Chapter 6

Multiobjective optimization : Swarm Intelligence

In this chapter, Pareto dominance based multiobjective particle swarm optimization (MOSPOS) algorithm is used to optimize parameters of 2DOF controller. The MOPSO algorithm is implemented in the software MATLAB and results are discussed.

6.1 Introduction

Particle Swarm Optimization (PSO) algorithm falls under the category of swarm intelligence. In swarm intelligence an intelligent behavior is created by particles (like bird, fish, ant etc.) in the swarm. The level of an intelligent achieved by cooperation of all members of the swarm is so high that, it is not possible to reach at that level by individual member of swarm. PSO is very simple to implement and has high speed of convergence hence, it's proposed for multiobjective optimization problems [62]. Probably, the first idea of using PSO for multiobjective optimization was proposed by Moore and Chapman in an unpublished document in the year 1999. The algorithm was based on Pareto dominance of multiobjective optimization. Advances in multiobjective optimization PSO algorithms was discussed in the chapter 2 of literature review.

Here, 2DOF controller parameters of feed forward type structure was optimized using particle swarm optimization and Pareto dominance based multiobjective particle swarm optimization algorithm. The comparison of results are provided in following sections in the form of graphs and tabulated in tables.

6.2 Algorithm for tuning 2DOF controller using particle swarm optimization

Step 1: Define the dimension of 2DOF controller parameters optimization problem (number of decision variables **'NVARS'** =5).
Step 2: Set the upper bound values UB= [100 100 100 1 1] & lower bound values LB= [0 0 0 0 0]. M Araki et al. [3] has tested different processes using 2DOF controller optimization and maximum value of any parameters of $C_s(s)$ is not greater than **'60'** hence, for safe side maximum value in $C_s(s)$ is selected to be **'100'**.
Step 3: Derive transfer function of plant **'plant'**, actuator **'actuator_tf'**, sensor **'sensor_tf'**, temperature disturbance **'distb_temp'**, flow disturbance **'distb_flow'**, serial controller **'C'**, and feed forward controller **'C_f'**.
Step 4: Define the step magnitude of input, flow disturbance and temperature disturbance as **1**, **0.1**, and **0.01** respectively [59].
Step 5: Initialize the PSO parameters: Maximum Number of Iterations **'100'**, Population Size **'200'**, Intertia Coefficient **'0.5'**, Damping ratio of intertia Coefficient **'1'**,Social Acceleration Coefficient **'0.5'**,Personal Acceleration Coefficient **'0.5'**.
Step 6: Call function PSO for optimization of 2DOF controller parameters.
Step 7: The algorithm halts when reached maximum number of iteration.

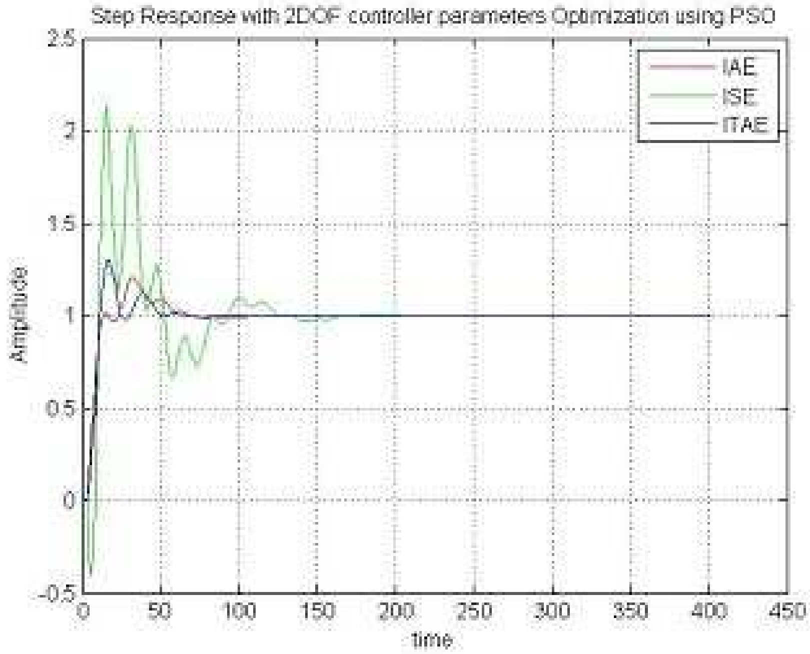

Figure 6.1: Step response of 2DOF controller optimization using PSO.

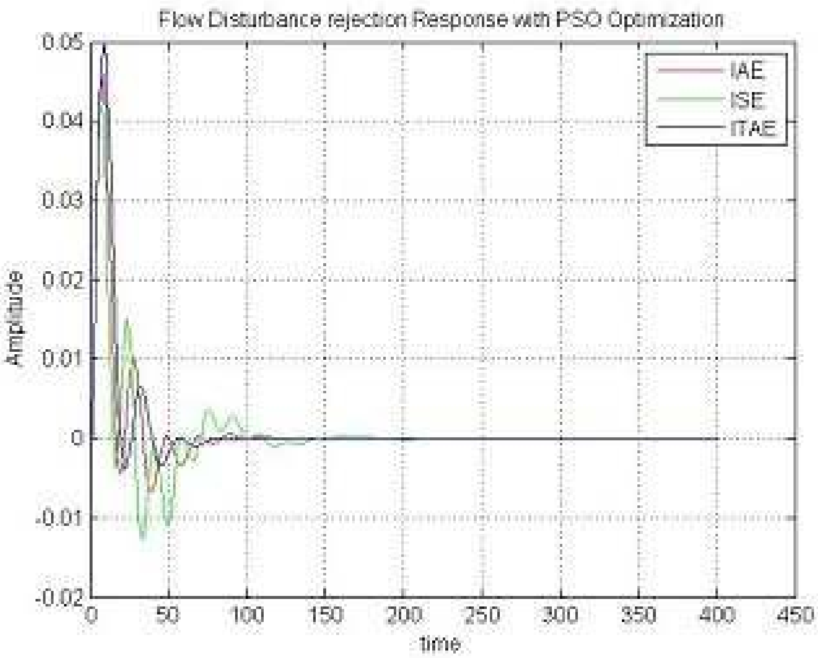

Figure 6.2: Flow disturbance response of 2DOF controller optimization using PSO.

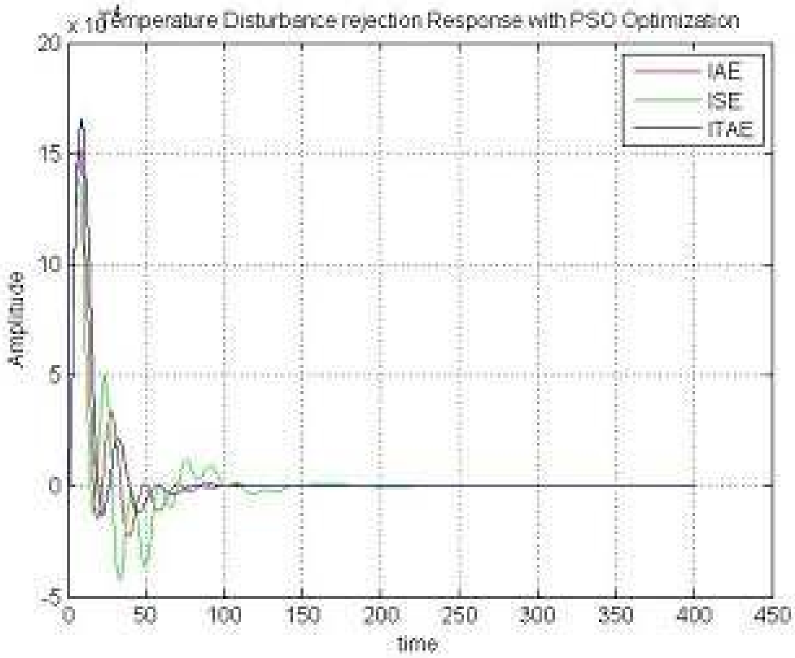

Figure 6.3: Temperature disturbance response of 2DOF controller optimization using PSO.

Table 6.1: Result of 2DOF controller parameter optimization using PSO.

PSO optimization of 2DOF controller parameters $[K_p, K_i, K_d, \alpha, \beta]$	Peak overshoot of of Step Response In (%)	Reduction Flow Disturbance Response In (%)	Reduction Temperature Disturbance Response In (%)
IAE $[0.530, 0.017, 3.399, 0.094, 0.0068]$	15.17	7.4	69
ISE $[0.854, 0.042, 5.242, 0.424, 0.0268]$	17.39	24.1	75
ITAE $[0.979, 0.048, 3.904, 0.700, 0.0308]$	1.22	21.6	74

From the Figure 6.1 ,6.2, 6.3 and parameters tabulated in Table 6.1, it is concluded that ITAE criterion for optimizing simultaneously all the five parameters of 2DOF controller using PSO method has minimum peak overshoot of step response(1.22%) and maximum reduction of flow disturbance(21.6%). The maximum reductions of temperature (75%) disturbances is obtained under the criterion of ISE compared to other two criteria IAE and ITAE. Here, results are obtained using single objective optimization by assigning equal weights(unity) to all three objective functions under three separate evaluation criteria. As being multiobjective optimization problem, it is first required to obtain multiple pareto optimal solutions and select best one using multi-criteria decision. Following section discusses prevalent multiobjective particle swarm optimization algorithm for tuning 2DOF controller parameters.

6.3 Working of multiobjective particle swarm optimization algorithm

MOPSO algorithm proposed by Carlos et al. [64] was used for optimization of 2DOF controller parameters. The algorithm is discussed step wise as under.

1. Initialize the position and speed of particle for determined maximum number.

    For i=0: **MAX**

        POP[i] = random initial values.

        VEL[i] = 0

    End

2. Evaluate each of the particles in the population.

3. Store the positions of the particles that represent nondominated vectors in the repository **REP**.

4. Generate hypercube of the search space explored so far, and locate the particles using these hyper cubes as a coordinate system where each particle's coordinates are defined according to the values of its objective functions.

5. Initialize the memory of each particle (this memory serves as a guide to travel through the search space. This memory is also stored in the repository):

    For i=0: **MAX**

        PBESTS[i] = POP[i]

End

6. **WHILE** maximum number of cycles has not been reached

DO

a) Compute the speed of each particle using the following expression:

VEL[i]= W*VEL[i]+r1*(PBEST[i]-POP[i]) +r2*(REP[h]-POP[i])

Where,

W= Inertia weight has value of 0.5

r1 and r2 are random numbers from 0 to 1.

PBEST[i] is the best position that particle 'i' has had.

REP[h] is the value taken from the repository.

The index of h is determined in following way:

Those hypercube containing more than one particle are assigned fitness equal to the result of dividing any number $x > 1$ (used x=10 in our experiments) by the number of particles that they contain. The objective of this is to reduce the fitness (as a form of fitness sharing [60]) of those hypercube that contain more particles. Then, roulette-wheel selection is applied using these fitness values to select the hypercube, from which the corresponding particle will be taken. Once the hypercube has been selected, a particle within such hypercube will be selected randomly. POP[i] is the current value of the particle 'i' (select h=i).

b) Compute the new positions of the particles adding the speed produced from the previous step.

POP[i] =POP[i] + VEL[i]

c) Keep the particles in the search space in case they go beyond these boundaries. If decision variables go beyond its search space then do either of following two things.

1. Apply lower or upper value of its corresponding boundary of decision space.
2. Multiply by -1 to its velocity so that its particle searches in the opposite direction.

d) Evaluate each of the particles in the population.

e) Update the contents of the repository and geographical representation of particles in the hypercube. This update is made by inserting all the currently nondominated locations into the repository and dominated locations are removed from the repository.

As the size of repository is limited, whenever it gets full, apply a secondary criterion for retention: those particles located in less populated areas of objective space are given priority over those lying in highly populated areas.

f) When the current position of the particle is better than the position contained in its memory, the particle's position is updated using PBEST[i] = POP[i]. The criterion to decide what position from memory should be retained is based on Pareto dominance (i.e., if the current position is dominated by the position in memory, then the position in memory is kept; otherwise, the current position replaces the one in memory; if neither of them is dominated by the other, then select one of them randomly).

g) Increment loop counter.

7. END WHILE

The concept of external repository and use of mutation operator are introduced discussed below:

External repository:

The main objective of an external repository (or archive) is to keep a historical record of the nondominated vectors found along the search process. The external repository consists of two main parts: the archive controller and the grid. These two components are discussed as under.

1. The Archive Controller: The function of the archive controller is to decide whether a certain solution should be added to the archive or not. The decision-making process is as follow.

The nondominated vectors obtained at every iteration in population of algorithm is compared (on a one-to-one basis) with respect to the contents of the external repository which, at the beginning of the search will be empty. Initially, current solution is accepted as non-dominated solutions as archive is empty. If this solution is dominated by an individual within the external archive, then this solution is removed. Otherwise, if none of the elements contained in the external population dominates the solution wishing to enter, then such a solution is stored in the external archive. If there are solutions in the archive that are dominated by the new element, then such solutions are removed from the archive. If the external population reaches its maximum allowable capacity, then the adaptive grid procedure is called.

2. Adaptive Grid procedure: The main objective of an adaptive grid procedure to accommodate all the nondominated solutions in the archive. If the solution which is

required to be inserted in the external archive lies outside the current bounds of the grid, then grid is recalculated to accommodate the solution. The advantage of adaptive grid is that its computational cost is lower than niching [64].

Mutation Operator:

PSO has very high convergence speed in comparison with other search and optimization algorithms. High convergence speed may be dangerous in the context of multiobjective optimization, because a PSO-based algorithm may converge to a false Pareto front. This drawback of the PSO motivated to introduce mutation operator. Mutation operator is applied to all the particles initially to provide high exploration at the initial level and then, effect of mutation operator is decrease as iteration increases [64].

6.4 Implementation of algorithm and comparison of results

The proposed steps for 2DOF controller parameters optimization using MOPSO algorithm is as under.

Step 1: Define the dimension of 2DOF controller parameters optimization problem (number of decision variables **'NVARS'** =5).

Step 2: Set the upper bound values UB= [100 100 100 1 1] & lower bound values LB= [0 0 0 0 0]. M Araki et al. [3] has tested different processes using 2DOF controller optimization and maximum value of any parameters of $C_s(s)$ is not greater than **'60'** hence, for safe side maximum value in $C_s(s)$ is selected to be **'100'**.

Step 3: Derive transfer function of plant **'plant'**, actuator **'actuator_tf'**, sensor **'sensor_tf'** , temperature disturbance **'distb_temp'**, flow disturbance **'distb_flow'** , serial controller **'C'**, and feed forward controller **'Cf'**.

Step 4: Define the step magnitude of input, flow disturbance and temperature disturbance as **1**, **0.1**, and **0.01** respectively [59].

Step 5: Initialize swarm of population size **'nPop'** =100 represents randomly generated solutions where, each solution has its current position **(pop(i).Position=unifrnd(VarMin,VarMax,VarSize))**; and velocity with zero value **(pop(i).Velocity=zeros(VarSize))**.

Step 6: Initialize MOPSO parameters: Size of repository **'nRep'**=100, Maximum Number of Iterations **'MaxIt'** =100, Inertia Weight **'w'** =0.5, Inertia Weight Damping Rate **'wdamp'**=0.99, Personal Learning Coefficient **'c1'**=1, Global Learning Coefficient **'c2'**=2, Number of Grids per Dimension **'nGrid'**=7, Inflation Rate **'alpha'**=0.1, Leader Selection Pressure **'beta'** =2, Mutation rate **'mu'** (varied from 0.1 to 0.9) , Deletion Selection Pressure **'gamma'** =2. Formulate problem with a vector of three objectives.

Step 7: Evaluate the objective function based on current position. The performance indices considered for evaluation of objective functions are Integral of Absolute Error (IAE), Integral of Squared Error (ISE), and Integral of Time-weighted Absolute Error (ITAE) one at a time.

Step 8: Rank the population according to solution of objective functions based on nondominated sorting approach front wise **pop=DetermineDomination(pop)**. Nondominated solutions (particles) are stored in an external repository **'rep'** (or archive) **rep=pop([pop.IsDominated])**.The **'rep'** has two control components: an archive controller and a grid. An archive controller dictates if a solution should be added to the **'rep'** or not. At each iteration of the algorithm, the nondominated solutions are compared one by one to the solutions in the **'rep'**. If the new solution is dominated by any member of the **'rep'**, the solution then will be discarded; otherwise, the new solution will be added to the **'rep'**. After adding the new solution, if there are any solutions in the **'rep'** dominated by the new solution, those solutions will be discarded. Since the capacity of the **'rep'** is limited, when it reaches its limit, the decision about adding a new solution is made using the adaptive grid.

Step 9: Create grid using concept of hypercube **Grid=CreateGrid(rep, nGrid, alpha)**. The main objective of the adaptive grid is to have well distributed Pareto fronts. The space of objectives is divided into regions that would be changed depending on the solutions in the **'rep'**, i.e., when a new solution is found outside of the current grid, the grid will be updated and the individual within it will be relocated. One can consider the grid as connected hyper cubes and each hypercube can have some individuals in it while some of them may be empty.

Step 10: Update personal best **'pop(i).Best.Position'** and global best **'leader.Position'**. Personal best represents the best position found by particle **'i'** up to iteration **'t'**. Global best represents the best solution found by swarm up to iteration **'t'**.

Step 11: While **it** < **MaxIt** do

for **each particle(i)** do

Update Velocity

pop(i).Velocity = w*pop(i).Velocity...

+c1*rand(VarSize).*(pop(i).Best.Position-pop(i).Position)...

+c2*rand(VarSize).*(leader.Position-pop(i).Position)...;

Update New Position

pop(i).Position = pop(i).Position + pop(i).Velocity;

Update personal best

pop(i).Best.Position

Apply mutation operator

This intends to produce a highly explorative behavior in the algorithm. As the number of iterations increases, the effect of the mutation operator decreases. This covers the full range of each design variable at the beginning of the search and then narrow the range covered over time, using following nonlinear function.

pm=(1-(it-1)/(MaxIt-1))^(1/mu);

Calculate new position of particle applying mutation function.

NewSol.Position=Mutate(pop(i).Position,pm,VarMin,VarMax);

END

Update Global best

Update repository

'w' is an inertia weight of the particle **'i'** and controls the trade-off between the global and local history. **'w'** has a large impact on the performance of MOPSO. Thus, the gradually decreasing linear inertia weight as suggested in [60] is applied as follows:

w=w*wdamp;

Where, **wdamp** = Weight Damping Rate.

it= it+ 1

END While

The following Figure 6.4 to Figure 6.6 is a plot of Pareto optimal front of optimization of three objective functions i.e. set point tracking and disturbance rejections (Both flow and temperature) obtained for all three tests criteria IAE, ISE & ITAE using MOPSO algorithm. This algorithm provides **'95'**, **'26'** and **'9'** nondominated set of solutions under test criteria IAE, ISE and ITAE respectively.

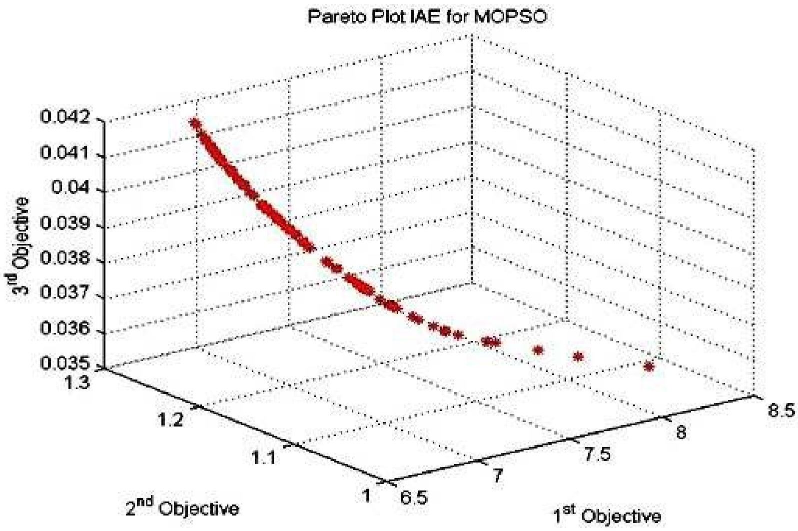

Figure 6.4: Plot of Pareto optimal front using MOPSO based optimization under IAE Criterion.

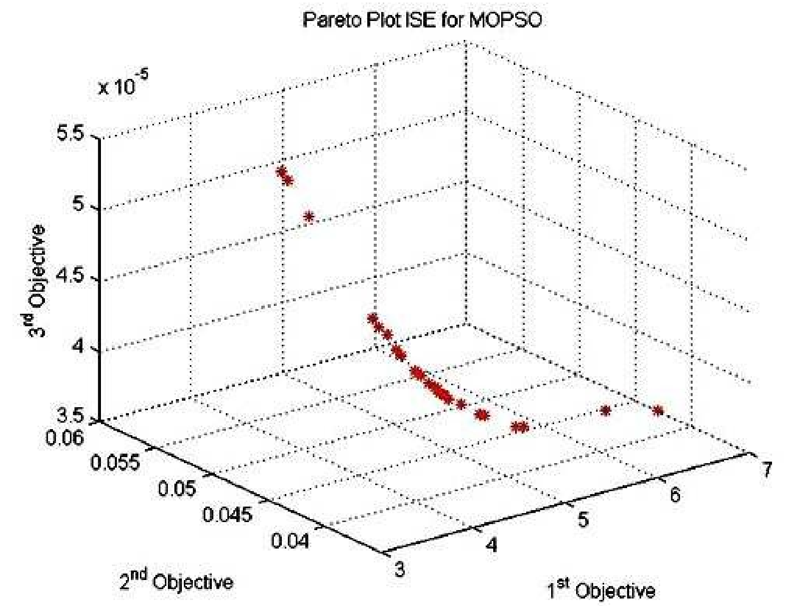

Figure 6.5: Plot of Pareto optimal front using MOPSO based optimization under ISE Criterion.

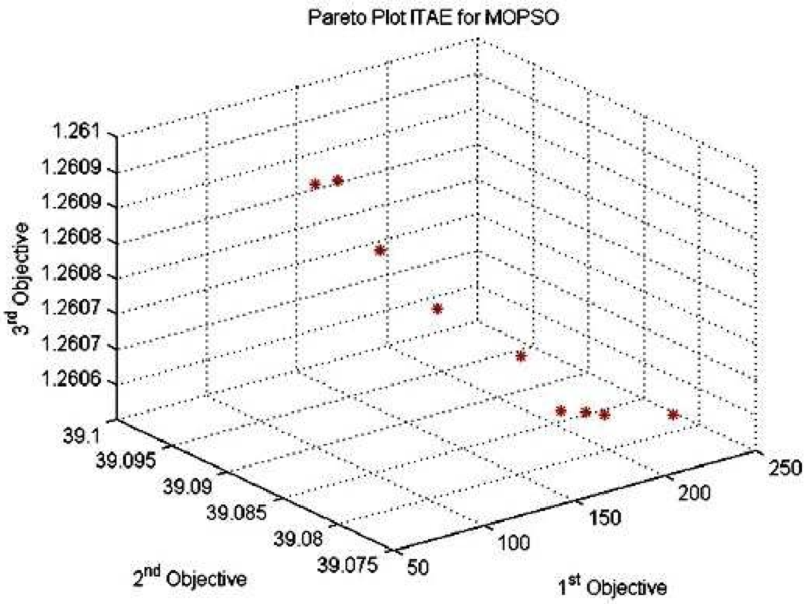

Figure 6.6: Plot of Pareto optimal front using MOPSO based optimization under ITAE Criterion.

The nondominated solutions obtained under different criteria are tabulated as under.

Table 6.2: Nondominated set of solutions obtained using MOPSO optimization under IAE criterion.

Sr.No	K_p	K_i	K_d	α	β
1	1.024007	0.048362	6.608518	0.486411	0.292564
2	1.362449	0.052017	6.854668	0.601088	0.439414
3	1.113922	0.051042	6.790711	0.527548	0.330018
4	1.149686	0.051746	6.837209	0.547475	0.363404
5	0.995107	0.046072	6.465139	0.471004	0.285858
6	1.142566	0.052012	6.811436	0.53246	0.36453
7	1.061573	0.0482	6.668703	0.489134	0.341552
8	1.253998	0.051724	6.84518	0.559015	0.399582
9	1.350209	0.051231	6.852343	0.577602	0.417164
10	1.333592	0.05177	6.839711	0.565704	0.36669

Sr.No	K_p	K_i	K_d	α	β
11	1.026934	0.048259	6.654233	0.487695	0.304087
12	1.192237	0.05188	6.76109	0.538246	0.348068
13	1.329296	0.051252	6.851525	0.57615	0.4176
14	1.233943	0.051778	6.837104	0.54654	0.369673
15	1.362205	0.052017	6.854668	0.601088	0.439581
16	1.294361	0.051796	6.837073	0.546528	0.376135
17	1.316415	0.051891	6.839923	0.563724	0.385974
18	1.267034	0.051803	6.835596	0.547901	0.380322
19	1.112907	0.051771	6.807975	0.530331	0.347839
20	1.338831	0.051908	6.844837	0.566669	0.391565
21	1.248228	0.051773	6.838224	0.562774	0.367966
22	1.276698	0.051557	6.833231	0.55156	0.410129
23	1.349177	0.05144	6.851313	0.570992	0.385513
24	0.995107	0.046072	6.465139	0.471004	0.285858
25	1.305848	0.051984	6.847643	0.556222	0.39363
26	1.163408	0.050888	6.802661	0.523014	0.339921
27	1.083258	0.052036	6.741211	0.523123	0.31271
28	1.4578	0.0782	8.3005	0.6556	0.2743
29	1.4578	0.0782	8.2291	0.6556	0.2743
30	1.4578	0.0782	8.2557	0.6556	0.2743
31	1.4578	0.0782	8.2306	0.6556	0.2743
32	1.4578	0.0782	8.6974	0.6556	0.2743
33	1.4578	0.0782	8.1086	0.6556	0.2743
34	1.4578	0.0782	8.3549	0.6556	0.2743
35	1.4578	0.0782	9.0804	0.6556	0.2743
36	1.4578	0.0782	8.0595	0.6556	0.2743
37	1.4578	0.0782	8.1048	0.6556	0.2743
38	1.4578	0.0782	8.391	0.6556	0.2743
39	1.4578	0.0782	8.2849	0.6556	0.2743
40	1.4578	0.0782	8.351	0.6556	0.2743

Sr.No	K_p	K_i	K_d	α	β
41	1.4578	0.0782	8.127	0.6556	0.2743
42	1.4578	0.0782	8.6055	0.6556	0.2743
43	1.4578	0.0782	8.7178	0.6556	0.2743
44	1.4578	0.0782	8.206	0.6556	0.2743
45	1.4578	0.0782	8.357	0.6556	0.2743
46	1.4578	0.0782	8.5947	0.6556	0.2743
47	1.4578	0.0782	8.7357	0.6556	0.2743
48	1.4578	0.0782	8.7285	0.6556	0.2743
49	1.4578	0.0782	8.4667	0.6556	0.2743
50	1.4578	0.0782	8.4651	0.6556	0.2743
51	1.4578	0.0782	8.714	0.6556	0.2743
52	1.4578	0.0782	9.3118	0.6556	0.2743
53	1.4578	0.0782	8.743	0.6556	0.2743
54	1.4578	0.0782	9.5092	0.6556	0.2743
55	1.4578	0.0782	10	0.6556	0.2743
56	1.4578	0.0782	9.1369	0.6556	0.2743
57	1.4578	0.0782	8.532	0.6556	0.2743
58	1.4578	0.0782	8.4406	0.6556	0.2743
59	1.4578	0.0782	8.3803	0.6556	0.2743
60	1.4578	0.0782	8.1565	0.6556	0.2743
61	1.4578	0.0782	8.1339	0.6556	0.2743
62	1.4578	0.0782	8.1763	0.6556	0.2743
63	1.4578	0.0782	8.1989	0.6556	0.2743
64	1.4578	0.0782	8.1558	0.6556	0.2743
65	1.4578	0.0782	8.866	0.6556	0.2743
66	1.4578	0.0782	8.8104	0.6556	0.2743
67	1.4578	0.0782	8.9687	0.6556	0.2743
68	1.4578	0.0782	8.0846	0.6556	0.2743
69	1.4578	0.0782	8.2562	0.6556	0.2743
70	1.4578	0.0782	8.8579	0.6556	0.2743

Sr.No	K_p	K_i	K_d	α	β
71	1.4578	0.0782	8.3368	0.6556	0.2743
72	1.4578	0.0782	8.8096	0.6556	0.2743
73	1.4578	0.0782	9.6934	0.6556	0.2743
74	1.4578	0.0782	9.5073	0.6556	0.2743
75	1.4578	0.0782	8.8468	0.6556	0.2743
76	1.4578	0.0782	9.2695	0.6556	0.2743
77	1.4578	0.0782	9.032	0.6556	0.2743
78	1.4578	0.0782	8.3956	0.6556	0.2743
79	1.4578	0.0782	10	0.6556	0.2743
80	1.4578	0.0782	8.7656	0.6556	0.2743
81	1.4578	0.0782	8.0783	0.6556	0.2743
82	1.4578	0.0782	9.0854	0.6556	0.2743
83	1.4578	0.0782	8.9406	0.6556	0.2743
84	1.4578	0.0782	8.6383	0.6556	0.2743
85	1.4578	0.0782	8.3928	0.6556	0.2743
86	1.4578	0.0782	8.6942	0.6556	0.2743
87	1.4578	0.0782	8.3351	0.6556	0.2743
88	1.4578	0.0782	10	0.6556	0.2743
89	1.4578	0.0782	8.429	0.6556	0.2743
90	1.4578	0.0782	8.1937	0.6556	0.2743
91	1.4578	0.0782	8.1771	0.6556	0.2743
92	1.4578	0.0782	8.3738	0.6556	0.2743
93	1.4578	0.0782	8.4125	0.6556	0.2743
94	1.4578	0.0782	8.4438	0.6556	0.2743
95	1.4578	0.0782	8.4436	0.6556	0.2743

Table 6.3: Nondominated set of solutions obtained using MOSPO optimization under ISE criterion.

Sr.No	K_p	K_i	K_d	α	β
1	1.3332	0.0817	9.5442	0.5872	0.2027
2	1.2981	0.0817	9.2933	0.5872	0.2039
3	1.3405	0.0817	9.0877	0.5872	0.196
4	1.9041	0.0817	10	0.5872	0.302
5	2.1724	0.0817	10	0.5872	0.2085
6	1.498	0.0817	10	0.5872	0.2369
7	1.4781	0.0817	10	0.5872	0.2465
8	1.9601	0.0817	10	0.5872	0.4004
9	1.6142	0.0817	10	0.5872	0.2775
10	1.5943	0.0817	10	0.5872	0.2917
11	1.6623	0.0817	9.9682	0.5872	0.2849
12	1.7036	0.0817	10	0.5872	0.2921
13	1.7949	0.0817	10	0.5872	0.2671
14	1.6028	0.0817	10	0.5872	0.2776
15	1.5978	0.0817	10	0.5872	0.2857
16	1.8026	0.0817	10	0.5872	0.2569
17	1.552	0.0817	10	0.5872	0.2665
18	1.5407	0.0817	9.9663	0.5872	0.2697
19	1.7352	0.0817	10	0.5872	0.2648
20	1.457	0.0817	10	0.5872	0.2455
21	1.6766	0.0817	10	0.5872	0.2843
22	1.665	0.0817	10	0.5872	0.2722
23	1.6396	0.0817	10	0.5872	0.273
24	1.9154	0.0817	10	0.5872	0.2571
25	1.6835	0.0817	10	0.5872	0.2625
26	1.688	0.0817	10	0.5872	0.3012

Table 6.4: Nondominated set of solutions obtained using MOPSO optimization under ITAE criterion.

Sr.No	K_p	K_i	K_d	α	β
1	1.0907	0.0349	7.4041	0.0988	0.1903
2	1.0907	0.0349	6.5487	0.2188	0.3046
3	1.0907	0.0349	6.3792	0.1896	0.1318
4	1.0907	0.0349	5.6552	0.314	0.3242
5	1.0907	0.0349	7.3334	0.1425	0.326
6	1.0907	0.0349	6.7678	0.1717	0.3359
7	1.0907	0.0349	5.9232	0.0483	0.3867
8	1.0907	0.0349	5.876	0.4325	0.4014
9	1.0907	0.0349	5.7593	0.312	0.3902

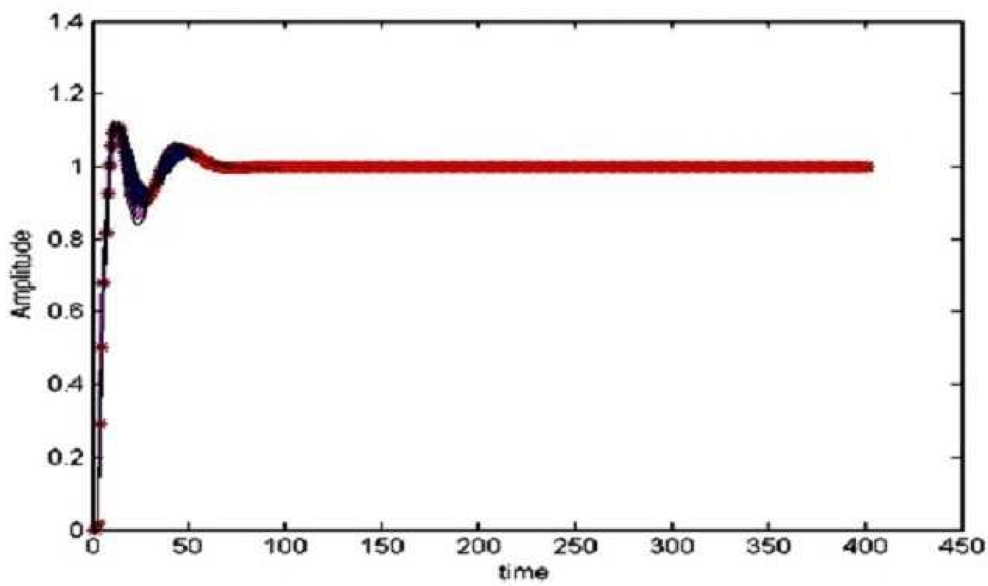

Figure 6.7: Set point response obtained using MOPSO optimization of 2DOF controller under IAE.

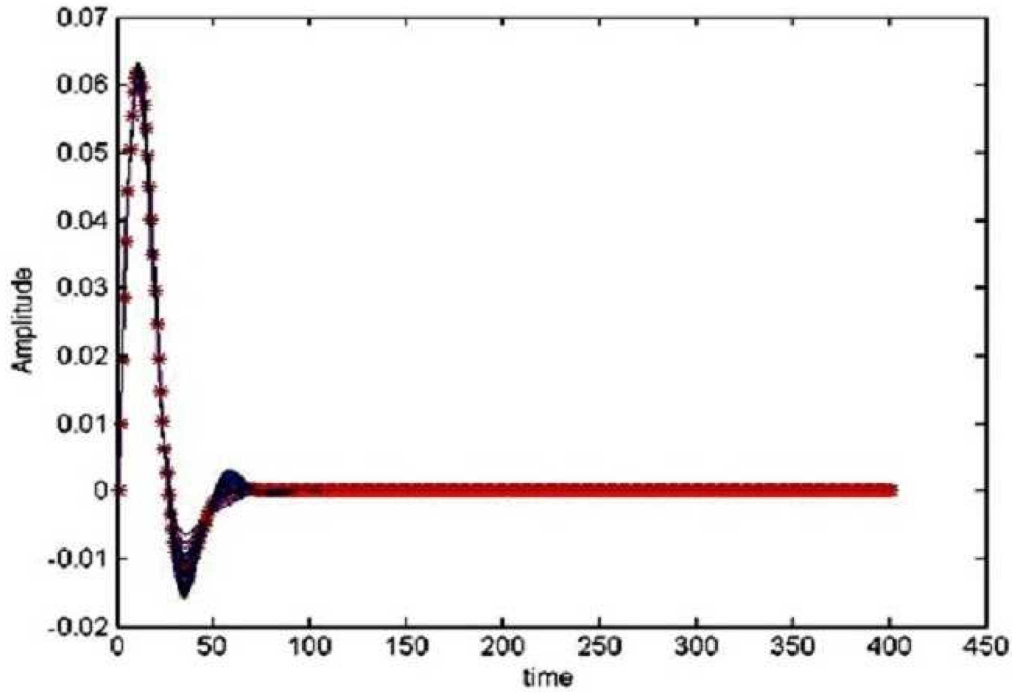

Figure 6.8: Flow disturbance rejection response obtained using MOSPO optimization of 2DOF controller under IAE.

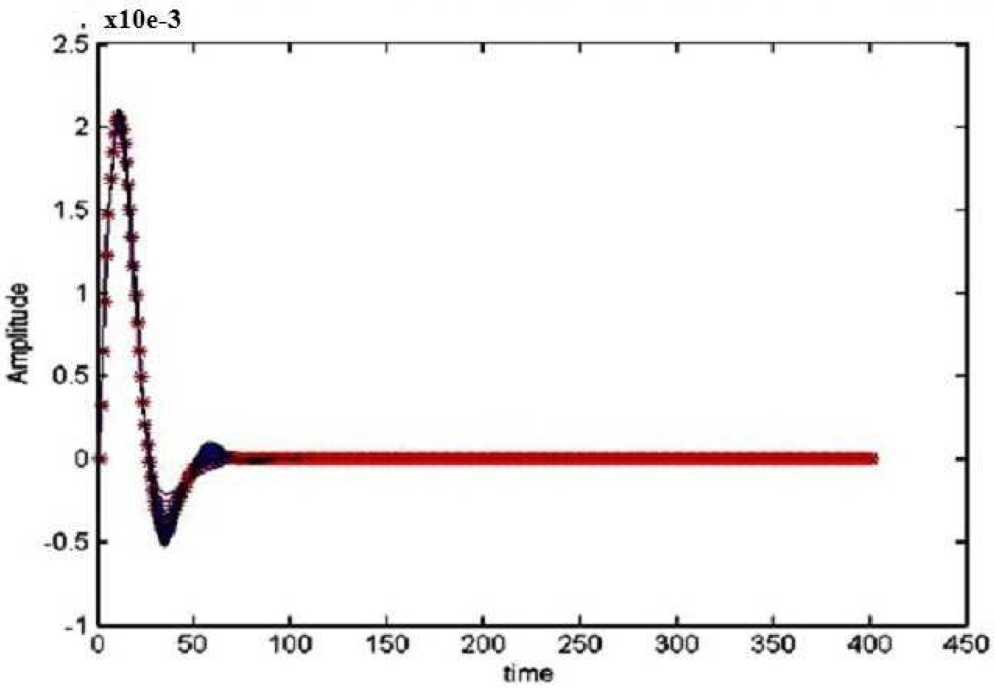

Figure 6.9: Temperature disturbance rejection response obtained using MOSPO optimization of 2DOF controller under IAE.

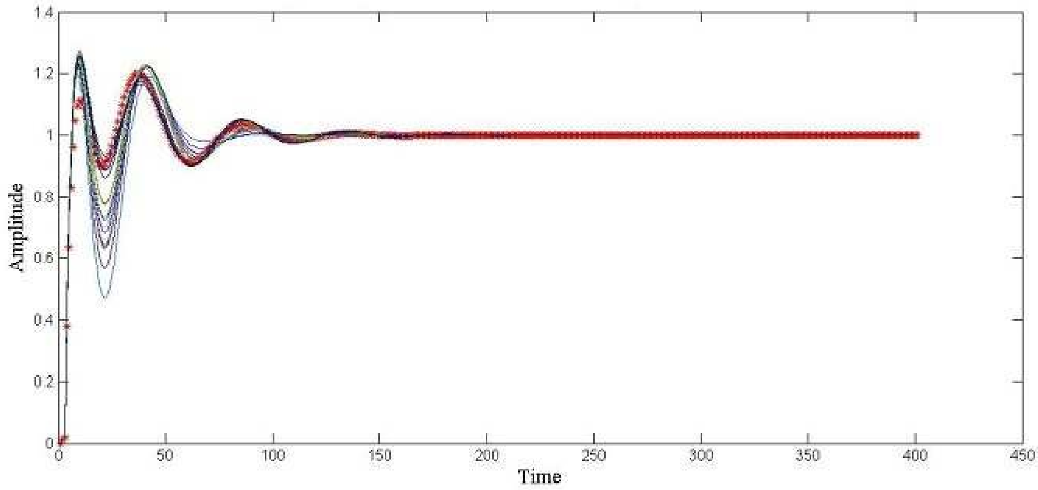

Figure 6.10: Set point response obtained using MOPSO optimization of 2DOF controller under ISE.

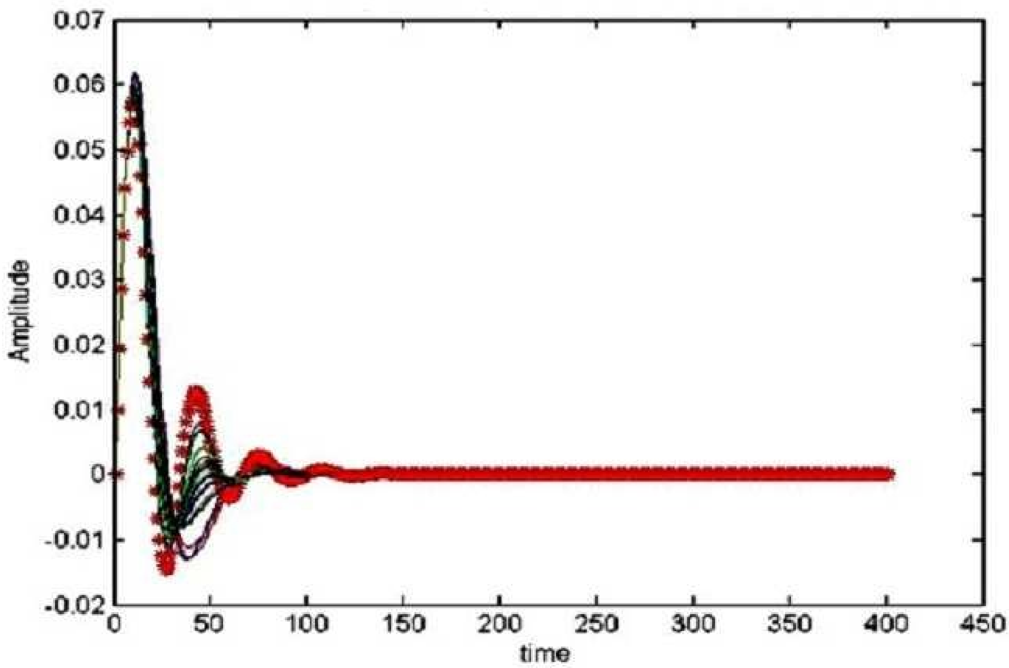

Figure 6.11: Flow disturbance rejection response obtained using MOSPO optimization of 2DOF controller under ISE.

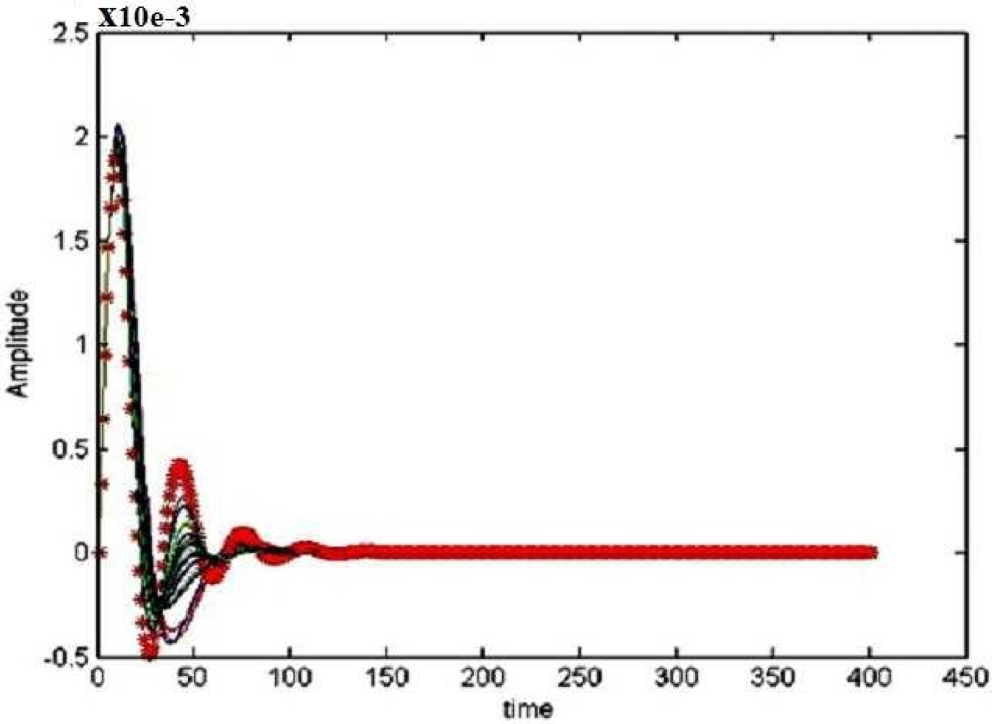

Figure 6.12: Temperature disturbance rejection response obtained using MOSPO optimization of 2DOF controller under ISE.

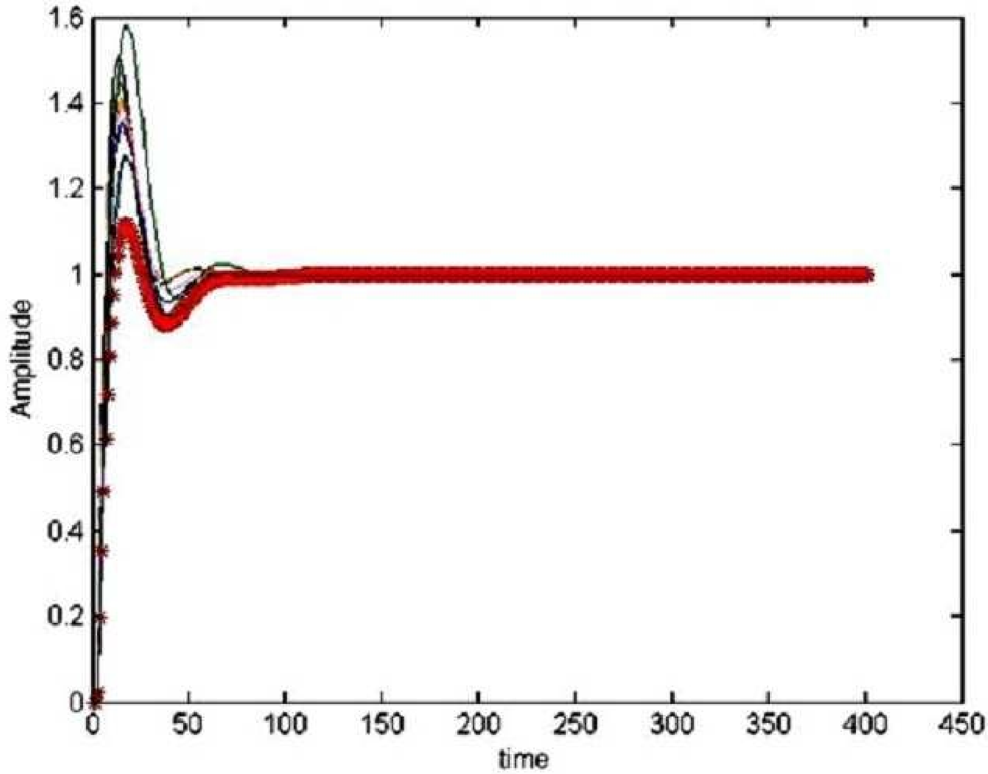

Figure 6.13: Set point response obtained using MOPSO optimization of 2DOF controller under ITAE.

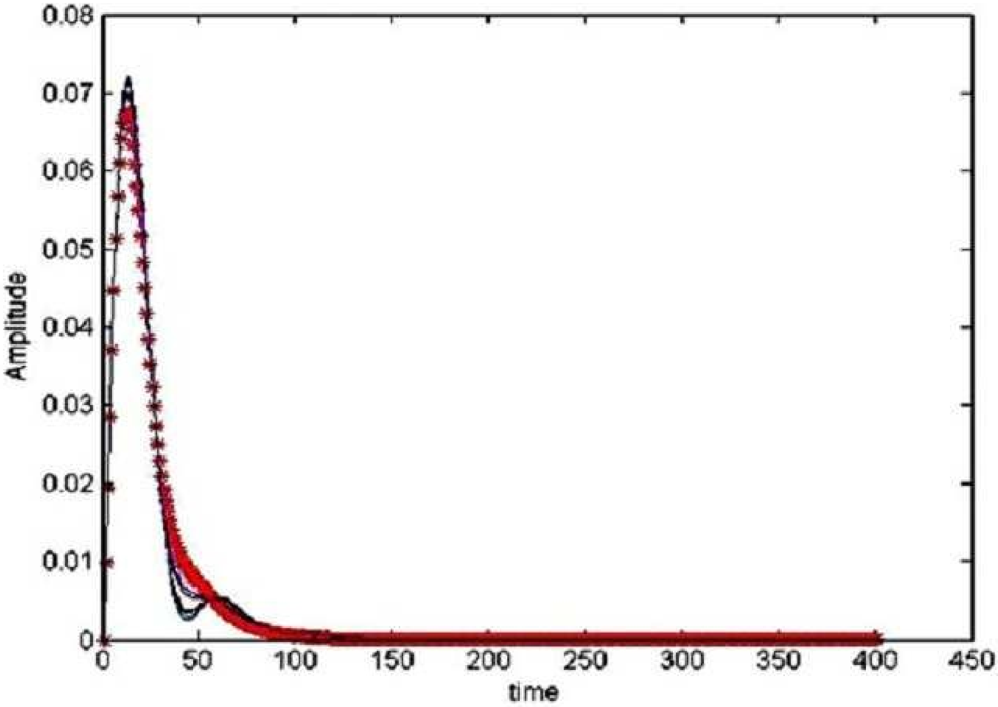

Figure 6.14: Flow disturbance rejection response obtained using MOSPO optimization of 2DOF controller under ITAE.

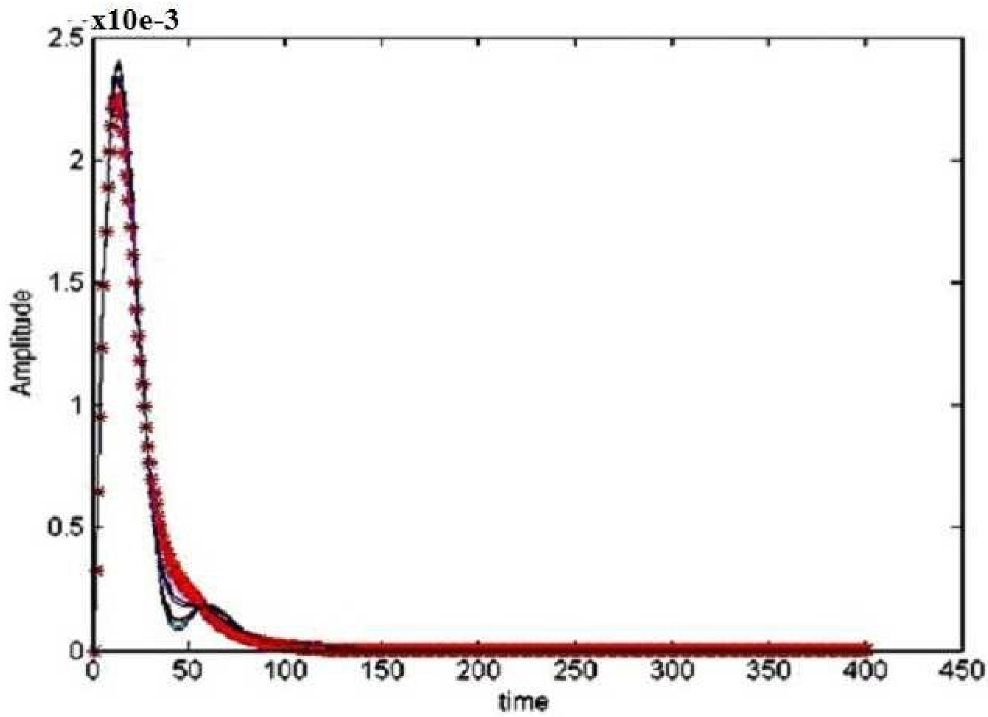

Figure 6.15: Temperature disturbance rejection response obtained using MOSPO optimization of 2DOF controller under ITAE.

The Figure 6.7 to Figure 6.15 are plots of set point tracking and disturbance rejections for all values of 2DOF controller parameters obtained under the criteria IAE, ISE and ITAE as nondominated set of solutions of multiobjective optimization of MOPSO. The MOPSO algorithm is run for different values of mutation rate starting from 0.1 to 0.9. The best solution obtained for mutation rate 0.7 for IAE and ISE criterion, while ITAE gives best solution for mutation rate 0.8. Here, the solutions are said to be best which gives more number of Pareto set and minimizes objective functions. The number of nondominated set of solutions obtained under IAE, ISE and ITAE are 95, 26 and 9 respectively. The best value obtained from the list of nondominated set of solutions are plotted as symbol * with red color. From the above Figure 6.7 to Figure 6.15, it is concluded that IAE criterion for optimizing simultaneously all the five parameters of 2DOF controller using MOPSO algorithm has minimum peak overshoot of step response(10.89%) for nondominated set of solution [1.458,0.078,8.88,0.655, 0.274] (Solution No-12). The maximum reductions of flow (42.58%) and temperature (80.86%) disturbances are obtained under the criterion of ISE for non dominated set of solutions [2.172, 0.082, 10.00, 0.587, 0.209] (Solution No-05), results are tabulated in Table 6.5.

Table 6.5: Result of 2DOF controller parameter optimization using MOPSO.

Multiobjective optimization 2DOF controller parameter MOPSO $[K_p, K_i, K_d, \alpha, \beta]$	Peak overshoot of of Step Response In (%)	Reduction Flow Disturbance Response In (%)	Reduction Temperature Disturbance Response In (%)
IAE (Sr.No-65, Table 6.2) $[1.458, 0.078, 8.88, 0.655, 0.274]$ (Best Set point tracking & disturbance rejections)	10.89	38.28	79.43
ISE (Sr.No-10, Table 6.3) $[1.594, 0.082, 10.00, 0.587, 0.29]$ (Best Set point tracking)	19.79	40.83	80.28
ISE (Sr.No-5, Table 6.3) $[2.172, 0.082, 10.00, 0.587, 0.20]$ (Best disturbances rejections)	34.58	42.58	80.86
ITAE (Sr.No-8, Table 6.4) $[1.090, 0.035, 5.876, 0.433, 0.40]$ (Best Set point tracking)	11.35	28.54	76.18
ITAE (Sr.No-1, Table 6.4) $[1.090, 0.035, 7.404, 0.099, 0.19]$ (Best disturbances rejections)	50.85	32.53	77.51

6.5 Conclusion

MOPSO optimization algorithm gives number of nondominated set of solutions called Pareto optimal solutions. Practically, user needs only one solution from the set of Pareto optimal solutions for particular problem. Generally, user is not aware of exact trade-off among objective functions. Hence, it is desirable to first obtain maximum possible Pareto optimal solutions and select best one using multi-criteria decision making technique. TOPSIS based multi-criteria decision making technique is applied to nondominated set of solutions discussed in the chapter 7.