Chapter 4

Comparative Performance Evaluation of Fractional Order PID Controller for Heat Flow System using Evolutionary Algorithms

4.1 Introduction

Recent advancement in control system engineering have gained focus of researchers in designing propotional integral derivative (PID) and fractionalpropotional integral derivative (FOPID) since past few decades [175-179, 249]. In spite of many advancement in controller design, still PID controller is very much popular in industry as well as in research area because of its validated advantages like simple construction , easyness in implementation and manageable three control parameters [141-145]. In present scenario, there have been much more advancements in design of PID controller by implimenting the concept of fractional order derivative and integrator to enhance its performances in many practical applications. The FOPID controller is a generalized form of the conventional PID controller. The FOPID controller yields superior closed loop performance than the conventional PID controller both for integer order as well as fractional order systems [180]. Also, it exhibits better robustness performance as against integer order PID controller [3]. However, tuning of FOPID poses a challenging problem due to its extra added tunable parameters such as fractional order integrator λ and fractional order derivative μ. Thus to mitigate the above difficulties, many advance control tuning methods have been implimentated to the design of FOPID controller [3,180-182]. As evolutionary optimization algorithm based design doesnot depend on the rigorous mathematical model of the plants, recently, many research works have been carried out by using different evolutionary optimization algorithms [12,173,181-194]. In this regard, to design FOPID controller, either Genetic algorithm [195], Fuzzy logic [196], Particle swarm optimization (PSO)

[197] or hybrid optimization [198] have been used. However the performance of the FOPID controller for heat flow system (HFS) have never been investigated using the evolutionary algorithms. Therefore, further attempts have been made to use the evolutionary algorithms in designing the FOPID controller for HFS. In this work, the meta-heuristic algorithms such as Particle Swarm Optimization (PSO), Grey Wolf Optimization (GWO), Ant Lion Optimizer (ALO), Moth Flame Optimization (MFO) and hybrid of PSO and Gravitational Algorithm (PSO-GSA) have been used due to their proven advantages discussed below.

Recently, a meta-heuristic algorithm inspired from the ant lions named as Ant Lion Optimizer (ALO) algorithm has been proposed by Mirjalili [171]. ALO has higher rate of exploration which helps to explore the search space effectively. In addition, ALO exhibits high exploitation which helps in rapid convergence for global solution. The efficacy of the ALO is reported in [171] using standard functions. The ALO has been used in the proposed work as it exhibits efficient exploration using random walk and exploitation using adjustive limits of traps.The probability of avoidance of local optima is very high as it is a population based optimization and also because of the uses of random walks and the roulette wheel in the various stages of the optimization process. The ALO is widely used in many engineering domain due to its validatedbenifits in designing different PID and FOPID controllers mentioned in many literatures [190,199-203]. The application of ALO in FOPID for heat flow system is not reported till now. In this chapter, we implement the social behaviour of ant lions for finding the optimum value of the FOPID controller parameters. By considering the discussed advantages of the ALO, the FOPID controller is tuned by ALO optimization for the heat flow systems.

In addition, in the present work Grey Wolf Optimization (GWO) [204] is selected due to its promising result reported in different aspects of science and engineering. Moreover GWO is simple to implement and gives faster convergence. Verma et al. [205] has verified the superiority of GWO in designing the PID control for AVR system. Yang et al. [206] tested the effectiveness of GWO in tracking maximum power point of doubly-fed induction generator based wind turbine. Soni et al. [207] designed a 2DOF-PID control for inter connected power plant using GWO. Moreover, GWO has been used by various authors in

designing and optimization of fractional order Spherical tank system [208], MPPT design for photovoltaic system [209], Wide-area power system stabilizer design [210], optimization for combined heat and power dispatch [211], solving combined economic emission dispatch problems [212], Automatic generation control of a multi-area ST–Thermal power system [213] etc.

Another recently developed optimization algorithm named Moth Flame Optimization (MFO) [214] which has not been used in heat flow system is considered in the present work. MFO is a nature based optimization technique inspired by the tranverse orientation of moths in space. MFO gives global solution by avoiding local minima. As this algorithm uses roulette wheel function, it converges faster. The efficacy of the MFO has been reported by various recent works such as parameter extraction of the three diode model for the multi-crystalline solar cell/module [215], optimal power flow with voltage stability improvement and loss reduction in power system [216], single level production planning in petrochemical industries [217], solving optimal power flow [218] etc.

Moreover, Particle Swarm Optimization (PSO) is a widely used algorithm in solving many engineering problems. In addition, the hybrid PSO-GSA has been applied in many problems. Both the PSO and PSO-GSA algorithms have been used for performance comparison with other algorithms. In this work, four performance indices such as IAE, ISE, ITAE and ITSE are considered to be optimzed using three diffrent algorithms such as PSO, GWO, ALO, MFO and PSO-GSA for designing FOPID controller. These algorithms are selected as these algorithms converges to global optimal solution by avoiding local minima. To show the effectiveness of the proposed controllers, a comparative study is carried out for FOPID controller using different algorithms over the classical PID controller. The main contributions of this chapter as follows:

(i) Formulated an optimization problem for laboratory based heat flow system

(ii) Proposed FOPID controller design using meta-heuristic based algorithms such as PSO, GWO, ALO, MFO and PSO-GSA optimization methods

(iii) Proposed algorithms are used to minimize the performance indices during the design of FOPID controller of the heat flow system.

(iv) Verified the transient and frequency response of the proposed methods by comparing with classical PID controller.

4.2 Preliminaries

4.2.1 Description of Heat Flow System

The HFS is a laboratory based experimental set up supplied by Quanser [219-220]. The main objective of HFS is to regulate at a constant temperature inside the fiber optic chamber. It consists of three sensor made up of platinum transducer placed at equidistance from one another, a blower and a heater. The blower is used to decrease the temperature whereas the heater is used to increase the temperature of the fiber optic chamber to maintain at a desired temperature. In this set up, the temperature of the chamber is maintained at a desired value by regulating the control voltage provides to heater. The physical structure of HFS is shown in Fig.4.1.The temperature of each sensor depends on the input voltage values of blower and heater, ambient chamber temperature and the distance present between heater and sensor. The four depended parameters make the system more complicated. Therefore, required simplification is necessary for designing of the required controller. Eqn 4.1 represents the open loop thermodynamic model of the HFS.

$$T_i = f(V_h, V_b, T_a, X_s) \quad (4.1)$$

Where T_i represents the temperature of i^{th} sensor, V_h and V_b denote the voltage supplied to heater and blower, T_a shows the ambient chamber temperature and X_s shows the distance of heater and sensor n.

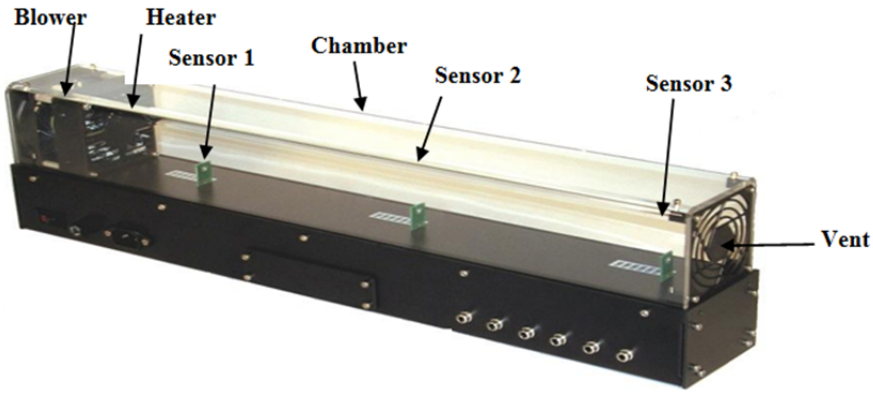

Fig.4.1 Physical diagram of HFS [219]

Here, the HFS system is modeled as a first order delay transfer function which is obtained experimentally from the input and output data in [221], where the output data is obtained by keeping the input voltage fixed for the blower and provides a step input voltage to the heater based on sensor 1 data, the transfer function of the HFS is given by Eqn. 4.2.

$$P(s) = \frac{9.5}{1+30.5s} e^{-0.3s} \tag{4.2}$$

The above system can be simplified into a linear approximated system by using any one from the following consideration.

- Neglecting the delay term
- Using Taylor series expansion
- Using 1st order pade approximation

In this work the system is linearized by using Taylor series approximation and are obtained as

$$P(s) = \frac{9.5}{1+30.5s} (1 - 0.3s) \tag{4.3}$$

4.2.2 The Integer order PID Controller

The transfer function of the basic PID controller is shown in Fig. 4.2. The PID controller is used to estimate the dynamic performance and stability limits of the HFS system. The transfer function of PID controller is given by Eqn.4.4.The proportional gain, integral gain, derivative gain, integral time constant or reset time and derivative time constant or rate time are denoted by K_p, K_i, K_d, T_i and $T_d$respectively. The time domain representation of the PID controller in closed loop system is given by Eqn. 4.5. Where $u(t)$ and $e(t)$ are the control signal and tracking error signal in time domain, respectively. The proportional controller (K_p) is used to control the rise time however the steady state error can't be controlled by it. The integral control (K_i) is used to eliminate the steady-state error, however it can't control the transient response. The derivative control (K_d) is used to increase the stability of the system; Moreover it reduces the overshoot and improves the transient response.

$$C(s) = K_p + \frac{K_i}{s} + sK_d$$

$$or\ C(s) = K_p(1 + \frac{1}{T_i s} + T_d s) \quad (4.4)$$

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau)d\tau + K_d \frac{de(t)}{dt} \quad (4.5)$$

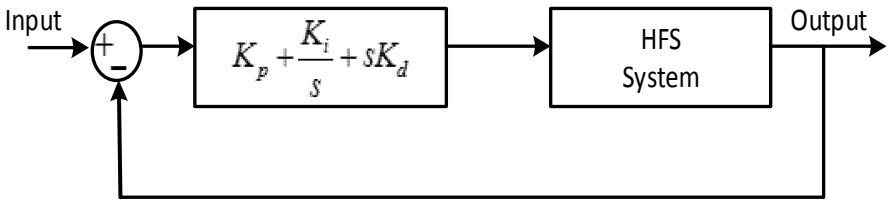

Fig.4.2 Close loop block diagram of Heat flow system with basic PID controller

4.2.3The FOPID Controller

FOPID controller is one of the applications of fractional calculus where the order of the integrator and derivative are in the form fraction. In literature, the most commonly used descriptions for fractional integro-differential are Grunwald-Letnikov, Rieman-Liouville and Caputo [173]. $aD_t^{\propto}$ denotes the general representation of the fractional order integrator and derivative.

$$aD_t^{\propto} = \begin{cases} \frac{d^{\propto}}{dt^{\alpha}} & \propto > 0 \\ 1 & \propto = 0 \\ \int_a^t (dt)^{\propto} & \propto < 0 \end{cases} \tag{4.6}$$

Where $\propto$ denotes the order of the operator, $a\ and\ t$ indicate the lower and upper limits of the operator.

FOPID controller contains two extra added parameters than the PID controller such as fractional order integer (λ) and fractional order derivative (μ). Eqn. 4.7 represents the mathematical expression for FOPID controller which consists of proportional gain (K_p), integral gain (K_i), derivative gain (K_d), λ and μ.

$$\mathrm{P_c(s)} = \frac{\mathrm{U(s)}}{\mathrm{E(s)}} = \mathrm{K_p} + \frac{\mathrm{K_i}}{\mathrm{S^{\lambda}}} + \mathrm{K_d S^{\mu}} \tag{4.7}$$

The FOPID controller represents the generalized form of PID controller. When $\lambda = \mu = 1$, the FOPID behaves as a PID controller. Eqn. 4.8 denotes the time domain representation of FOPID controller.

$$\mathrm{u(t)} = \mathrm{K_p e(t)} + \mathrm{K_i D^{-\lambda} e(t)} + \mathrm{K_d D^{\mu} e(t)} \tag{4.8}$$

In this study, fractional order proportional integrator derivative (FOPID) is taken into consideration for the analysis of HFS system because of their proven advantages. Fig. 4.3 shows the closed loop block diagram of the proposed FOPID- HFS.

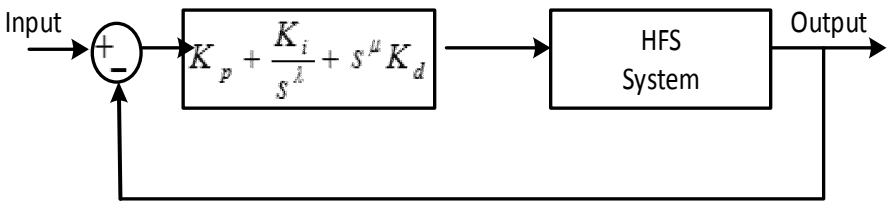

Fig.4.3 Close loop block diagram of HFS with basic FOPID controller

4.3 The Proposed Algorithm based Controller

4.3.1 Problem Formulation and Performance Criteria

These optimum parameters are estimated by minimizing the objective function. Eqns 4.9 – 4.12 represents the objective functions which are selected to minimize the time response characteristics due to the time dependency on error.Fig.4.4 shows block diagram of proposed FOPID controller based on evolutionary algorithms.

J_1 = IAE (Integral Absolute Error) = $\int_0^\infty |e(t)|\, dt$(4.9)

J_2 = ISE (Integral Square Error) = $\int_0^\infty e^2(t)dt$(4.10)

J_3 = ITAE (Integral with Time Absolute Error) = $\int_0^\infty t|e(t)|\, dt$(4.11)

J_4 = ITSE (Integral with Time Square Error) = $\int_0^\infty te^2(t)dt$(4.12)

The problem can be represented as

Minimize J (4.13)

Subjected to

$$K_{pmin} < K_p < K_{pmax}$$

$$K_{imin} < K_i < K_{imax}$$

$$K_{dmin} < K_d < K_{dmax}$$

$$\lambda_{min} < \lambda < \lambda_{max}$$

$$\mu_{min} < \mu < \mu_{max}$$

Here, J denotes the objective function ($J_1, J_2, J_3,$ and J_4) and e(t) is the error $e = max|r(t) - y(t)|$.Where $r(t)$ and $y(t)$ are defined as the desired input of the system and system output respectively.

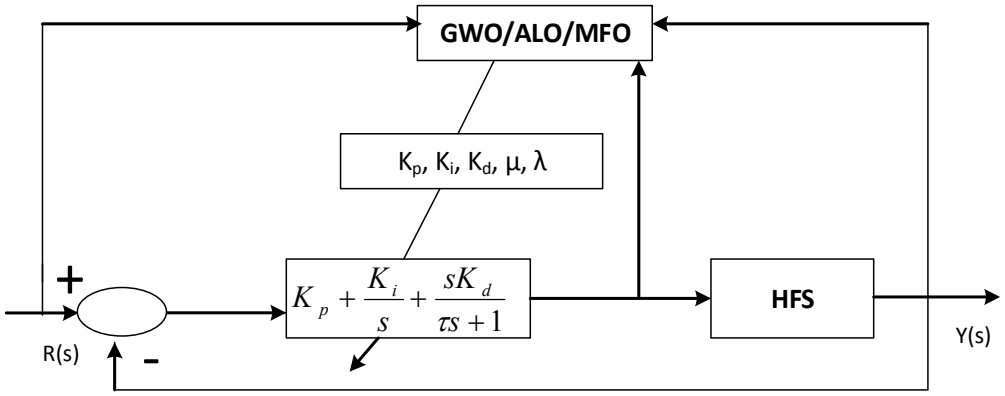

Fig.4.4 Block diagram of designed FOPID controller tuned by Evolutionary Algorithms

4.3.2 Proposed Algorithms

4.3.2.1 Proposed FOPID controller design using Particle Swarm Optimization (PSO)

- **Input**: Tuning parameters of Heat flow system (HFS), population size and velocity of swarms.
- **Output:** Optimal values for the tunable control parameters for FOPID controller

1.Initialize the population (n), velocity (v), random position (x) of swarms and the lower and upper bounds of the tunable control parameters such as K_P, K_I, K_D, λ, μ. Eqn. 4.14 represents the position matrix and Eqn. 4.15 represents the velocity matrix.

$$x = \begin{bmatrix} K_{p1} & K_{i1} & K_{d1} & \lambda_1 & \mu_1 \\ K_{p2} & K_{i2} & K_{d2} & \lambda_2 & \mu_2 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ K_{pn} & K_{i3} & K_{d3} & \lambda_n & \mu_n \end{bmatrix} \tag{4.14}$$

$$v = \begin{bmatrix} v_{p1} & v_{i1} & v_{d1} & v_{\lambda 1} & v_{\mu 1} \\ v_{p2} & v_{i2} & v_{d2} & v_{\lambda 2} & v_{\mu 2} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ v_{pn} & v_{in} & v_{dn} & v_{\lambda n} & v_{\mu n} \end{bmatrix} \tag{4.15}$$

2. Eqn 4.16 represents the error index function which is known as the fitness function $f(x)$The fitness of each population is evaluated and based on the best fitness, the global best (g_{bst}) is determined.

$$f(x) = \begin{bmatrix} f(K_{p1}, K_{i1}, K_{d1}, \lambda_1, \mu_1) \\ f(K_{p2}, K_{i2}, K_{d2}, \lambda_2, \mu_2) \\ \vdots \\ f(K_{pn}, K_{in}, K_{dn}, \lambda_n, \mu_n) \end{bmatrix} \tag{4.16}$$

3. The velocity of each particle is updated in each iteration based on the local and global particle positions. The updated velocity and positions are evaluated based on Eqn. 4.17 and Eqn.4.18.

$$v_{updated(i)} = v_{old(i)} + c_1 r_1 (x_{local} - x_i) + c_2 r_2 (x_{global} - x_i) \tag{4.17}$$

$$x_{updated(i)} = x_{old(i)} + v_{updated(i)} \tag{4.18}$$

Where c_1, c_2represents the constant values and r_1, r_2denotes random numbers.

4. The fitness functions are calculated using the updated particle position and the local positions are updated using the fitness function.

5. Repeat Step 3 and 4 until the maximum number of iterations.

6. After the last iteration, record the best fitness as optimal value for controller parameter.

4.3.2.2 Proposed controller design using Grey Wolf Optimization (GWO)

The basic idea behind the GWO-FOPID controller design for HFS system is the nature inspired algorithm named as Grey Wolf Optimization, which is recently developed by Sayed Mirjalil [204].The block diagram of the proposed FOPID is given in Fig. 4.2. The stepwise details of the proposed GWO-FOPID controller design are presented as follows:

- **Input**: Tuning parameters of Heat flow system (HFS) with randomly generated grey wolf size n in 5 dimensions.
- **Output:** Optimal values for the tunable control parameters for FOPID controller.

1. Initialize the position (x), maximum iteration (N), lower and upper limits of tunable controller parameters like K_P, K_I, K_D, λ, μ. note: grey wolf population size should be within 8 to 12 [204].

$$x = \begin{bmatrix} K_{p1} & K_{i1} & K_{d1} & \lambda_1 & \mu_1 \\ K_{p2} & K_{i2} & K_{d2} & \lambda_2 & \mu_2 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ K_{pn} & K_{i3} & K_{d3} & \lambda_n & \mu_n \end{bmatrix} \tag{4.18}$$

2. The fitness function $f(x)$ is considered as the error index and it is evaluated for every grey wolfs.

$$f(x) = \begin{bmatrix} f(K_{p1}, K_{i1}, K_{d1}, \lambda_1, \mu_1) \\ f(K_{p2}, K_{i2}, K_{d2}, \lambda_2, \mu_2) \\ \vdots \\ f(K_{pn}, K_{in}, K_{dn}, \lambda_n, \mu_n) \end{bmatrix} \tag{4.19}$$

3. Three best positioned wolves are determined using fitness value and named as α, β, γ respectively and rest wolves are named as ω who are followers of α, β, γ.

4. The positions of the wolves are updated in each iteration according to the positions of $\alpha, \beta \; and \; \gamma$ as shown in the following equations.

$$\overrightarrow{D_\alpha} = \left|\overrightarrow{C_1}.\overrightarrow{X_\alpha(t)} - \overrightarrow{X(t)}\right|, \overrightarrow{D_\beta} = \left|\overrightarrow{C_1}.\overrightarrow{X_\beta(t)} - \overrightarrow{X(t)}\right|, \overrightarrow{D_\gamma} = \left|\overrightarrow{C_1}.\overrightarrow{X_\gamma(t)} - \overrightarrow{X(t)}\right| \tag{4.20}$$

$$\overrightarrow{X_1} = \overrightarrow{X_\alpha(t)} - A_1.\overrightarrow{D_\alpha}, \overrightarrow{X_2} = \overrightarrow{X_\beta(t)} - A_2.\overrightarrow{D_\beta}, \overrightarrow{X_3} = \overrightarrow{X_\gamma(t)} - A_1.\overrightarrow{D_\gamma} \tag{4.21}$$

$$A_{(1,2,3)} = 2.\vec{a}.rand(0,1) - \vec{a}, C_{(1,2,3)} = 2.rand(0,1) \quad (4.22)$$

$$\overrightarrow{X(t+1)} = \frac{X_1+X_2+X_3}{3} \quad (4.23)$$

Here t is the current iteration, $X_\alpha(t)$, $X_\beta(t)$ and $X_\gamma(t)$ are the positions of α, β and γ respectively at t^{th} iteration and $\vec{a}$ is a variable which decreases linearly from 0 to 2.

5. α, β and γ Wolves positions are replaced according to the updated wolves position.

6. Repeat Step 4 and 5 until maximum iterations.

7. After the last iteration, record the best fitness as optimal value for controller parameter.

4.3.2.3 Proposed controller design using Ant Lion Optimization (ALO)

- **Input:**Tuning parameters of Heat flow system (HFS) with randomly generated ant and ant lion of size n in 5 dimensions.
- **Output:**Optimal values for the tunable control parameters for FOPID controller.

1. Initialize the population of ant, ant lions and number of iterations by considering the boundary values of the controller parameters (KP, KI, KD, λ, μ).

2. The position matrix Mant represents the position of ants. In the matrix, the index n indicates the number of ants and d represents the number of variables or dimensions. As the five control parameters need to be adjusted, the number of variable is five in this case.

$$\mathrm{M_{ant}} = \begin{bmatrix} \mathrm{A_{1,1}} & \mathrm{A_{1,2}} & \cdots & \mathrm{A_{1,d}} \\ \mathrm{A_{2,1}} & \mathrm{A_{2,2}} & \cdots & \mathrm{A_{2,d}} \\ \vdots & \vdots & \vdots & \vdots \\ \mathrm{A_{n,1}} & \mathrm{A_{n,2}} & \cdots & \mathrm{A_{n,d}} \end{bmatrix} \quad (4.24)$$

3. The fitness function for each ant is calculated and kept in a fitness matrix. It is given in equation (4.25). Here, the index n and d indicates the population size of ant and dimensions respectively and f represents the fitness function.

$$M_{FA} = \begin{bmatrix} f(A_{1,1} & A_{1,2} & \dots & A_{1,d}) \\ f(A_{2,1} & A_{2,2} & \dots & A_{2,d}) \\ \vdots & \vdots & \vdots & \vdots \\ f(A_{n,1} & A_{n,2} & \cdots & A_{n,d}) \end{bmatrix} \quad (4.25)$$

4. The position of ant lions within the search space is represented in a matrix $M_{antlion}$. It is given in equation (4.26). Fitness value of ant lion is shown in matrix M_{FAL} (equation (4.27)). The index n represents the ant lions and d indicates the dimensions.

$$M_{antlion} = \begin{bmatrix} AL_{1,1} & AL_{1,2} & \dots & AL_{1,d} \\ AL_{2,1} & AL_{2,2} & \dots & AL_{2,d} \\ \vdots & \vdots & \vdots & \vdots \\ AL_{n,1} & AL_{n,2} & \cdots & AL_{n,d} \end{bmatrix} \quad (4.26)$$

$$M_{FAL} = \begin{bmatrix} f(AL_{1,1} & AL_{1,2} & \dots & AL_{1,d}) \\ f(AL_{2,1} & AL_{2,2} & \dots & AL_{2,d}) \\ \vdots & \vdots & \vdots & \vdots \\ f(AL_{n,1} & AL_{n,2} & \cdots & AL_{n,d}) \end{bmatrix} \quad (4.27)$$

5. The best fitness function is chosen from M_{FAL} and respective ant lion is selected as an optimum one.

6. In each iteration, an ant lion is selected using roulette wheel. According to the ant lion, the boundary positions are updated by using equations (4.28) and (4.29) which are proportional to the current iteration.

$$C^t = \frac{C^t}{I} \quad (4.28)$$

$$d^t = \frac{d^t}{I} \quad (4.29)$$

$$I = 10^w \frac{t}{T} \quad (4.30)$$

Here, C^t and d^t are the leat and highest value of all variables at t^{th} iteration respectively, t is the present iteration, T indicates total number of iteration, w is the constant which depends on the present iteration and I presents the ratio.

7. Random walk for each ant is created and normalized its step around the search space of selected ant lion by using equation (4.31) and (4.33) for every ant. After that the ants' positions are updated by considering equation (4.35).

$$X(t) = [0, \text{cusum}(2r(t_1) - 1), \cdots \text{cusum}(2r(t_n) - 1)] \quad (4.31)$$

Here, cusum evaluates aggregate, n and r represents the highest numbers of iteration and the steps of random walk respectively. Eqn. 4.32 defines the function r(t).

$$r(t) = \begin{cases} 1 \text{ if ran\_num} > 0.5 \\ 0 \text{ if ran\_num} \leq 0.5 \end{cases} \quad (4.32)$$

Theran_num presents a random number produces using the uniform distribution with a limit from 0 to 1. Eqn. 4.33 represents the normalized random walk of ant.

$$X_i(t) = \frac{(X_i^t - a_i) \times (d_i^t - C_i^t)}{(b_i - a_i)} + C_i^t \quad (4.33)$$

Here, a_i and b_i denotes the least and highest value of the random walk of i^{th}variable respectively, C_i^t *and* d_i^trepresnts the minimum value and the maximum value of the i^{th} variable at t^{th}iteration.

$$\text{Antlion}_j^t = \text{Ant}_i^t \text{ if } f(\text{Ant}_i^t) > f(\text{Antlion}_j^t) \quad (4.34)$$

Where t is the current iteration, Antlion_j^t and Ant_i^t deonotes as the position of the selected j^{th} ant lion and position of i^{th} ant, respectively. The fitness value of ant and ant lion is denoted by the function f().

$$\text{Ant}_i^t = \frac{R_A^t + R_E^t}{2} \quad (4.35)$$

Here,at t^{th} iteration, R_A^t and R_E^t represents the random walk around the selected ant lion and the random walk near by the elite respectively. Ant_i^trepresents the position of i^{th} ant.

8. Then the fitness values of all the ants are calculated and replaced the ant lion with its corresponding ant if it becomes fitter as shown in Eqn.4.34.

9. The position of optimum ant lion is updated, if the ant lion is having highest fitness function than the optimum.

10. The above process is repeated until the maximum iteration is not satisfied. Once all the iterations are over the best or elite solution is calculated.

4.3.2.4 Proposed controller design using Moth Flame Optimization (MFO)

The MFO is a nature inspired algorithm, which is recently developed by Sayed Mirjalil [214]. The stepwise details of the proposed MFO-FOPID controller design are presented as follows:

- **Input:**System with FOPID controller parameter with n number moths in 5 dimensions
- **Output:**Optimal values for the tunable control parameters for FOPID controller.

1. Initialize the position of moths (M), dimension (d), population (n), maximum iterations (N), lower and upper limits of the tunable controller parameters like K_p, K_i, K_d, λ, μ.

$$\mathrm{M_{moth}} = \begin{bmatrix} \mathrm{M_{1,1}} & \mathrm{M_{1,2}} & \dots & \mathrm{M_{1,d}} \\ \mathrm{M_{2,1}} & \mathrm{M_{2,2}} & \dots & \mathrm{M_{2,d}} \\ \vdots & \vdots & \vdots & \vdots \\ \mathrm{M_{n,1}} & \mathrm{M_{n,2}} & \cdots & \mathrm{M_{n,d}} \end{bmatrix} \tag{4.36}$$

2. The fitness function for each moth is calculated and kept in a fitness matrix. It is given in equation (4.37). Here, the index n and d indicates the population size of moth and dimensions respectively and f represents the fitness function.

$$\mathrm{F_{moth}} = \begin{bmatrix} \mathrm{f(M_{1,1}} & \mathrm{M_{1,2}} & \dots & \mathrm{M_{1,d})} \\ \mathrm{f(M_{2,1}} & \mathrm{M_{2,2}} & \dots & \mathrm{M_{2,d})} \\ \vdots & \vdots & \vdots & \vdots \\ \mathrm{f(M_{n,1}} & \mathrm{M_{n,2}} & \cdots & \mathrm{M_{n,d})} \end{bmatrix} \tag{4.37}$$

3. Perform the following (4 to 6) steps while iteration is less than the maximum iteration
4. Calculate the flame number using the following equation

$$FlameNumber = round(N - CurrentIteration * \frac{N-1}{MaxItr}) \tag{4.38}$$

where, N is the maxmum number of flames.

5. When moths are within the limits, then sort the moths according to the fitness value and determine the best flame and best flame fitness.
6. Update the moth position according to the flame using spiral function.

7. The above process is continued until the maximum iteration is not satisfied. Once all the iterations are over the best solution is calculated.

4.3.2.5 Proposed controller design using Hybrid PSO-GSA for Heat Flow System

- **Input:** Tunable parameters of the Heat flow system (HFS) with population size of swarm, velocity of particles and mass.
- **Output:** Optimal values for the tunable control parameters for FOPID controller.

1. Initialize the position (x), velocity (v), mass (m), population size (n), maximum iteration (N) and lower and upper limits of tunable parameters. Ens.4.39 represents the position matrix, Eqn 4.40 represents the velocity matrix and Eqn 4.41 represents the mass matrix.

$$x = \begin{bmatrix} K_{p1} & K_{i1} & K_{d1} & \lambda_1 & \mu_1 \\ K_{p2} & K_{i2} & K_{d2} & \lambda_2 & \mu_2 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ K_{pn} & K_{i3} & K_{d3} & \lambda_n & \mu_n \end{bmatrix} \tag{4.39}$$

$$v = \begin{bmatrix} v_{p1} & v_{i1} & v_{d1} & v_{\lambda 1} & v_{\mu 1} \\ v_{p2} & v_{i2} & v_{d2} & v_{\lambda 2} & v_{\mu 2} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ v_{pn} & v_{in} & v_{dn} & v_{\lambda n} & v_{\mu n} \end{bmatrix} \tag{4.40}$$

$$m = \begin{bmatrix} m_{p1} & m_{i1} & m_{d1} & m_{\lambda 1} & m_{\mu 1} \\ m_{p2} & m_{i2} & m_{d2} & m_{\lambda 2} & m_{\mu 2} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ m_{pn} & m_{in} & m_{dn} & m_{\lambda n} & m_{\mu n} \end{bmatrix} \tag{4.41}$$

2. Eqn 4.42 represents the fitness matrix which is evaluated based on the fitness function (x).

$$f(x) = \begin{bmatrix} f(K_{p1}, K_{i1}, K_{d1}, \lambda_1, \mu_1) \\ f(K_{p2}, K_{i2}, K_{d2}, \lambda_2, \mu_2) \\ \vdots \\ f(K_{pn}, K_{in}, K_{dn}, \lambda_n, \mu_n) \end{bmatrix} \tag{4.42}$$

3. The velocity of each particle is updated in each iteration based on the local and global particle positions. The updated velocity and positions are evaluated based on Eqn. 4.43 and Eqn.4.44.

$$v_{updated(i)} = v_{old(i)} + c_1 r_1 a_i + c_2 r_2 (x_{global} - x_i) \quad (4.43)$$

$$x_{updated(i)} = x_{old(i)} + v_{updated(i)} \quad (4.44)$$

Here, c_1, c_2 denotes the constant values, a_i represents the acceleration of i^{th} particle and r_1, r_2 denotes random numbers.

4. The fitness functions are calculated using the updated particle position and the local positions are updated using the fitness function.

5. Repeat Step 3 and 4 until the maximum number of iterations.

6. After the last iteration, record the best fitness as optimal value for controller parameter.

4.4 Result and Analysis

4.4.1 Parameters Setting

The lower bound and upper bound are set as5 and 20 respectively for proportional gain (K_p), integral gain (K_i), derivative gain (K_d) and for integer order (λ) and derivative order (μ) the lower and upper bound are set as 0 and 1. During the design of evolutionary based FOPID controller for HES the following parameter values has been considered. Population size = 12, Maximum number of iterations =100, Number of evaluations = 20, Inertia weight=0.9, swarm best weight =2, particle best weight = 0.5, c_1= 2, c_2=2, Number of evaluations = 20, G=1, a=30.

4.4.2 Transient Response Analysis

Simulation studies are carried out to make comparison analysis among different stated objective functions. In this work, the evolutionary algorithms such as PSO, ALO, GWO, MFO and PSO-GSA are used to optimize the control parameters of FOPID controller for Heat Flow System (HFS). In order to compare the transient response of the PID controller for HFS, simulations has been carried out and the best of the PID-HFS tuned by the above evolutionary algorithms are presented in this paper. The best parameters of GWO-FOPID, ALO-FOPID, MFO-FOPID, PSO- FOPID and PSOGSA-FOPID controller with different objective functions are shown in Table 4.1, 4.3, 4.5, 4.7 and 4.9 respectively. The step response with optimized FOPID controller is depicted in Figs. 4.5, 4.6, 4.7, 4.8 and 4.9 for GWO, ALO, MFO, PSO and PSO-GSA algorithm respectively. Fig. 4.5 represents the step response of the optimized FOPID controller with different objective function tuned by GWO algorithm. From Fig. 4.5, it is clear that the systems settling time is minimum for the ITSE objective function than other objective functions tuned by GWO algorithm. Moreover, the rise time and overshoot value are in considerable range. Similarly, Fig. 4.6 represent comparison of transient responses of various stated objective function tuned by ALO algorithm. The systems settling time is minimum in case of ITAE objective function than other objective functions. In Fig. 4.7, the comparison of step response of various objective functions tuned by MFO is presented. It is evident from the Fig. 4.7 that the systems settling time is minimum in case of ITSE objective function. In addition, the rise time and settling time are also minimum for ITSE objective function. From the Fig. 4.8, it is clear that the systems settling time is minimum for the IAE objective function than other objective functions tuned by PSO algorithm. Similarly, Fig. 4.9 represent comparison of transient responses of various stated objective function tuned by PSO-GSA algorithm. The systems settling time is minimum in case of ITSE objective function than other objective functions. From, Table 4.2 and 4.4, it is clear that GWO-FOPID and MFO-FOPID are giving the better result for the performance criteria ITSE. However, for ALO-FOPID, ITAE performance criterion is giving better result, which is shown in Table 4.6. Similarly, in case of PSO and PSO-GSA, the better result is obtained from IAE and ITSE

which is shown in Table 4.8 and Table 4.10. The proposed FOPID controller has been compared with other methods such as fractional order filter with IMC-PID technique [194], ALO-PID, GWO-PID, MFO-PID etc. The comparative result is shown in Table 4.11. The above different tuning techniques are compared with respect to the transient performance such as rise time (T_r), settling time (T_s), peak time (T_p), maximum overshoot ($\%M_p$) and steady state performance i.e. steady state error (E_{ss}). From the simulation result presented in Table 4.11, it has been observed that the proposed ALO tuned FOPID controller gives comparative good performance than the other presented methods.

The step response for all PID controlled HFS and all FOPID-HFS is given in Fig. 4.10 and 4.11 respectively. In Fig. 4.10, the comparison of responses of IMC-PID, ALO-PID, GWO-PID and MFO-PID are given. Among the compared methods, GWO-PID performs better than other methods in terms of rise time, settling time and overshoot. Fig 4.11 represents the comparison of responses of FOPID controlled Heat Flow System optimized by PSO, PSO-GSA, MFO, ALO and GWO algorithm. It is evident form Fig.4.11 that the ALO-FOPID gives much better result than GWO-FOPID, PSO-FOPID and PSO-GSA-FOPID in terms of settling time and overshoots. The best of PID i.e., GWO-PID and the best of FOPID i.e., ALO-FOPID responses are compared and presented in Fig. 4.12. It is clear from Fig. 4.12 that the ALO-FOPID performs better than GWO-PID for Heat Flow System in terms of transient responses.

Table 4.1 Various objective functions of GWO tuned FOPID controller for HFS

Parameters/ Objective functions	IAE	ISE	ITAE	**ITSE**
K_P	3.631	1.3589	3.7437	**4.637**
K_I	0.36	0.6946	0.35	**0.329**
K_D	1.3244	1.47	0.768	**0.946**
λ	0.8119	0.33	0.972	**0.819**
μ	0.874	0.43	0.684	**0.739**

Table 4.2 Transient parameters of GWO tuned FOPID for HFS

Method	T_r(s)	T_s(s)	Overshoot (p.u)	Peak (p.u)	T_p(s)
GWO-FOPID-IAE	0.275	2.663	0.015	1.015	4.863
GWO-FOPID-ISE	2.382	6.579	0	0.982	8.696
GWO-FOPID-ITAE	1.225	2.01	0.02	1.02	4.453
GWO-FOPID-ITSE	**0.955**	**1.817**	**0.004**	**1.004**	**7.673**

Table 4.3 Various objective functions of ALO tuned FOPID controller for HFS

Parameters/ Objective functions	IAE	ISE	**ITAE**	ITSE
K_P	3.31	2.81	**4.73**	3.70
K_I	0.41	0.98	**0.63**	0.29
K_D	1.13	1.89	**0.60**	1.83
λ	0.45	0.30	**0.61**	0.53
μ	0.94	0.74	**0.62**	0.39

Table 4.4 Transient parameters of ALO tuned FOPID for HFS

Method	T_r(s)	T_s(s)	Overshoot (p.u)	Peak (p.u)	T_p(s)
ALO-FOPID-IAE	1.70	3.34	0	0.993	5.38
ALO-FOPID-ISE	1.53	3.70	0	0.99	5.31
ALO-FOPID-ITAE	**0.58**	**0.90**	**0.004**	**1.004**	**3.814**
ALO-FOPID-ITSE	0.53	4.65	0	0.99	151.1

Table 4.5 Various objective functions of MFO tuned FOPID controller for HFS

Parameters/ Objective functions	IAE	ISE	ITAE	**ITSE**
K_P	3.9364	0.2506	3.1176	**4.1383**
K_I	0.6017	1.1509	1.1846	**0.6373**
K_D	0.5966	1.1657	0.5954	**0.4350**
λ	0.94748	0.2101	0.4862	**0.4343**
μ	0.43006	0.6387	0.4710	**0.9288**

Table 4.6 Transient parameters of MFO tuned FOPID for HFS

Method	T_r(s)	T_s(s)	Overshoot(p.u)	Peak(p.u)	T_p(s)
MFO-FOPID-IAE	0.769	11.46	0.045	1.045	3.843
MFO-FOPID-ISE	3.891	14.12	0	0.983	10.42
MFO-FOPID-ITAE	0.826	3.959	0.042	1.042	3.235
MFO-FOPID-ITSE	**0.944**	**1.472**	**0.009**	**1.009**	**2.33**

Table 4.7 Various objective functions of PSO tuned FOPID controller for HFS

Parameters/ Objective functions	**IAE**	ISE	ITAE	ITSE
K_P	**4.5384**	4.3872	4.9474	4.375
K_I	**0.4286**	0.4178	0.9687	1.810
K_D	**1.8441**	1.7414	1.9197	1.403
λ	**0.7059**	0.9198	0.7794	0.365
μ	**0.2780**	0.3748	0.6647	0.723

Table 4.8 Time domain parameters of PSO tuned FOPID for HFS

Method	T_r(s)	T_s(s)	Overshoot (p.u)	Peak (p.u)	T_p(s)
PSO-FOPID-IAE	**0.551**	**2.394**	**0.309**	**1.309**	**1.227**
PSO-FOPID-ISE	0.552	5.697	0.25	1.25	1.234
PSO-FOPID-ITAE	0.479	5.282	0.187	1.187	1.175
PSO-FOPID-ITSE	0.517	2.941	0.248	1.248	1.229

Table 4.9 Various objective functions of PSOGSA tuned FOPID controller for HFS

Parameters/ Objective functions	IAE	ISE	ITAE	**ITSE**
K_P	3.94	2.81	3.32	**4.33**
K_I	0.28	0.87	1.38	**0.59**
K_D	1.95	1.79	0.61	**0.48**
λ	0.81	0.41	0.51	**0.47**
μ	0.9	0.69	0.39	**0.89**

Table 4.10 Time domain Performance of various objective function tuned by PSOGSA

Method	T_r (s)	T_s (s)	Overshoot (p.u)	Peak (p.u)	T_p(s)
PS-GSA-FOPID-IAE	1.534	3.112	0.004	1.004	8.287
PSOGSA-FOPID-ISE	1.407	3.147	0	0.997	4.669
PSOGSA-FOPID-ITAE	0.691	4.244	0.066	1.066	1.662
PSOGSA-FOPID-ITSE	**0.858**	**1.406**	**0.007**	**1.007**	**2.383**

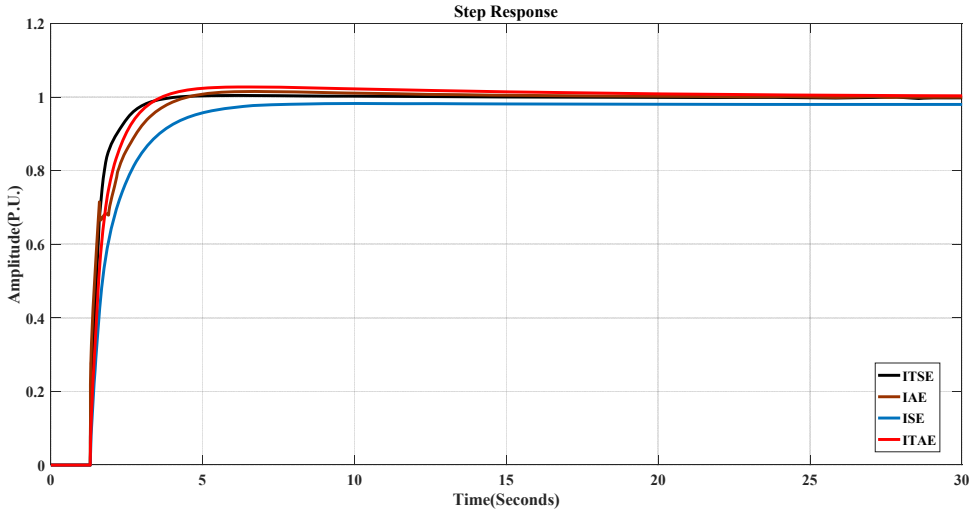

Fig.4.5 Step response of different objective functions tuned by GWO for HFS

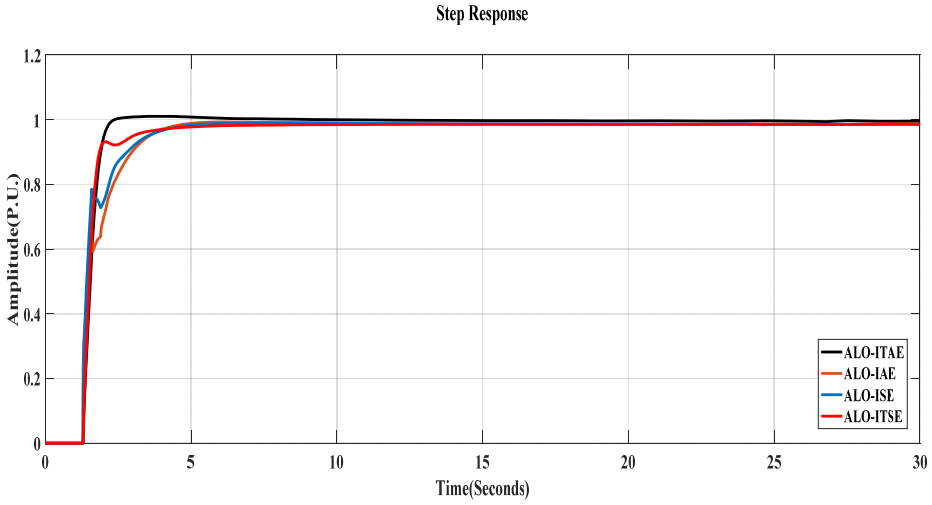

Fig.4.6 Step response curve of different objective functions tuned by ALO for HFS

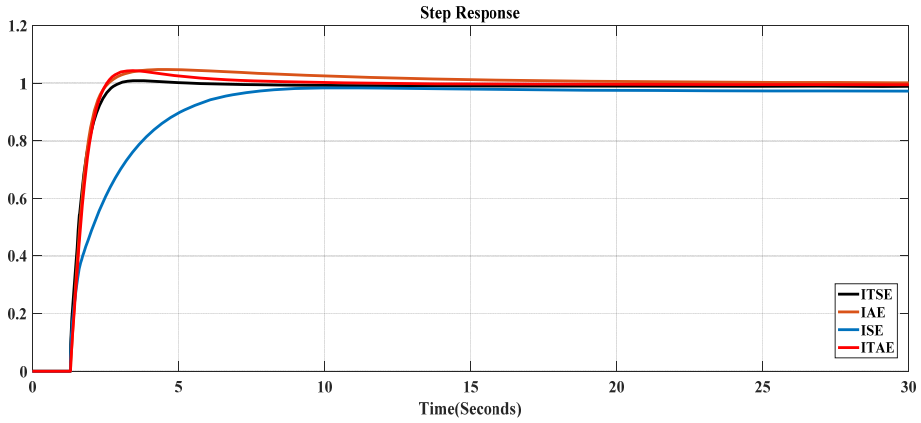

Fig.4.7 Step response curve of different objective functions tuned by MFO for HFS

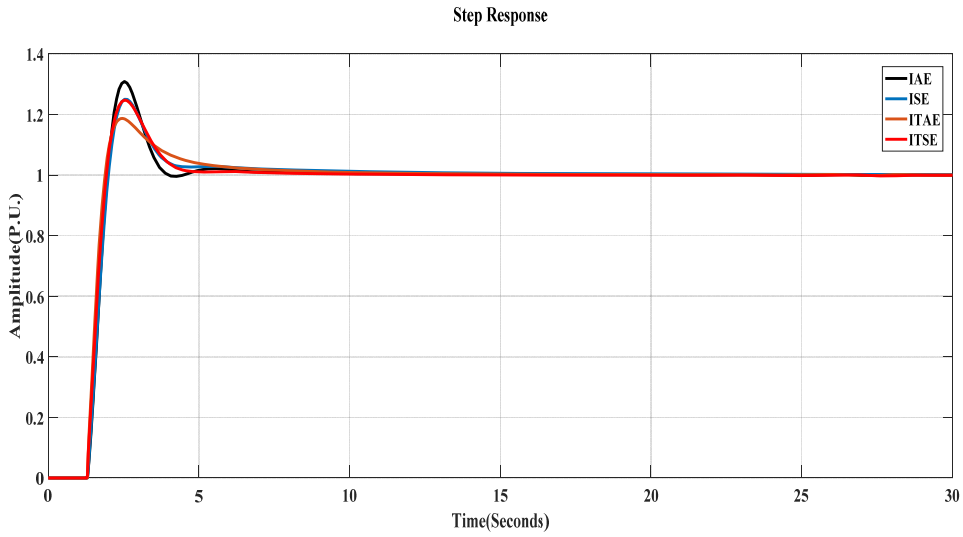

Fig 4.8 Step response curve of different objective functions tuned by PSO for HFS

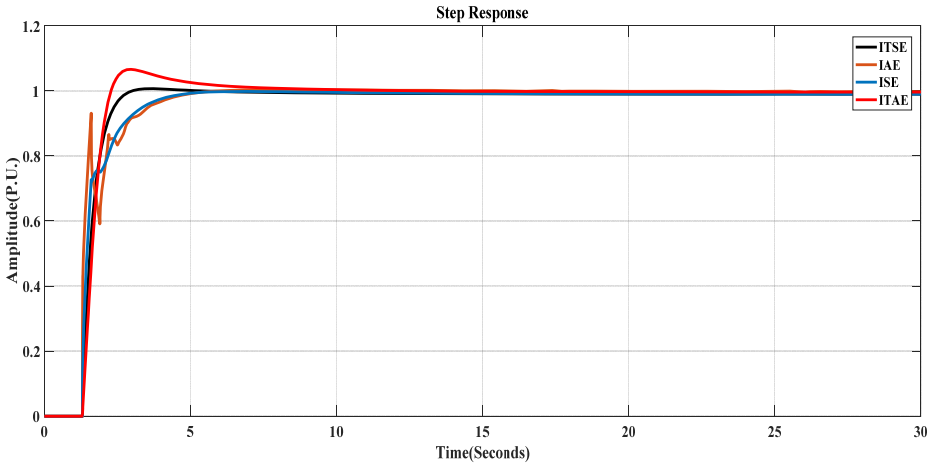

Fig 4.9 Step response curve of different objective functions tuned by PSOGSA for HFS

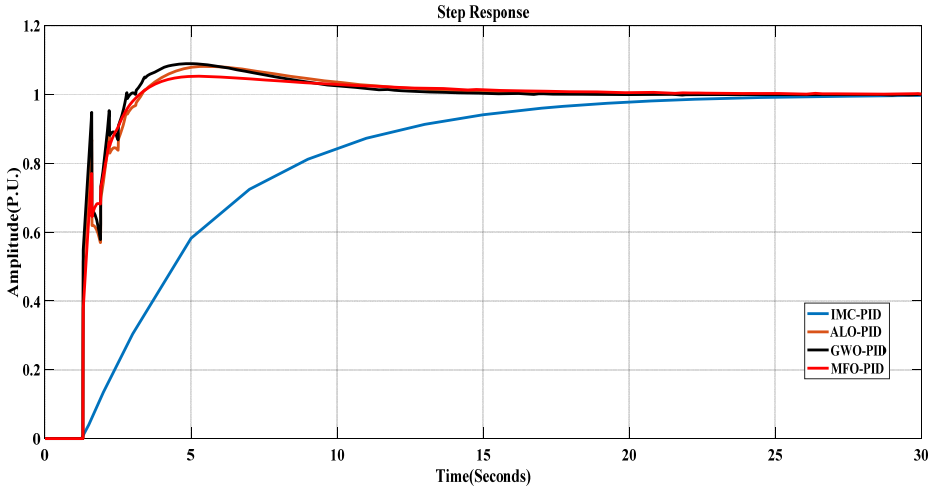

Fig.4.10 Comparative transient analysis of various PID controlled Heat Flow System

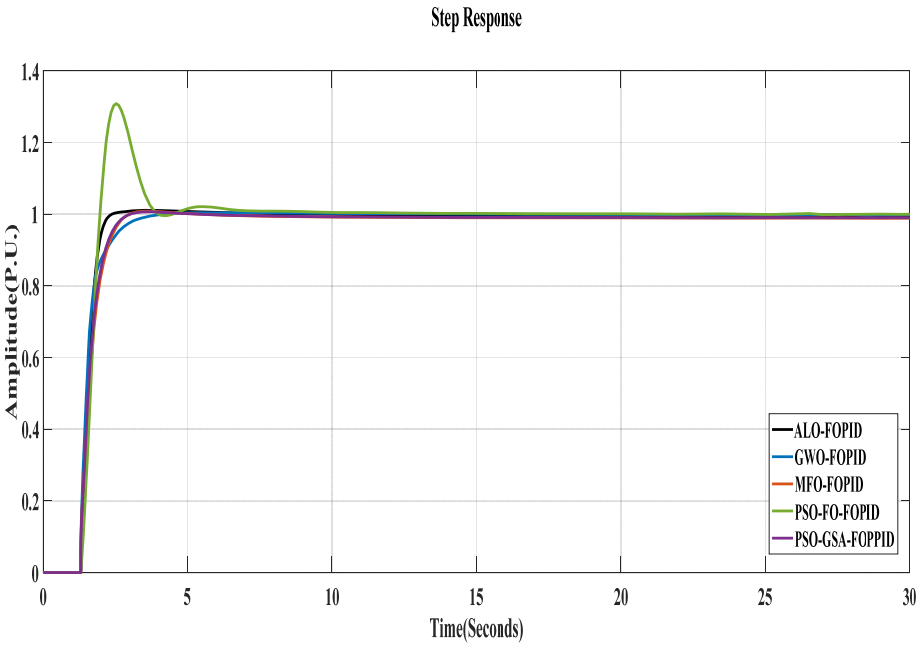

Fig.4.11 Comparative transient analysis of FOPID controlled Heat Flow system

Table 4.11 Comparative transient analysis of various methods for HFS

Method	T_r (s)	T_s (s)	Overshoot(p.u)
Fractional filter PID [221]	-	19.6	-
IMC PID	11.7	19.7	0
ALO-PID	1.338	10.71	0.082
GWO-PID	0.267	9.34	0.09
MFO-PID	1.21	10.16	0.053
ALO- FOPID	**0.58**	**0.90**	**0.004**
GWO-FOPID	0.955	1.817	0.004
MFO-FOPID	0.944	1.472	0.009
PSO-FOPID	0.551	2.394	0.309
PSOGSA-FOPID	0.85	1.4	0.007

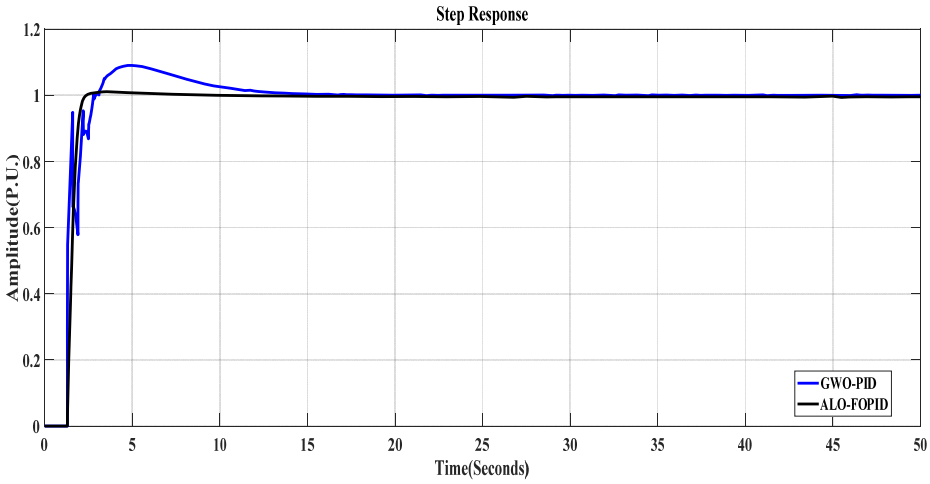

Fig.4.12 Comparative transient analysis of best of PID and FOPID controlled Heat Flow System

4.4.3 Bode Analysis

The frequency response analysis by using bode plot for PID- HFS and FOPID-HFS system is shown in Figs. 4.13 and 4.14. Fig. 4.13 represents bode plot of the IMC-PID, FO-filter PID, ALO-PID, GWO-PID and MFO-PID for the Heat Flow System. From, Fig. 4.13, it is clear that all the system tuned by above methods are stable. However, the IMC-PID gives the best value for peak gain, ALO-PID gives the best value for phase margin and FO-filter PID gives best value for the gain margin among the compared methods. Fig 4.14 represents bode plot of the FOPID-HFS tuned by GWO, ALO, MFO, PSO and PSO-GSA algorithm. It is clear from Fig. 4.14 that all the system tuned by above methods are stable. PSO-GSA-FOPID controller gives best value for peak gain and gain margin; however GWO-FOPID gives best for phase margin. In Table 4.12, delay margin, phase margin and peak gain are presented for the HFS tuned by different methods. From bode plot, the minimum peak gain, maximum phase margin and delay margin are obtained. Therefore we conclude that, the evolutionary algorithms result the best frequency response.

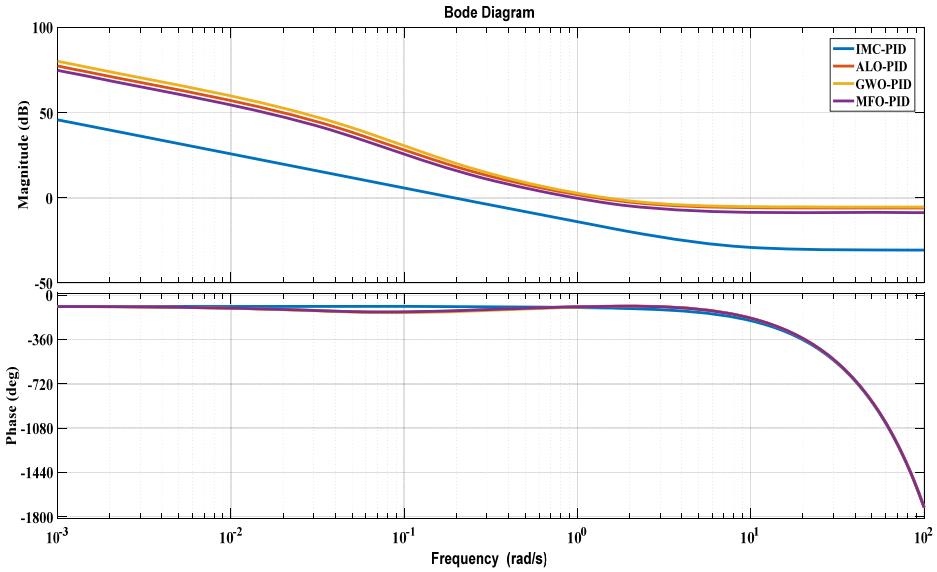

Fig.4.13 Bode Plot of the HFS with PID controller tuned by different evolutionary methods

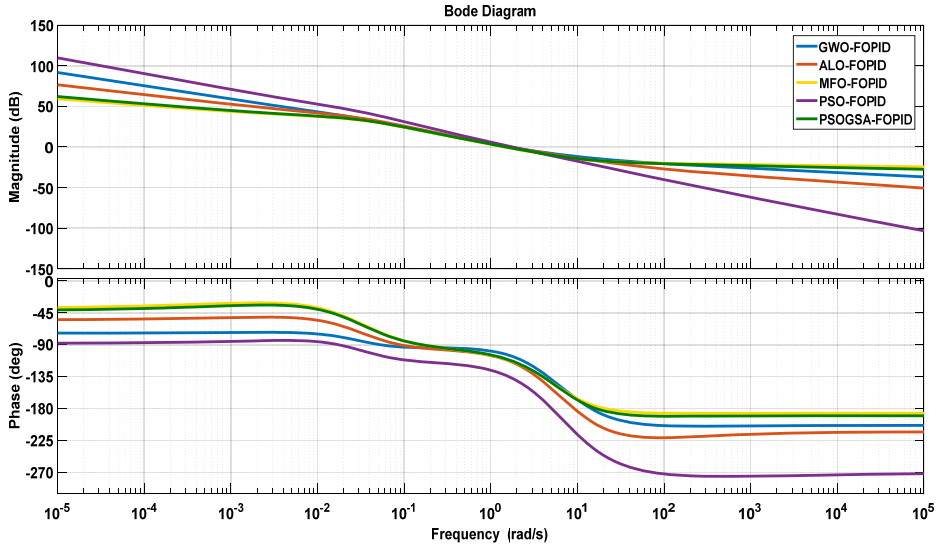

Fig. 4.14 Bode Plot of the HFS with FOPID controller tuned by different evolutionary methods

Table 4.12 Bode Analysis of various methods for HFS

Algorithms	Peak Gain(dB)	Phase Margin(deg)	Gain Margin(dB)
FO- filter+PID	102	65.1	45.95
IMC-PID	25.3	88.3	28.5
ALO-PID	77.1	90.5	5.98
GWO-PID	79.8	88.5	5.42
MFO-PID	74.5	85.4	8.71
ALO-FOPID	76.6	66.1	13.2
GWO- FOPID	91.8	74.4	13.6
MFO-FOPID	59.4	70.1	17.7
PSO-FOPID	110	42.3	10.8
PSOGSA-FOPID	62.2	70	16.6

4.4.4 Robustness Analysis

The robustness of the proposed system is investigated by considering the norm evaluation and disturbance rejection. The norm measures the robustness of a system. The norm value of the proposed closed loop system is 0.9923. Hence, the proposed controller is robust. The disturbance rejection behavior of the proposed controller has been verified by injecting an impulse signal of magnitude 10% of reference at the output of the system. At time *t=4*sec, the disturbance is injected and it has been observed that the disturbance has persisted up to *t=4.05* sec at output side of the HFS system. Therefore, the system rejects the external disturbance and returns to its normal position rapidly.

4.5 Summary

This chapter introduces an application of optimization algorithms such as Particle Swarm Optimization, Grey Wolf Optimization, Ant Lion Optimization, Moth Flame Optimization and PSO-GSA for the design and analysis of FOPID controller for heat flow system. The above algorithms are deployed to optimize different objective functions. The important contribution of the chapter includes

i First, a comparative analysis has been presented by considering four objective functions such as IAE,ISE, ITAE and ITSE. It has been observed from the simulation that the ALO-ITAE gives superiority values.

ii The heat flow system optimized by ALO performs promisingly better in terms of settling time, overshoot and steady state error in comparison with other methods.

iii Further, performance of the proposed methods supported by the transient analysis and frequency response analysis has been shown.

In the next chapter, the PID and FOPID control design has been presented for the Automobile cruise control system.