbol{t_s}$ $\boldsymbol{(s)}$
GSA-PSO	131.3	510.7	924.38	999.9	1	285.8	0.988	1	0.0018	0.431	0.739
ES-PSO	1	695.9	1000	983.4	1	309.7	0.984	1	0.0012	0.411	0.709

The plots of tracking system response and controller output signal of the proposed controllers depicted in Figure 12 and Figure 13.

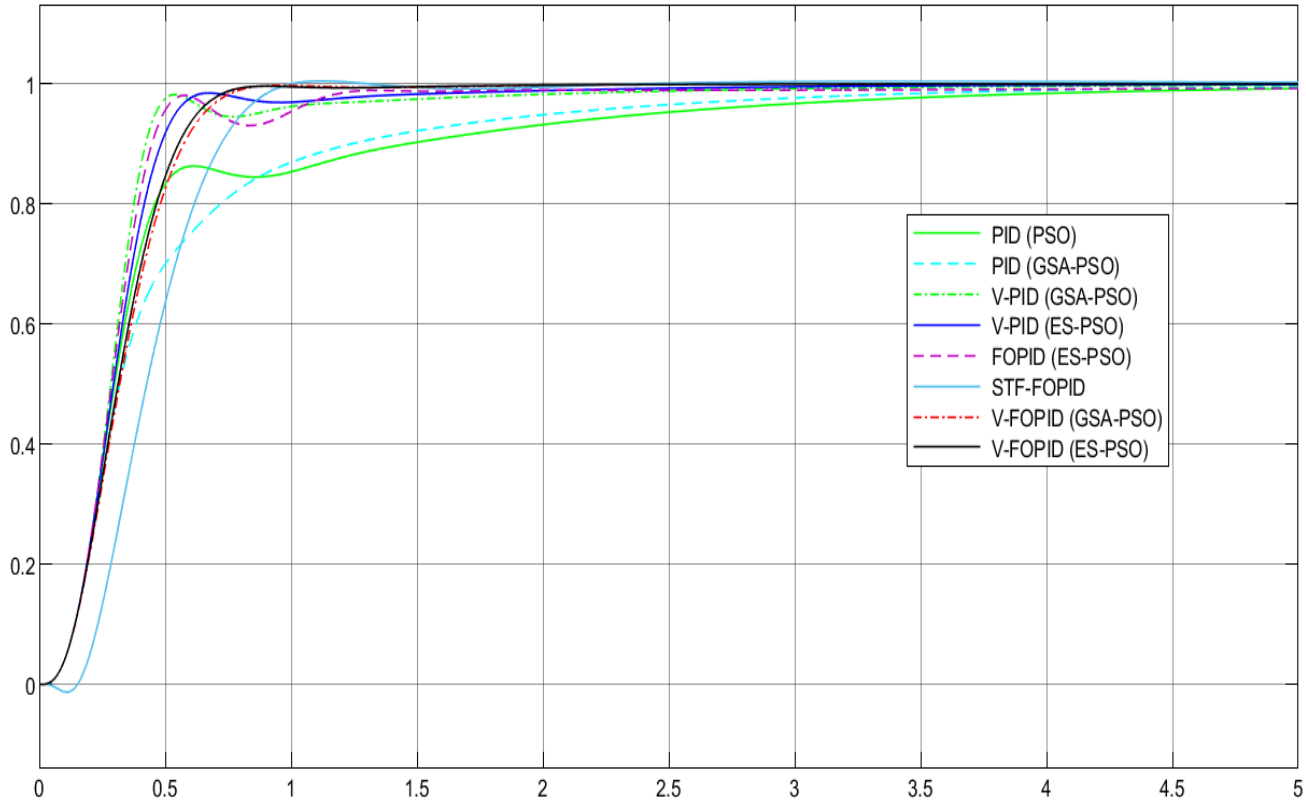

Figure 12. System response with different control strategies

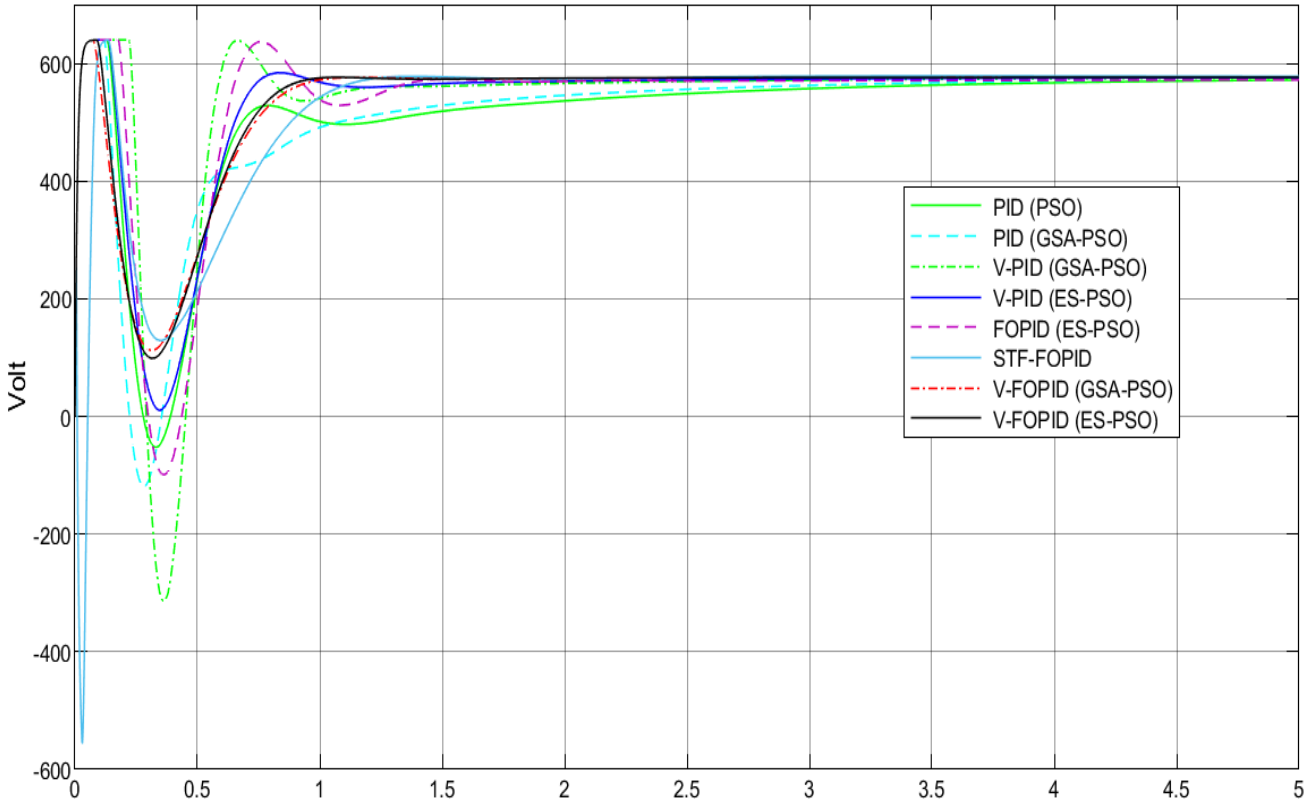

Figure 13. Output signal of different control strategies

PID controller tuned by PSO achieved rise time 1.3 sec and achieved settling time 3.65 sec, also when tuned using GSA-PSO the rise time decreased to 1.09 sec and settling time receded to 3.18 sec. The V-PID controller tuned by GSA-PSO dropped rise time to 0.28 sec and settling time receded to 1.8 sec as compared with PID controller. In addition to, the settling time faded to 1.3 sec when V-PID tuned using ES-PSO. Besides, The FOPID controller tuned by ES-PSO stooped settling time to 1.12 sec. On the other hand, STF-FOPID controller decreased settling time to 1.09 sec. The V-FOPID controller tuned by GSA-PSO receded settling time to 0.73 sec. Plus, when V-FOPID tuned using ES-PSO settling time faded to 0.70 sec. Overshoot percentage for different control strategies nearly equal zero, although PID and FOPID have slight oscillations under refence value and these oscillations increased with V-PID controller. The STF-FOPID controller oscillate around steady state value. These reasons explain why overshoot increased with V-PID and STF-FOPID controllers. Although, the V-PID and STF-FOPID controllers achieved superior outcomes compared to conventual PID controller. The best and the smoothest tracking process and the fastest settling achieved by V-FOPID controller tuned using ES-PSO strategy.

7. Conclusion

This paper aims to design optimal controller to enhance tracking process of multiple mission satellite ground station based on geared DC servomotor. The system is controlled using a variety of control techniques including optimal PID controller, optimal variable coefficient PID (V-PID), optimal fractional order PID (FOPID) controller and optimal variable coefficient fractional order PID (V-FOPID) controller. These controllers' parameters have been fine-tuned by particle swarm optimization (PSO), gravitational search algorithm with particle swarm optimization (GSA-PSO) and Eagle strategy with particle swarm optimization (ES-PSO). The performance index has been determined using dynamic performance indices based objective functions. Additionally, self-tuning fuzzy FOPID (STF-FOPID) is proposed for satellite tracking control system. The system's response is analysed and the outcomes of various control strategies are measured and compared with those of other strategies. In order to improve the performance of tracking system, the FOPID controller is developed to accomplish settling time equal to 1.12 sec when tuned by ES-PSO. Generally, it is difficult to find an accurate non-linear model of actual DC motor. As a result, the STF-FOPID controller is established to deal with the system's uncertain dynamics and non-linearity while still delivering the desired response. The STF-FOPID controller achieved settling time 0.88 sec. PID and FOPID controllers distinguished by constant coefficients which are proportional gain K_P , integral gain K_I and derivative gain K_D. These parameters have the same value throughout the operation regardless of the system error. As a consequence, it is not possible to separate transient and steady-state responses independently. The V-PID and V-FOPID controllers are differentiated by variable coefficient gains symbolized by K_P', K_I' and K_D'. These gains values are dependent on the contemporary values of the system's exact error. As a result, the controller parameters' values vary in relation to the system's exact error. The variable coefficient controllers are capable of improving the transient state while preserving the steady-state response unaltered. This novel variable coefficient fractional order PID technique allows to enhance system's responses in both transient and steady-state conditions, in addition to providing far more freedom in constructing PID & FOPID controllers, it also allows to fine-tune dynamics of control system. The best and the smoothest tracking process and the fastest settling achieved by V-FOPID controller tuned using the ES-PSO strategy. The V-FOPID controller receded settling time from 3.65 sec to 0.70 sec compared with PID controller.

8. Further Work

Further work may focus on designing adaptive control techniques based on fuzzy logic control as follow: design a self-tuning fuzzy PID controller to perform online tuning of (K_P, K_I, K_D, N) parameters of PID controller. Design a self-tuning fuzzy fractional order PID controller, to perform online tuning of $(K_P, K_I, K_D, \lambda, \mu)$ parameters of FOPID controller leading to enhance the controller's

performance and increase its robustness. However, the fuzzy system become more complex as we need five fuzzy system with five tables of rules to fire five ($K_P, K_I, K_D, \lambda, \mu$) outputs. Design a self-tuning fuzzy variable coefficient PID controller to perform online tuning of ($c_1, c_2, c_3, c_4, c_5, c_6, N$) parameters of V-PID controller. Additionally, design a self-tuning fuzzy variable coefficient FOPID to perform online tuning ($c_1, c_2, c_3, c_4, c_5, c_6, \lambda, \mu$) parameters of V-FOPID controller. These variable controller structures will be designed to improve transient state response without impacting steady state response. Besides, capability to control system dynamics and non-linearity, it also provides an opportunity for better control system dynamics adjustment, implying significantly increased system robustness and stability. The Combination of the V-FOPID with a fuzzy control approach may be used to automatically modify the controller's parameters online, enhancing the control parameters' selectability. However, FLC transformed V-FOPID to an adaptive controller, the system's complexity rises, then necessitating to find proper rules for firing fuzzy system outputs. Finally, an adaptive neuro fuzzy inference system (ANFIS) will be used to train the rules of eight fuzzy system based on the running data of V-FOPID controller, the gains ($c_1, c_2, c_3, c_4, c_5, c_6, \lambda, \mu$) of the V-FOPID controller replaced by the eight trained fuzzy logic systems to achieve superior outcomes.

Authors' Contributions

MS designed the structure, produced the necessary software and wrote the article. MH edited the article and contributed to the language. Both authors read and approved the final manuscript.

Competing Interests

The authors declare that they have no competing interests.

References

[1]. M. Shweta et al., "Satellite Dish Positioning System," *IJIRST-International J. Innov. Res. Sci. Technol.*, vol. 4, 2017, [Online]. Available: www.ijirst.org.

[2]. A. Uthman and S. Sudin, "Antenna Azimuth Position Control System Using PID Controller & State-Feedback Controller Approach," *Int. J. Electr. Comput. Eng.*, vol. 8, no. 3, pp. 1539-1550, 2018. doi: 10.11591/ijece.v8i3.pp1539-1550.

[3]. T. Yamaoka, A. Ming, T. Kida, C. Kanamori, and M. Satoh, "Accuracy Improvement of Ship Mounted Tracking Antenna for Satellite Communications," *Nihon Kikai Gakkai Ronbunshu, C Hen/Transactions Japan Soc. Mech. Eng. Part C*, vol. 73, no. 1, pp. 170-176, 2007. doi: 10.1299/kikaic.73.170.

[4]. A. Mahmood, A. Abdulla, and I. Mohammed, "Helicopter Stabilization Using Integer and Fractional Order PID Controller Based on Genetic Algorithm," *Proceedings of the 1st International Multi-Disciplinary Conference Theme: Sustainable Development and Smart Planning, IMDC-SDSP*, Cyperspace, 28-30 June 2020. doi: 10.4108/eai.28-6-2020.2297914.

[5]. Y. Zhang, S. Wang, and G. Ji, "A Comprehensive Survey on Particle Swarm Optimization Algorithm and Its Applications," *Math. Probl. Eng.*, 2015, doi: 10.1155/2015/931256.

[6]. S. Mirjalili and S. Z. M. Hashim, "A New Hybrid PSOGSA Algorithm for Function Optimization," *Proc. ICCIA 2010 - 2010 Int. Conf. Comput. Inf. Appl.*, no. 1, pp. 374-377, 2010, doi: 10.1109/ICCIA.2010.6141614.

[7]. H. Yapici and N. Çetinkaya, "An Improved Particle Swarm Optimization Algorithm Using Eagle Strategy for Power Loss Minimization," *Math. Probl. Eng.*, 2017, doi: 10.1155/2017/1063045.

[8]. B. Ataşlar Ayyıldız and O. Karahan, "Controller Tuning Approach With Tlbo Algorithm for the Automatic Voltage Regulator System," *Eskişehir Tech. Univ. J. Sci. Technol. A - Appl. Sci. Eng.*, vol. 21, no. 1, pp. 128-146, 2020. doi: 10.18038/estubtda.581895.

[9]. N. S. Nise, Control Systems Engineering, Sixth Edit. California State Polytechnic University, Pomona: John Wiley & Sons, Inc., 2000.
[10]. P. Routray, “Dynamic Modelling of DC Motor and Simulation.” pp. 15, 2010.
[11]. S. Javiya and A. Kumar, “Comparisons of Different Controller for Position Tracking of DC Servo Motor,” *Int. J. Adv. Res. Electr. Electron. Instrum. Eng. An ISO*, vol. 3297, no. 2, pp. 966-974, 2007. doi: 10.15662/IJAREEIE.2016.0502058.
[12]. L. Xuan, J. Estrada, and J. Digiacomandrea, “Antenna Azimuth Position Control System Analysis and Controller Implementation,” 2009.
[13]. K. Corporation, “Kollmorgen AKM Servomotor Selection Guide with AKD Servo Drive Systems.” *Kollmorgen Corporation.*, pp. 1-72, 2010. [Online]. Available: https://www.kollmorgen.com/en-us/products/catalogs/kollmorgen-akm-servomotor-selection-guide/.
[14]. K. J. Åström and T. Hägglund, PID Controllers: Theory, Design, and Tuning, 2nd Editio., vol. 2, 1995.
[15]. M. El-Sayed M. Sakr and M. A. Moustafa Hassan, “Satellite Tracking Control System Using Different Optimal Controllers Based on Evolutionary Optimization Techniques,” *Journal of Research in Engineering and Applied Sciences(JREAS)*, vol. 7, no. 1, pp. 226-249, 2022. doi: 10.46565/jreas.2022.v07i01.004.
[16]. I. Podlubny, “Fractional-Order Systems and PI/Sup /Spl Lambda//D/Sup /Spl Mu//-Controllers,” *IEEE Trans. Automat. Contr.*, vol. 44, no. 1, pp. 208-214, 1999. doi: 10.1109/9.739144.
[17]. D. Maiti, S. Biswas, and A. Konar, “Design of a Fractional Order PID Controller Using Particle Swarm Optimization Technique,” *ReTIS-08*, pp. 1-5, 2008. [Online]. Available: http://arxiv.org/abs/0810.3776.
[18]. Q. Gao, K. Li, Y. Hou, R. Hou, and C. Wang, “Balancing and Positioning for a Gun Control System Based on Fuzzy Fractional Order Proportional-Integral-Derivative Strategy,” *Adv. Mech. Eng.*, vol. 8, no. 3, pp. 1-9, 2016. doi: 10.1177/1687814016639854.
[19]. M. Al-Dhaifallah, N. Kanagaraj, and K. S. Nisar, “Fuzzy Fractional-Order PID Controller for Fractional Model of Pneumatic Pressure System,” *Math. Probl. Eng.*, 2018. doi: 10.1155/2018/5478781.
[20]. O. Aydogdu and M. Korkmaz, “Optimal Design of a Variable Coefficient Fractional Order PID Controller by Using Heuristic Optimization Algorithms,” *Int. J. Adv. Comput. Sci. Appl.*, vol. 10, no. 3, pp. 314-321, 2019. doi: 10.14569/IJACSA.2019.0100341.
[21]. D. Bratton and J. Kennedy, “Defining a Standard for Particle Swarm Optimization,” *Proc. 2007 IEEE Swarm Intell. Symp. SIS 2007*, pp. 120-127, 2007, doi: 10.1109/SIS.2007.368035.
[22]. X. S. Yang, S. Deb, and X. He, “Eagle Strategy with Flower Algorithm,” *Proc. 2013 Int. Conf. Adv. Comput. Commun. Informatics, ICACCI 2013*, pp. 1213-1217, 2013. doi: 10.1109/ICACCI.2013.6637350.