

Development of Nature Inspired Temperature Control for Mixing Process

A DISSERTATION

Submitted in the fulfillment of requirement for the award of the degree of

Doctor of Philosophy

by

Vishal Vishnoi

(Reg. No. 15506004)

Under the supervision of

Dr. Sheela Tiwari

Associate Professor

Dept. of Instrumentation and Control Engineering,
Dr. B R Ambedkar National Institute of Technology,
Jalandhar, Punjab (144011), India

Dr. Rajesh Singla

Associate Professor

Dept. of Instrumentation and Control Engineering,
Dr. B R Ambedkar National Institute of Technology,
Jalandhar, Punjab (144011), India



Department of Instrumentation and Control Engineering

Dr. B R Ambedkar National Institute of Technology

Jalandhar-144011, Punjab, India

April, 2023

CANDIDATE'S DECLARATION

I hereby certify that the work which is being presented in the dissertation entitled "*Development of Nature Inspired Temperature Control for Mixing Process*" in the fulfilment of the requirement for the award of the degree in Doctor of Philosophy is an authentic record of my own research work carried out under the supervision of **Dr. Sheela Tiwari** and **Dr. Rajesh Singla**.

The subject matter presented in the dissertation has not been submitted elsewhere in part or fully to any other University or Institute for the award of the degree.

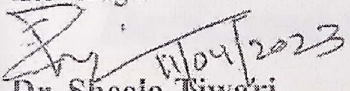

Vishal Vishnoi 11/04/2023

Roll No. 15506004

Department of Instrumentation and Control Engineering

Dr. B R Ambedkar National Institute of Technology, Jalandhar

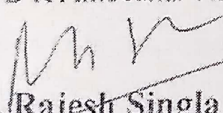
This is to certify that the above statement made by the candidate is correct to the best of our knowledge.


Dr. Sheela Tiwari 11/04/2023

Associate Professor

Department of Instrumentation and Control Engineering

Dr. B R Ambedkar National Institute of Technology, Jalandhar

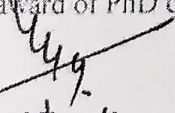

Dr. Rajesh Singla 11/04/23

Associate Professor

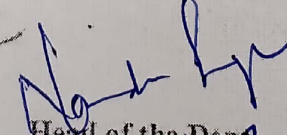
Department of Instrumentation and Control Engineering

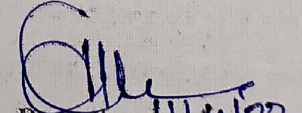
Dr. B R Ambedkar National Institute of Technology, Jalandhar

The PhD viva-voce examination of Mr. Vishal Vishnoi has been held on 11/04/23 and accepted for the award of PhD degree.


 External Examiner 11/04/2023


 Supervisor 11/04/23


 Head of the Dept 11/04/23


 Dean Academic 11/04/23

Dedicated to my beloved Family Members. . .

ACKNOWLEDGEMENTS

At this moment of accomplishment, toward the completion of strenuous and challenging research work, I am truly grateful to GOD for all that I gained and learned during these years that would benefit me in the sphere of academic, professional, and personal life. This journey of Ph.D. has been seen through to completion with the support and inspiration of numerous people, including my family, friends, colleagues, and well-wishers. It is delight to convey my gratitude to them in my humble acknowledgement.

I would like to convey my gratitude to my research supervisors, **Dr. Sheela Tiwari** and **Dr. Rajesh Singla**, Associate Professors in the Department of Instrumentation and Control Engineering at Dr. B R Ambedkar National Institute of Technology, Jalandhar, Punjab, for accepting me as a research scholar despite their many other academic and professional commitments. The work presented in this dissertation would not have been possible without their abundant help, support, and guidance. It is with immense gratitude that I acknowledge all their contributions of time, ideas, and advice to make my Ph.D. experiences productive and stimulating, despite their busy schedule and numerous responsibilities.

I would like to express my deep sense of gratitude to my RAC member, **Prof. Sathans**, NIT, Kurukshetra, for their helpful suggestions and comments during my research progress.

I express my sincere gratitude to **Prof. Binod Kumar Kanaujia**, Director, Dr. B R Ambedkar National Institute of Technology, Jalandhar, Punjab, for unparalleled and unwavering support during the course of my journey toward achieving my objectives. He encouraged and inspired me from time to time but also put in every effort to help me out. I am grateful to **Er. Narinder Singh Bhangal**, Associate Professor and Head of Department of Instrumentation and Control Engineering at Dr. B R Ambedkar National Institute of Technology, Jalandhar, Punjab, for allowing me to use the necessary facilities for carrying out my research work.

I am also thankful to all faculty, staff, and students of the Institute for extending their support in tough times. It would be difficult to name all of them in this acknowledgment, but in an endeavor to honor their efforts, I would like to thank Mr. Mahesh Sharma, **Rohit Gupta**, Anurag Sohane, Dr. Om Prakash Verma, Yamini Gogna, Nikhil Rathi, Akshay Katyal, Rahul Kumar, to name a few.

ACKNOWLEDGEMENTS

Last but not least I would like to take this opportunity to thank my family for their love, support, and blessings. First, I wish to express my warmest thanks to my loving parents, **Dr. Krishna Kumar Vishnoi** and **Mrs. Luxmi Vishnoi**, whose love, blessings, and teachings brought me this far. I thank my father for showing faith in me and giving me the liberty to choose what I desired. I would like to thank my family members, Mr. Vikas Vishnoi and Mrs. Nidhi Vishnoi, Mrs. Ruchi Vishnoi and Mr. Rajesh Vishnoi, Mrs. Rachana Vishnoi and Mr. Ankit Vishnoi, for their constant encouragement and motivation during my tough times. Also, I would like to thank my father-in-law **Mr. Krishna Chandra Vishnoi**, and mother-in-law **Mrs. Anjana Vishnoi**, who not only supported but encouraged me during the tough time of my course.

To end on a high note, I would like to express my gratitude to my lovely wife, **Mrs. Garima Vishnoi**, who has been my pillar of strength during the last few years of my research. She has shown incredible patience and unwavering support during my extended college career, always motivated and stood by me through all tough and happy situations. A happy experience with my lovely daughter **Ishi Vishnoi** as she made to forget all my work pressure, tension, and tiredness through her sweet smile and innocent words.

It will be endless to list each and every friend, relative, and well-wisher of mine who have directly or indirectly wished for my success in all endeavors. I thank one and all.

Place: Jalandhar

Date: August, 2022

Vishal Vishnoi

(15506004)

ABSTRACT

This research work presents the PID controller in a conventional scheme, a standard split range scheme, and a variable range of split range scheme for temperature control of the MISO (multiple input single output) water tank system (mixing process). A MISO system is considered for the proposed work as most practical systems comprise numerous MISO systems. Initially, in all the schemes, the controller gain parameters are tuned using a classical Ziegler-Nichols (Z-N) tuning method for the different cases under consideration within the working range of temperature, and the performances of the controller for all the considered cases are analyzed in terms of settling time. At last, the simulation results in all the control schemes are obtained using their best value of the controller gain parameters (K_p , K_i , and K_d) for the different temperature setpoints in terms of steady-state error and settling time, and compared for the same. It was observed that the variable range of split range PID controller outperformed the conventional PID controller as well as a standard split range PID controller. Investigations are also conducted on the basis of the effect of dead time in the valve, the effect of disturbance in the process, and utility consumption.

To enhance the performance of the variable range of split range PID (SR-PID) controller, this work uses nature-inspired optimization techniques such as particle swarm optimization (PSO), whale optimization algorithm (WOA), and moth flame optimization (MFO) for optimizing the gain parameters of the controller. The simulation results are obtained for the various temperature setpoints. Based on these simulation results, a comparative study is made for the controller performance using the Z-N method, PSO, WOA, and MFO algorithms on the basis of settling time. The results show that as compared to PSO and WOA, MFO based controller provides better performance than the Z-N-based controller in all the situations, i.e., the effect of dead time in the valve, the effect of the process disturbance, and utility consumption.

Further, to improve the performance of the system, this work proposes modifications in the original MFO algorithm in three phases: change in the spiral path, change in the initial population based on the opposition theory, and change in the selection of the flames for updating mechanism. A new version of the MFO algorithm is developed by combining all the modifications mentioned above. The performance of the variable range SR-PID controller using the proposed algorithms is investigated for the different temperature setpoints within the working range, and compared with

the controller performance using the original MFO algorithm in terms of settling time. To show the effectiveness of the controller using a new version of the MFO algorithm (namely, the EMFO algorithm), the investigations are carried out for the effect of system dynamics, the effect of disturbance on the process, and utility consumption, and also compared with the performance of controller using the original MFO algorithm. The results demonstrate that the EMFO algorithm-based controller provides superior performance as compared to the original MFO algorithm-based controller in all the scenarios.

A comparison between the original MFO and EMFO algorithms is made with regard to the values of fitness function and its convergence as the iterations progressed. It is shown that the proposed EMFO algorithm converged fast with respect to the fitness function, as compared to the original MFO algorithm.

Further, this work proposes an online tuning approach using the Moth flame optimization (MFO) algorithm to optimize the gain parameters of the SR-PID controller to control the temperature of the mixing process. By considering the online tuning approach, the controller gain parameters are updated continuously after a definite time interval while the actual plant is running. The controller's performance for the individual temperature setpoints is investigated in terms of settling time, and compared with the offline tuning approach with the MFO algorithm. The simulation results show a significant improvement with the online tuning approach as compared to the offline approach.

In order to further improvement in the system performance, this work uses the improved versions of the MFO algorithm-based online tuning of the controller to control the temperature of the mixing process. To study the efficacy of controller tuned online with the proposed EMFO algorithm, the investigations are carried out for all the scenarios and compared with the performance of controller tuned online using the original MFO algorithm. The EMFO algorithm-based online tuning approach provides better performance as compared to the MFO algorithm (with offline and online tuning approaches) in all the scenarios.

Before implementing the developed EMFO algorithm in the actual plant, it is necessary to investigate the performance of the same in the real environment. Therefore, in this research work, an electrical analogous model of the practical environment is simulated for investigation by considering several effects, namely imperfect insulation, density, viscosity, and compressibility.

ABSTRACT

Further, to check the effectiveness of the proposed algorithm, the controller performance using the EMFO algorithm is compared with the performance using the original MFO algorithm. The validation results show a significant improvement in the case of EMFO-based controller with an online tuning method in comparison to MFO-based controller. Furthermore, investigations are also carried out for the effect of system dynamics, process disturbance, and utility consumption. It is concluded that the developed EMFO algorithm gives superior performance in a simulated real environment paving the way for possible implementation in practical situations.

CONTENTS

Candidate's Declaration	ii
Acknowledgements	iv
Abstract	vi
Contents	ix
List of Figures	xii
List of Tables	xiv
List of Abbreviations	xix
Chapter 1 Introduction and Literature Review	1
1.1. Introduction	1
1.1.1. Process Control	1
1.1.2. Process Variables	1
1.1.3. Process Control Schemes	3
1.1.3.1. Feedback Control Scheme	3
1.1.3.2. Feedforward Control Scheme	8
1.1.3.3. Feedforward-Feedback Control Scheme	10
1.1.3.4. Advanced Process Control Schemes	11
1.1.4. Nature-Inspired Optimization Techniques	13
1.2. Literature Review	17
1.3. Research Gaps	21
1.4. Objectives	22
1.5. Dissertation Outline	22
Chapter 2 System Modelling and Control Strategies	25
2.1. System Description and Modelling	25
2.2. Control Strategies	28
2.2.1. PID controller in the Conventional Scheme	29
2.2.2. PID Controller in Split Range Scheme	33
2.3. Results and discussion	41
2.3.1. Effect of Dead Time in the Valve	44
2.3.2. Effect of Process Disturbance	47

2.3.3. Utility Consumption	49
Chapter 3 Controller Performance using Different Nature Inspired Optimization Techniques	54
3.1. Particle Swarm Optimization	56
3.2. Whale Optimization Algorithm	60
3.2.1. Encircling Prey	60
3.2.2. Bubble-Net Attacking Method	61
3.2.3. Search for Prey	62
3.3. Moth Flame Optimization	64
3.4. Comparison of the Nature-Inspired Algorithms for Tuning the Controller	67
3.4.1. Effect of Dead Time in the Valve	69
3.4.2. Effect of Disturbance in the Process	70
3.4.3. Utility Consumption	71
3.5. Improvements in MFO	74
3.5.1. Improved Moth Flame Optimization 1 (IMFO 1) Algorithm	75
3.5.2. Improved Moth Flame Optimization 2 (IMFO 2) Algorithm	76
3.5.3. Improved Moth Flame Optimization 3 (IMFO 3) Algorithm	77
3.5.4. Enhanced Moth Flame Optimization (EMFO) Algorithm	77
3.6. Results and Discussion	79
Chapter 4 Online Implementation of Enhanced MFO-Based SR-PID Controller	90
4.1. Online Tuning Approach	91
4.2. Results and Discussion	91
4.2.1. Effect of System Dynamics	99
4.2.2. Effect of Process Disturbance	101
4.2.3. Utility Consumption	103
Chapter 5 Validation in Simulated Real Environment	108
5.1. Electrical Analogous Model	109
5.2. Results and Discussion	113
5.2.1. Effect of Imperfect Insulation	120
5.2.2. Effect of Density	124
5.2.3. Effect of Combination of Imperfect Insulation and Density	128

CONTENTS

Chapter 6 Conclusion and Future Scope	137
6.1. Conclusion	137
6.2. Future Scope	138
<i>Research Outcomes</i>	140
<i>References</i>	141
<i>Appendix</i>	165

LIST OF FIGURES

Figure	Caption	Page No.
Figure 1.1	Input/output representation of a process	2
Figure 1.2	Schematic diagram of a feedback control scheme	4
Figure 1.3	Process reaction curve showing graphical representation to compute first-order transport lag model	6
Figure 1.4	Schematic diagram of a feedforward control scheme	8
Figure 1.5	Open-loop response to compute lead-lag time constants (T_1 and T_2) in feedforward tuning rules: (a) Lead must predominate in G_f ; (b) lag must predominate in G_f	9
Figure 1.6	Block diagram of a feedforward-feedback control scheme	10
Figure 1.7	Block diagram of Cascade control	11
Figure 1.8	Classical split range control for the case with two inputs and one output	12
Figure 1.9	Classification of optimization algorithms	14
Figure 2.1	Mixing process of two different water flows in water tank system	25
Figure 2.2	Schematic diagram of temperature control using the conventional PID controller (a) for increasing temperature setpoint, (b) for decreasing temperature setpoint	30
Figure 2.3	Schematic diagram of temperature control using the variable range of split range PID controller	34
Figure 2.4	Split range configuration of the controller output: (a) for increasing temperature setpoint, (b) for decreasing temperature setpoint	35
Figure 2.5	Plot of action of two valves V_1 and V_2	36
Figure 3.1	PSO inspiration: (a) Bird flocking behavior, (b) Movement of the i^{th} particles in the swarm space	56
Figure 3.2	Flow chart of PSO algorithm	59
Figure 3.3	Bubble-net attacking behavior of humpback whales	60
Figure 3.4	Bubble-net search: (a) Shrinking mechanism, and (b) Spiral updating position	62
Figure 3.5	Flowchart of WOA	63
Figure 3.6	MFO inspiration: (a) Transverse orientation, (b) Spiral flying path around a close light source	64
Figure 3.7	Logarithmic spiral and space around a flame	65
Figure 3.8	Flowchart of MFO algorithm	66
Figure 3.9	Different spirals path	75
Figure 3.10	Percentage improvement in the performances of SR-PID controller using MFO and improved MFO algorithms	79

LIST OF FIGURES

Figure 3.11	Convergence diagram for MFO and EMFO for the cases: (a) Case 5, (b) Case 7, (c) Case 10, and (d) Case 12	88
Figure 4.1	Percentage improvement in the performances of SR-PID controller using MFO and improved MFO algorithms	94
Figure 4.2	Required number of iterations after every 3 seconds of a time interval for the cases: (a) Case 2, (b) Case 3, (c) Case 4, (d) Case 5	98
Figure 5.1	Mixing Process	110
Figure 5.2	Analogous electrical system	110
Figure 5.3	Analogous electrical system with consideration of imperfect insulation	112
Figure 5.4	Required number of iterations after every 3 seconds of time interval (in case of the combined effect) for the cases: (a) Case 5, (b) Case 7, (c) Case 10, (d) Case 12	119
Figure 5.5	Heat deficit in the case of the effect of imperfect insulation in various cases: (a) 22.5°C -25.5°C, (b) 25°C -28°C, (c) 27.5°C -30.5°C, and (d) 29°C -32°C	122
Figure 5.6	Heat accumulation in the case of the effect of imperfect insulation in various cases: (a) 29°C -26°C, (b) 27.5°C -24.5°C, and (c) 25°C -23°C	124
Figure 5.7	Heat deficit in the case of the effect of density in various cases: (a) 22.5°C -25.5°C, (b) 25°C -28°C, (c) 27.5°C -30.5°C, and (d) 29°C -32°C	126
Figure 5.8	Heat accumulation in the case of the effect of density in various cases: (a) 29°C -26°C, (b) 27.5°C -24.5°C, and (c) 25°C -23°C	127
Figure 5.9	Heat deficit in the case of the combined effect in various cases: (a) 22.5°C -25.5°C, (b) 25°C -28°C, (c) 27.5°C -30.5°C, and (d) 29°C -32°C	130
Figure 5.10	Heat accumulation in the case of the combined effect in various cases: (a) 29°C -26°C, (b) 27.5°C -24.5°C, and (c) 25°C -23°C	131

LIST OF TABLES

Table	Caption	Page No.
Table 1.1	Tuning formula for PID Controller	7
Table 1.2	Feedforward control tuning parameters	9
Table 1.3	Literature survey on the temperature control, split range PID controller, and controller tuning methods	18
Table 1.4	Dissertation outline based on chapters	22
Table 2.1	Controller gains of PID controller tuned with Z-N method for all the cases under consideration	31
Table 2.2	Performance of PID controller corresponding to each set of parameters (for increasing setpoints)	32
Table 2.3	Performance of PID controller corresponding to each set of parameters (for decreasing setpoints)	32
Table 2.4	Performance of PID controller for the various temperature setpoints using the best set of parameters (Case 3 and Case 5)	33
Table 2.5 (a)	Action of valves V_1 and V_2 , when $d \leq 0$ and $Q_{10} \neq LSS$	36
Table 2.5 (b)	Action of valves V_1 and V_2 , when $d \geq 0$ and $Q_{20} \neq LSS$	37
Table 2.6	Controller gains of the standard SR-PID controller tuned with the Z-N method for all the cases under consideration	37
Table 2.7	Performance of the standard SR-PID controller corresponding to each set of parameters (for increasing setpoints)	38
Table 2.8	Performance of the standard SR-PID controller corresponding to each set of parameters (for decreasing setpoints)	38
Table 2.9	Performance of the standard SR-PID controller for the various temperature setpoints using the best set of parameters (Case 3 and Case 10)	39
Table 2.10	Controller gains of the variable range of SR-PID controller tuned with Z-N method for all the cases under consideration	39
Table 2.11	Performance of the variable range of SR-PID controller corresponding to each set of parameters (for increasing setpoints)	40
Table 2.12	Performance of the variable range of SR-PID controller corresponding to each set of parameters (for decreasing setpoints)	40
Table 2.13	Performance of the variable range of SR-PID controller for the various temperature setpoints using the best set of parameters (Case 3 and Case 5)	41
Table 2.14	Combinations of steady-state flow rates	42
Table 2.15	Performances of PID controller (in all the schemes) with different combinations of flow rates	43
Table 2.16	Comparative study of the transient response of PID controller (in all the schemes) with no dead time	44

LIST OF TABLES

Table 2.17	Comparative study of the transient response of PID controller (in all the schemes) with a dead time of 0.25 sec.	46
Table 2.18	Comparative study of the transient response of PID controller (in all the schemes) with a dead time of 0.5 sec.	46
Table 2.19	Comparative study of the transient response of PID controller (in all the schemes) with a dead time of 0.5 sec.	47
Table 2.20	Comparative study of the transient response of PID controller (in all the schemes) with a dead time of 0.25 sec. and disturbance	48
Table 2.21	Comparative study of the transient response of PID controller (in all the schemes) with a dead time of 0.5 sec. and disturbance	48
Table 2.22	Comparative study of the amount of Q_1 and Q_2 flows using PID controller (in all the schemes) with no dead time	49
Table 2.23	Comparative study of the amount of Q_1 and Q_2 flows using PID controller (in all the schemes) with disturbance and no dead time	50
Table 2.24	Comparative study of the amount of Q_1 and Q_2 flows using PID controller (in all the schemes) with a dead time of 0.25 sec.	50
Table 2.25	Comparative study of the amount of Q_1 and Q_2 flows using PID controller (in all the schemes) with a dead time of 0.5 sec.	51
Table 2.26	Comparative study of the amount of Q_1 and Q_2 flows using PID controller (in all the schemes) with a dead time of 0.25 sec. and disturbance	51
Table 2.27	Comparative study of the amount of Q_1 and Q_2 flows using PID controller (in all the schemes) with a dead time of 0.5 sec. and disturbance	52
Table 3.1	PSO algorithm parameters	58
Table 3.2	Comparative study of the transient response of SR-PID controller using classical Z-N and nature-inspired techniques with no dead time	68
Table 3.3	Performances of SR-PID controller using classical Z-N and nature-inspired techniques with a dead time of 0.25 sec.	69
Table 3.4	Performances of SR-PID controller using classical Z-N and nature-inspired techniques with a dead time of 0.5 sec	70
Table 3.5	Performances of SR-PID controller using classical Z-N and nature-inspired techniques with disturbances	71
Table 3.6	Comparative study of the amount of Q_1 and Q_2 flows using SR-PID controller with Z-N and nature-inspired optimization techniques in case of valve with no dead time	72
Table 3.7	Comparative study of the amount of Q_1 and Q_2 flows using SR-PID controller with Z-N and nature-inspired optimization techniques in case of valve with a dead time of 0.25 sec.	72
Table 3.8	Comparative study of the amount of Q_1 and Q_2 flows using SR-PID controller with Z-N and nature-inspired optimization techniques in case of valve with a dead time of 0.5 sec.	73

LIST OF TABLES

Table 3.9	Comparative study of the amount of Q_1 and Q_2 flows using SR-PID controller with Z-N and nature-inspired optimization techniques in case of valve with disturbance and no dead time	73
Table 3.10	Comparative study of the transient response of SR-PID controller using MFO and improved MFO algorithms	79
Table 3.11	Performances of SR-PID controller using MFO and EMFO algorithms with no dead time	81
Table 3.12	Performances of SR-PID controller using MFO and EMFO algorithms with a dead time of 0.25 sec.	81
Table 3.13	Performances of SR-PID controller using MFO and EMFO algorithms with a dead time of 0.5 sec.	82
Table 3.14	Performances of SR-PID controller using MFO and EMFO algorithms with disturbance and no dead time	82
Table 3.15	Performances of SR-PID controller using MFO and EMFO algorithms with a dead time of 0.25 sec. and disturbance	83
Table 3.16	Performances of SR-PID controller using MFO and EMFO algorithms with a dead time of 0.5 sec. and disturbance	83
Table 3.17	Comparative study of the amount of Q_1 and Q_2 flows using SR-PID controller with MFO and EMFO in case of valve with no dead time	84
Table 3.18	Comparative study of the amount of Q_1 and Q_2 flows using SR-PID controller with MFO and EMFO in case of valve with a dead time of 0.25 sec.	84
Table 3.19	Comparative study of the amount of Q_1 and Q_2 flows using SR-PID controller with MFO and EMFO in case of valve with a dead time of 0.5 sec.	85
Table 3.20	Comparative study of the amount of Q_1 and Q_2 flows using SR-PID controller with MFO and EMFO in case of disturbance	85
Table 3.21	Comparative study of the amount of Q_1 and Q_2 flows using SR-PID controller with MFO and EMFO in case of disturbance and valve with a dead time of 0.25 sec.	86
Table 3.22	Comparative study of the amount of Q_1 and Q_2 flows using SR-PID controller with MFO and EMFO in case of disturbance and valve with a dead time of 0.5 sec.	86
Table 4.1	Comparative study of the transient response of SR-PID controller tuned offline and online (with the various duration of the fixed interval of time) using the MFO algorithm	92
Table 4.2	Comparative study of the transient response of SR-PID controller using MFO and improved MFO algorithms	93
Table 4.3	Performances of SR-PID controller using MFO and EMFO algorithms with no dead time	99
Table 4.4	Performances of SR-PID controller using MFO and EMFO algorithms with a dead time of 0.25 second	100
Table 4.5	Performances of SR-PID controller using MFO and EMFO algorithms with a dead time of 0.5 second	100

LIST OF TABLES

Table 4.6	Performances of SR-PID controller using MFO and EMFO algorithms with disturbance	101
Table 4.7	Performances of SR-PID controller using MFO and EMFO algorithms with disturbance and a dead time of 0.25 second	102
Table 4.8	Performances of SR-PID controller using MFO and EMFO algorithms with disturbance and a dead time of 0.5 second	102
Table 4.9	Comparative study of the amount of Q_1 and Q_2 flows using SR-PID controller with MFO (offline and online) and EMFO (online) in case of valve with no dead time	103
Table 4.10	Comparative study of the amount of Q_1 and Q_2 flows using SR-PID controller with MFO (offline and online) and EMFO (online) in case of valve with a dead time of 0.25 sec.	104
Table 4.11	Comparative study of the amount of Q_1 and Q_2 flows using SR-PID controller with MFO (offline and online) and EMFO (online) in case of valve with a dead time of 0.5 sec.	104
Table 4.12	Comparative study of the amount of Q_1 and Q_2 flows using SR-PID controller with MFO (offline and online) and EMFO (online) in case of disturbance	105
Table 4.13	Comparative study of the amount of Q_1 and Q_2 flows using SR-PID controller with MFO (offline and online) and EMFO (online) in case of disturbance and valve with a dead time of 0.25 sec.	105
Table 4.14	Comparative study of the amount of Q_1 and Q_2 flows using SR-PID controller with MFO (offline and online) and EMFO (online) in case of disturbance and valve with a dead time of 0.5 sec.	106
Table 5.1	Electrical analogous quantities of thermal system	110
Table 5.2	G_1 and G_2 ranges corresponding to Q_1 and Q_2	111
Table 5.3	Practical environment-based (with the effect of imperfect insulation) performances of SR-PID controller optimized online using MFO and EMFO algorithms	114
Table 5.4	Practical environment-based (with the effect of density) performances of SR-PID controller optimized online using MFO and EMFO algorithms	114
Table 5.5	Practical environment-based (with the combined effect of imperfect insulation and density) performances of SR-PID controller optimized online using MFO and EMFO algorithms	115
Table 5.6	Performances of SR-PID controller tuned online using EMFO algorithm in case of the ideal environment and the practical environment (with the effect of imperfect insulation)	121
Table 5.7	Performances of SR-PID controller tuned online using EMFO algorithm in case of the ideal environment and the practical environment (with the effect of density)	125

LIST OF TABLES

Table 5.8	Performances of SR-PID controller tuned online using EMFO algorithm in case of the ideal environment and the practical environment (with the combined effect of imperfect insulation and density)	128
Table 5.9	Comparative study of the amount of flows using SR-PID controller tuned online with EMFO algorithm in case of the ideal environment and the practical environment (with the effect of imperfect insulation)	132
Table 5.10	Comparative study of the amount of flows (in case of disturbance) using SR-PID controller tuned online with EMFO algorithm in case of the ideal environment and the practical environment (with the effect of imperfect insulation)	133
Table 5.11	Comparative study of the amount of flows using SR-PID controller tuned online with EMFO algorithm in case of the ideal environment and the practical environment (with the effect of density)	133
Table 5.12	Comparative study of the amount of flows (in case of disturbance) using SR-PID controller tuned online with EMFO algorithm in case of the ideal environment and the practical environment (with the effect of density)	134
Table 5.13	Comparative study of the amount of flows using SR-PID controller tuned online with EMFO algorithm in case of the ideal environment and the practical environment (with the combined effect of imperfect insulation and density)	134
Table 5.14	Comparative study of the amount of flows (in case of disturbance) using SR-PID controller tuned online with EMFO algorithm in case of the ideal environment and the practical environment (with the combined effect of imperfect insulation and density)	135

LIST OF ABBREVIATIONS

ABC	Artificial Bee Colony
ACO	Ant Colony Optimization
ALO	Ant Lion Optimization
ANN	Artificial Neural Network
BA	Bat Algorithm
BFOA	Bacteria Foraging Optimization Algorithm
CM	Cauchy mutation
COMFO	Cross Moth Flame Optimization
CS	Cuckoo Search
CSA	Crow Search Algorithm
DE	Differential Evolution
DEFG	Differential Evolution Flame Generation
DFG	Dynamic Flame Guidance
DMFO	Differential Moth Flame Optimization
EMFO	Enhanced Moth Flame Optimization
FCM	Fuzzy c-means
GM	Gaussian mutation
GSA	Gravitational Search Algorithm
GSO	Glowworm Swarm Optimization
GWO	Grey Wolf Optimization
HC	Hill Climbing
HMOMFO	Hybrid Multi-Objective Moth Flame Optimization
HPSO-MFO	Hybrid Algorithm Particle Swarm Optimization-Moth Flame Optimizer
HSS	Higher Saturation State
IAE	Integral Absolute Error
IMFO 1	Improved Moth Flame Optimization 1
IMFO 2	Improved Moth Flame Optimization 2
IMFO 3	Improved Moth Flame Optimization 3
ISE	Integral Square Error
ITAE	Integral Time Absolute Error
ITSE	Integral Time Square Error
JMFO	JAYA Moth Flame Optimization
KHA	Krill Herd Algorithm
LM	Lévy mutation
LSS	Lower Saturation State

LIST OF ABBREVIATIONS

MAGO	Multidynamics Algorithm for Global Optimization
MATLAB	Matrix Laboratory
MFO	Moth Flame Optimization
MFORSFS	Moth Flame Optimization and Rough Set Theory for Feature Selection
MISO	Multiple Input-Single Output
MV	Manipulated Variable
NMS	Nelder-Mead Simplex
NSMFO	Non-Dominated Sorting Moth Flame Optimizer
OL	Orthogonal Learning
PBP	Permutation-Based Problems
PI	Proportional Integral
PID	Proportional Integral Derivative
PMU	Phasor Measurement Unit
PSO	Particle Swarm Optimization
SFLA	Shuffled frog leaping algorithm
SISO	Single Input-Single Output
SR	Split Range
SR-PID	Split Range-Proportional Integral Derivative
SSA	Salp Swarm Algorithm
TLBO	Teaching Learning-Based Optimization
UC	Unit Commitment
VCRS	Vapour Compression Refrigeration System
WCA	Water Cycle Algorithm
WOA	Whale Optimization Algorithm
Z-N	Ziegler-Nichols

Chapter 1

Introduction

This chapter covers a background of Process Control, details about the process variables, control schemes, and tuning approaches being explored are discussed. To attain improved system performance, various studies on the process, controllers, tuning methods, and nature-inspired optimization techniques have been reported in the past. The survey of the literature is structured in chronological order (year-wise). The chapter also reports a list of observed research gaps which lay the foundation for the objectives of the dissertation and provides details about the outline of the dissertation.

1.1 Introduction

1.1.1 Process control

Process control is an engineering discipline dealing with methods, structures, and algorithms to regulate the process' output within the specified range. Basically, processes can be understood as a set of actions that perform the physical or chemical transformation in which the fluid or solid materials are optimally transformed into a more usable state. In the duration of transformation, a large number of internal and external atmospheric and surrounding conditions affect the performance of the processes, and usually, these surrounding conditions are considered as process variables (such as level, flow, pressure, temperature, volume, etc.) (Singh, 2009). Process control regulates and controls these process variables in such a way as to get desired output by utilizing the best possible resources, which enhances productivity and also ensures safety (Marlin, 1995).

1.1.2 Process variables

Process variables are generally those parameters or quantities which are being monitored or controlled within the specified limit. There are four primary variables that affect industrial

processes: level, flow, pressure, and temperature. In these process variables, the temperature is the most important variable in view of safety, quality, and production. Measurement of this variable is comparatively economical (due to its low-cost sensors) and easy. But it is difficult to maintain at its desired set-point value due to its inherent slow varying nature. Taking into account the above-mentioned characteristics of temperature control, and the wide scope in the process control industries, the temperature variable can be considered to be controlled.

Process variables can be classified into two major categories:

Input variables: These variables show the effect of environmental change on the process output (Bequette, 2003). Input variables can be categorized into two types, namely, manipulated and disturbance variables.

- *Manipulated Variables:* These variables are controlled by the process operator manually or by an automatic control system in order to keep the controlled variables at the desired set-point value.
- *Disturbance Variables:* The disturbance inputs are the variables that influence the process output as they cannot be controlled.

Output variables: These variables are the process outputs that have a direct impact on the environment. Output variables are categorized as measured variables and unmeasured variables, which are defined on the basis of their measurement feasibility. Fig. 1.1 shows the input/output variables of a process.



Fig. 1.1 Input/output representation of a process.

1.1.3 Process control schemes

There are various control schemes such as feedback, feedforward, feedforward-feedback, and advanced control schemes that are used in the process industries (Coughanowr & LeBlanc, 2009, Naimi *et al.*, 2022, Reyes-Lúa & Skogestad, 2020, Saxena *et al.*, 2002; Seborg *et al.*, 2011). The description of the different elements in process control schemes is given as follows:

Process: This is the main part of the control loop, which requires the control of some specific physical parameters. Generally, a process can be described as a collection of phenomena related to some industrial sequence. The most popular processes in the process industries are temperature control (boiler, distillation column), heat exchanger, and evaporator units.

Measurement: The measurement unit measures the process variable and translates the measured quantity into an analogue representation (electrical voltage or current). This unit normally uses a sensor that conducts the initial measurement and energy conversion of a variable into equivalent electrical information.

Controller: The controller is the part within the process control loop that interprets the sensor's readings and decides the appropriate control action based on an error signal which is the difference between the reference signal (reference value of the variable) and the actual output (measured indication of the controlled variable), and provides the output as a control signal to the final control element.

Final control element: It is a part of the industrial process control loop that produces the actuating signal corresponding to the control signal generated by the controller. The most important part of the final control element is an actuator which actuates the valve in response to the control signal from a controller.

1.1.3.1 Feedback control scheme

In the feedback control scheme, input or controlling action depends on the output or change in output (controlled variable). In this type of system, the controlled variable is measured and used to estimate the error, i.e., the difference between the actual output and the desired value. The schematic of the feedback control scheme is depicted in [Fig. 1.2](#).

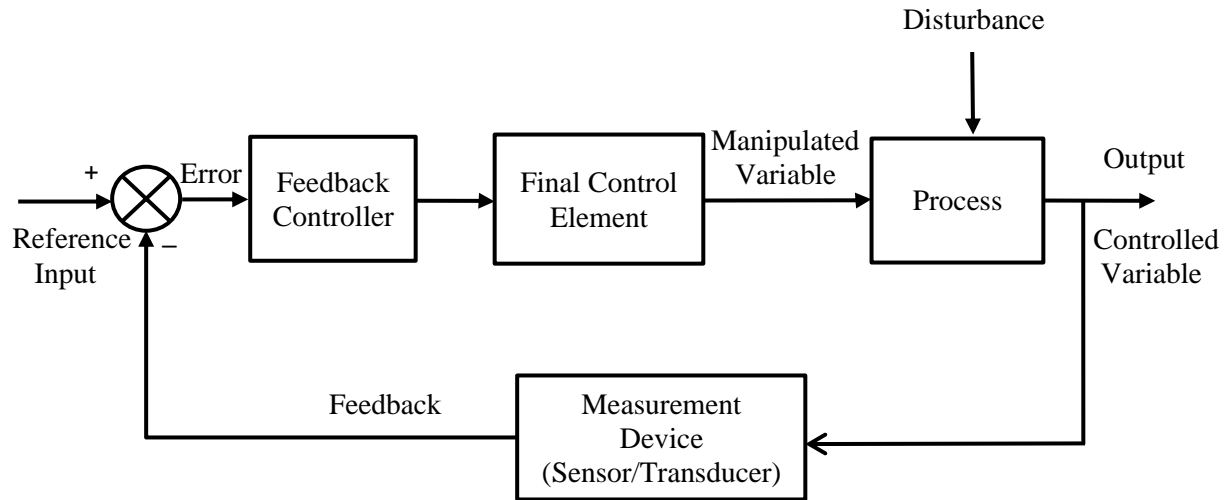


Fig. 1.2 Schematic diagram of a feedback control scheme.

Advantages of feedback control

- The controller modifies the controlled variable (influenced by disturbances) in accordance with the set-point value.
- This scheme does not necessitate the identification of any process (*i.e.*, the dynamics of the process model) and measurement of the disturbances.

Disadvantages of feedback control

- It waits until the effect of the disturbances has been felt by the system before controller action takes place.
- This control scheme is not suitable for processes having large time delays.
- This scheme may produce instability in the closed-loop response.

In the feedback control schemes, the PID controller is the most widely used controller due to its simple design, ease of implementation, and maintenance. In most of the control system applications, 90% of control modes are of PID type (Åström & Hägglund, 2001). It makes use of three elements: proportional term, integral term, and derivative term. The mathematical representation of the PID controller usually used in process control with combined action is illustrated by Eq. 1.1 (Goud & Swarnkar, 2019):

$$u(t) = K_c \left(e(t) + \frac{1}{\tau_I} \int_0^t e(t) dt + \tau_D \frac{de(t)}{dt} \right) \quad (1.1)$$

However, the same equation may be modified as [Eq. 1.2](#)

$$u(t) = K_p e(t) + K_i \int_0^t e(t) dt + K_d \frac{de(t)}{dt} \quad (1.2)$$

where,

- $u(t)$ - the control signal in the time domain
- $K_p = K_c$ - proportional gain
- $K_i = \frac{K_c}{\tau_I}$ - τ_I , reset time or integral time and K_i , integral mode gain
- $K_d = K_c \tau_D$ - τ_D , derivative time and K_d , derivative mode gain
- $e(t)$ - the error signal in the time domain

To achieve the satisfactory performance of the system, each parameter of the PID controller should be tuned.

Tuning of the controller

Tuning is referred to as determining the best possible gain parameters of a PID controller for the satisfactory response of the system. There are various **classical tuning approaches** reported in the literature such as the Trial and Error method ([Bucz & Kozáková, 2018](#)), Cohen-Coon method ([Cohen & Coon, 1953](#)), Åström and Hägglund method ([Åström & Hägglund, 1984](#)), Ziegler–Nichols method ([Ziegler & Nichols, 1942](#)), etc.

Trial and error method: This method is popularly known as empirical gain tuning which is based on the experiment and used with the feedback loop only ([Bucz & Kozáková, 2018](#)). In this method, controller settings (gain of the controllers such as K_p , K_i and K_d) are chosen based on the observation of the deviation in the system response. The trial-and-error method works on the following steps:

- Initially, the integral and derivative terms of the PID controller are set to be ideally zero by setting integral time (τ_I) very high (ideally infinite) and derivative gain (τ_D) very small (ideally zero), and the proportional gain is increased until the system's response

oscillates. This increase of proportional gain should be in such a way that the system is always bound to be in stable condition.

- Further, τ_I is adjusted by decreasing it in order to reduce the oscillation and obtain the minimum steady-state error, but at the same time, it may increase the overshoot.
- Finally, τ_D is increased to achieve the set-point quickly by minimizing the overshoot that occurred due to integral control action.

Cohen-Coon method: In this tuning method, an open-loop transient is induced by changing a step signal to the control valve, and hence, the step response is recorded at the output. The response of the system obtained is known as the process reaction curve (shown in Fig. 1.3) and usually, exhibits an S shape.

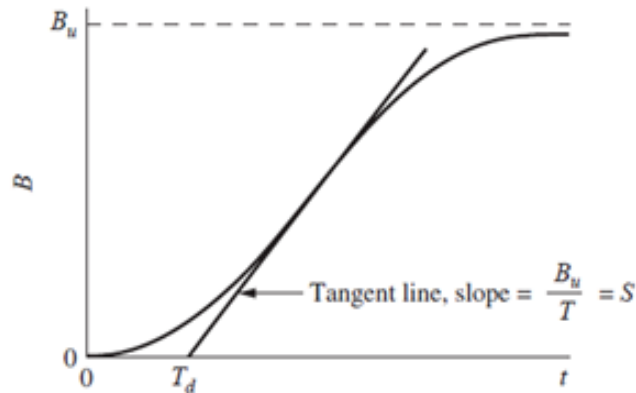


Fig. 1.3 Process reaction curve showing graphical representation for computing the first-order transport lag model. (Source: Seborg *et al.*, 2011)

To compute the values of time constant, T , and transport lag, T_d , the tangent is drawn to the curve at the point of inflection, as shown in Fig. 1.3. The intersection of the tangent line with the time axis is the transport lag, T_d , and the first-order time constant T is obtained as $T = \frac{B_u}{S}$, where, B_u is the ultimate value of B at large t , and S is the slope of the tangent line. The steady-state gain can be determined as $K_p = \frac{B_u}{M}$, where M is the magnitude of the step signal. Using the values of K_p , T , and T_d , the controller settings (K_c , τ_I , τ_d) can be determined as per Table 1.1.

Åström and Hägglund method: It is popularly known as the relay-based auto-tuning method. In this method, a push-button tuning is used to act as an ON/OFF controller. The auto-tuning

method initiates with the identification of definite and repeated oscillation patterns around the nominal value. When such a frequency response pattern is achieved, the period (T_u), amplitude (A) of its oscillation and the relay amplitude (h) for the first harmonic are computed and utilized to determine the PID controller settings (Åström & Hägglund, 1995; Hang *et al.*, 2002). The computation of ultimate gain is given by Eq. 1.3.

$$K_u = \frac{4h}{\pi A} \quad (1.3)$$

The controller settings (K_c , τ_I , τ_d) can be determined by using Table 1.1.

Table 1.1 Tuning formula for PID Controller. (George *et al.*, 2021)

Tuning methods	K_c	τ_I	τ_D
Ziegler–Nichols method	$0.6K_u$	$\frac{P_u}{2}$	$\frac{P_u}{8}$
Cohen-Coon method	$\frac{T}{K_p T_d} \left(\frac{4}{3} + \frac{T_d}{4T} \right)$	$T_d \left(\frac{32 + 6T_d/T}{13 + 8T_d/T} \right)$	$T_d \left(\frac{4}{11 + 2T_d/T} \right)$
Åström–Hägglund method	$\frac{0.67}{K_u}$	T_u	$\frac{T_u}{6}$

Ziegler–Nichols method: This method is the most widely acceptable tuning technique, developed by J.G. Ziegler and N.B. Nichols in 1942 (Ziegler & Nichols, 1942). It has the following steps for tuning the PID controller (Bequette, 2003; Ziegler & Nichols, 1942).

- Initially, the integral and derivative terms of the PID controller are set to be disabled (ideally zero) by setting, τ_I very high (ideally infinite) and τ_D very small (ideally zero).
- The proportional gain (K_p) of the controller is increased from zero to some critical value (K_u i.e., ultimate gain) at which sustained oscillations occur. The period of oscillations is noted (P_u).
- From the values of K_u and P_u obtained above, the values of controller parameters (K_c , τ_I , τ_d) can be determined as per Table 1.1.

Out of the above-discussed classical tuning methods, the Ziegler–Nichols method is widely used in industry due to its popularity and simplicity (Bharat *et al.*, 2019; Sen *et al.*, 2015). Due to

the inherent shortcomings of the feedback controller, it is essential to discuss the other control schemes.

1.1.3.2 Feedforward control scheme

The basic concept of feedforward control is to detect or measure the disturbances and control them before affecting the process and therefore, eliminate the impact of external disturbances before disturbing the process. The feedforward control scheme is illustrated in Fig. 1.4.

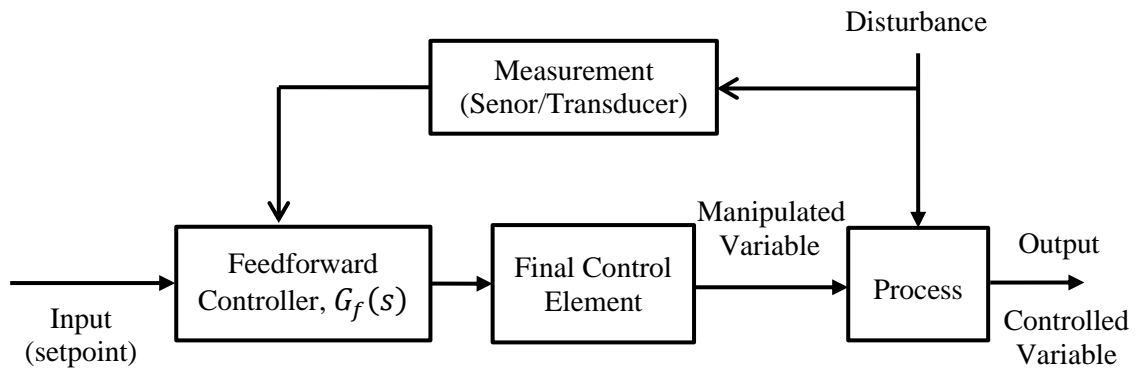


Fig. 1.4 Schematic diagram of a feedforward control scheme.

While practically realizing the feedforward controller, feedforward transfer function, $G_f(s)$ may take the form of a lead expression, such as $G_f(s) = (1 + \tau_f s)$. As realization of $G_f(s) = (1 + \tau_f s)$ is practically not possible, it is necessary to approximate $(1 + \tau_f s)$ by a lead-lag expression (Coughanowr & LeBlanc, 2009), such as

$$G_f(s) = \frac{(1 + \tau_f s)}{(1 + \beta \tau_f s)} \quad (1.4)$$

where, $\beta \ll 1$ is constant, and τ_f indicates feedforward time constant.

To employ feedforward control effectively, at least an approximate process model should be available, and all the disturbances should be accurately measured.

To describe the tuning rules for feedforward control (Coughanowr & LeBlanc, 2009), the transfer function for the feedforward controller is considered ideally in lead-lag compensator form with gain K_f , and may be written by Eq. 1.5.

$$G_f(s) = \frac{K_f(T_1s + 1)}{(T_2s + 1)} \quad (1.5)$$

where, K_f , T_1 , and T_2 are the adjustable parameters of the feedforward controller, T_1 , and T_2 are the time constants of the dynamic part of the feedforward controller, respectively. The tuning of these parameters (K_f , T_1 , and T_2) is presented below:

Initially, the value of feedforward controller gain, K_f put so as to compensate ultimately for a step-change in disturbance variable, d . This means that the dynamic portion of $G_f(s)$ in Eq. 1.5 will be removed, and only the gain K_f will remain. The lead (i.e., $T_1 > T_2$) and lag (i.e., $T_1 < T_2$) predominancy in $G_f(s)$ will be estimated by observing the open-loop response while making a step-change in the disturbance variable. Fig. 1.5 shows responses for the output, C with respect to time, t , and where t_p is the peak time. The values of T_1 and T_2 in Eq. 1.5 can be determined by Table 1.2.

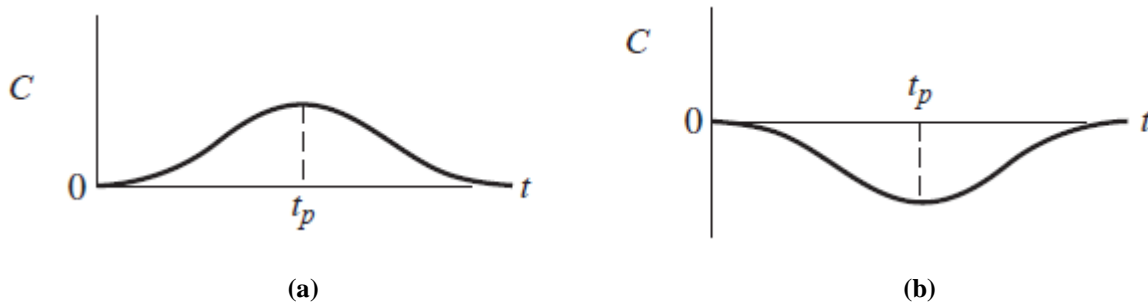


Fig. 1.5 Open-loop response to compute lead-lag time constants (T_1 and T_2) in feedforward tuning rules: (a) Lead must predominate in G_f ; (b) lag must predominate in G_f . (Source: Coughanowr & LeBlanc, 2009)

Table 1.2 Feedforward control tuning parameters.

Predominant mode	T_1	T_2
Lead	$1.5t_p$	$0.7t_p$
Lag	$0.7t_p$	$1.5t_p$

Besides the above discussion, other tuning procedures are described in the literature (Adam & Marchetti, 2004; Coughanowr & LeBlanc, 2009; Guzmán & Hägglund, 2021; Guzmána & Hägglund, 2011; Montoya-Ríos *et al.*, 2020; Rodríguez *et al.*, 2020; Seborg *et al.*, 2011; Shinskey, 1996; Veronesi *et al.*, 2017).

Advantages

- Feedforward control works before the controlled output is affected and provides superior performance with an accurate model.
- It does not cause instability in the closed-loop response.

Disadvantages

- This scheme necessitates the identification of all the disturbances and their direct measurement.
- Feedforward control requires good knowledge of the process model.

Since the feedforward controller is neither realizable nor can be used alone, it is generally used in conjunction with the feedback controller.

1.1.3.3 Feedforward-Feedback control scheme

A combined feedforward-feedback control scheme provides better performance as the limitations of feedforward control are taken care of by feedback control and vice versa. The schematic diagram of the feedforward-feedback control scheme is shown in [Fig. 1.6](#).

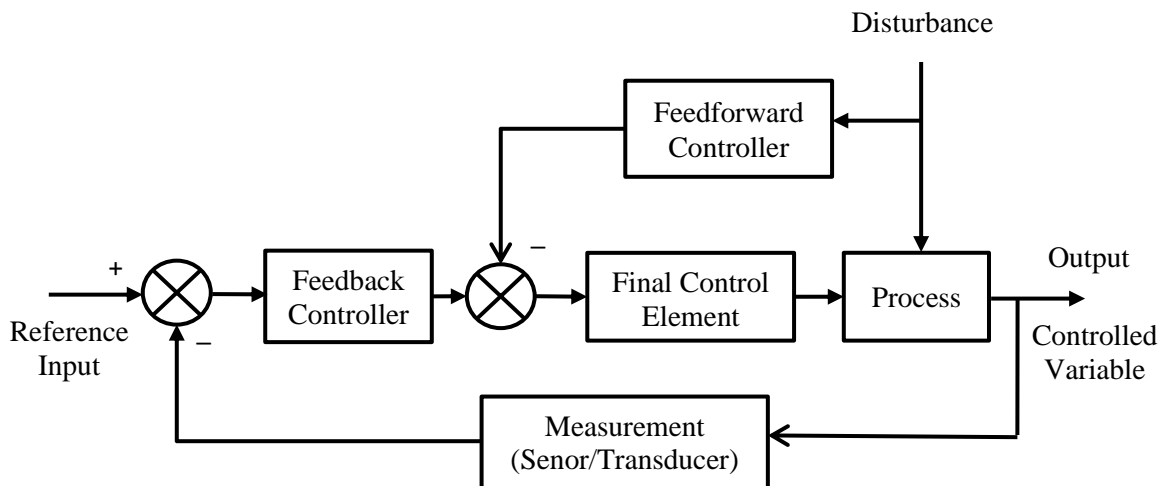


Fig. 1.6 Block diagram of a feedforward-feedback control scheme.

However, these basic control schemes provide an unsatisfactory response in the presence of process lag, large disturbances, and modelling uncertainties; therefore, advanced process control schemes need to be investigated to improve the system's response.

1.1.3.4 Advanced process control schemes

The conventional feedback control scheme consists of one measurement and one manipulated variable in a single loop. But, most of the process models have been configured with more than one measurement, manipulated variables, and loops, which make the analysis more complex and tedious. These complexities can be handled by the advanced process control schemes, which are discussed as follow:

Cascade control: The main drawback of the basic feedback control scheme is that it does not take control action until the process output variable deviates from its set point. The feedforward controller has significant improvements for processes having a large time constant or time delay in comparison to the feedback controller. However, the feedforward controller necessitates measuring the disturbances correctly as well as the availability of a model for computing the controller output. The cascade control scheme is an alternative strategy that can significantly improve the dynamic response to disturbances by utilizing a secondary feedback controller or a secondary measurement. A block diagram of the cascade control scheme is shown in Fig. 1.7.

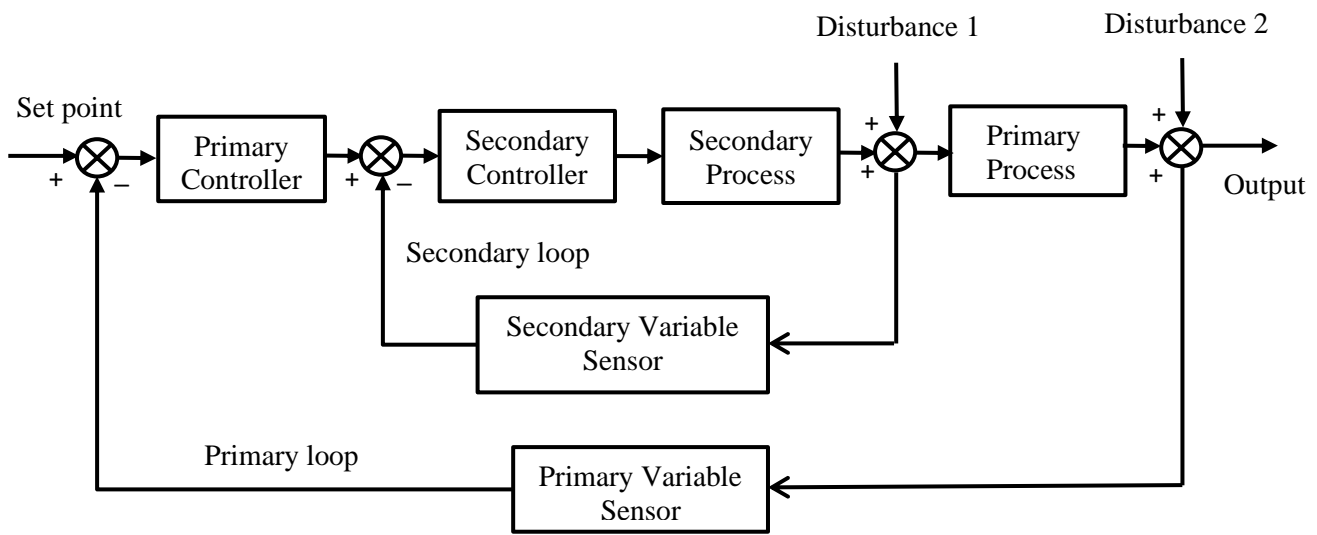


Fig. 1.7 Block diagram of Cascade control.

It consists of one manipulated variable and more than one measurement and two loops, namely, primary loop and secondary loop. It is helpful only when the secondary control loop (inner loop) has faster dynamics than the primary control loop to eliminate disturbances.

The cascade control scheme has more sensors, transmitters, and controllers. Therefore, the systems become more complex, and tuning also becomes more tedious. This control scheme has more than one controller for one manipulated variable. It is not applicable for systems that have more than one manipulated variable and one controller variable (For example, Multiple input single output (MISO) system). For such systems, a split range control strategy has wide scope in the process industries (P. Gupta *et al.*, 2015; Krishnamoorthy, 2020; Reyes-Lúa & Skogestad, 2020; Yewale *et al.*, 2020).

Split range control: Split range control scheme is used when a single controller is employed to control two or more valves (final-control elements). In such systems, the controller uses two or more manipulated variables to maintain one controlled variable at the set-point. The split range control scheme becomes unique as compared to the above-mentioned control schemes because two or more manipulated variables (MV) cover the total steady-state range of the controlled variable (Reyes-Lúa *et al.*, 2018).

Fig. 1.8 shows the block diagram of a split range control scheme with two inputs (u_1 and u_2) for one output (y). The output is controlled by inputs, u_1 and u_2 (Reyes-Lúa & Skogestad, 2019).

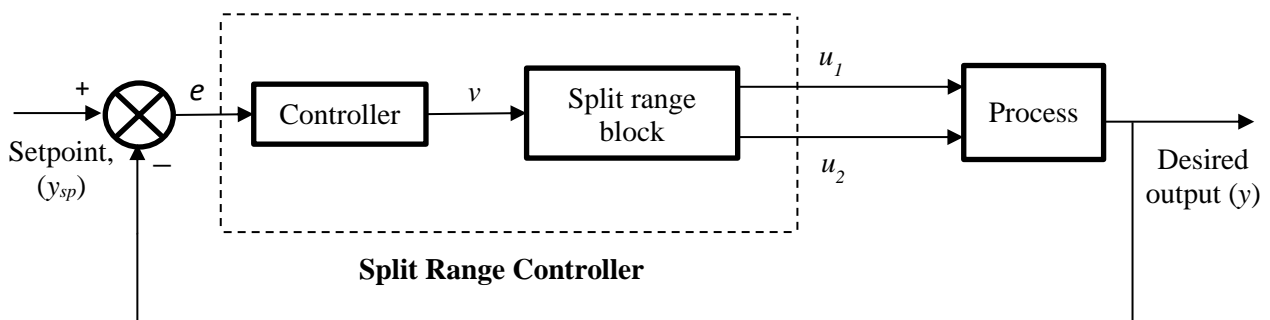


Fig. 1.8 Classical split range control for the case with two inputs and one output.

Usually, the classical split range scheme is used in which the controller output range is divided into two ranges, from 0 to 50% and 50 to 100% (Reyes-Lúa & Skogestad, 2020). Using this configuration, the controller may take effective action to control the output variable by manipulating the two input variables. To further improve system performance, **variable range of split range configuration** of controller output can be used.

The conventional tuning approaches have few drawbacks. In most cases, they do not provide effective tuning of controller gain parameters, resulting in oscillations and a significant overshoot (Bingul & Karahan, 2018; Chidambaram & Saxena, 2018; Solihin *et al.*, 2011). Hence, to overcome these drawbacks and to ensure system stability, the performance of the controller can be optimized through **nature-inspired optimization techniques**.

The conventional tuning methods are mostly considered as **offline tuning approach**, which is used for tuning the gain parameters of the controller. In offline tuning, parameters are obtained before running the process, and these parameters remain fixed during the run of the process. To further enhance the controller performance, many researchers are not only using the offline tuning approach but also updating the controller gain parameters continuously **online** using nature-inspired optimization algorithms (Davanipour *et al.*, 2018; El-Gendy *et al.*, 2020; X. Zhou *et al.*, 2019).

1.1.4 Nature-inspired optimization techniques

Nature-inspired optimization techniques are a set of problem-solving approaches which are derived from natural processes. In recent years, nature-inspired optimization algorithms have shown their efficacy in many research areas (Cui, Li, *et al.*, 2019; Cui, Zhang, *et al.*, 2019; Mohanty, 2019; Ning *et al.*, 2018; Cortés *et al.*, 2018; Xiong *et al.*, 2018; Pathak & Singh, 2017; Guha *et al.*, 2016; Odili *et al.*, 2017; Sharma & Saikia, 2015; Wang *et al.*, 2014; Bansal *et al.*, 2014; Gandomi & Alavi, 2012). These optimization techniques provide adaptive computational approaches for complex problems and a wide range of engineering applications.

Nature-inspired techniques are subclass of optimization techniques. Optimization has a broad range of applications in engineering as it offers a very powerful methodology for modelling

and problem-solving (Ali *et al.*, 2015, Kumar *et al.*, 2014, Yalsavar *et al.*, 2022, Zarachoff *et al.*, 2018). The methods available for obtaining the solution of optimization problems can be broadly categorized as **deterministic and stochastic methods**, as shown in Fig. 1.9. The **deterministic methods** follow the same path in repeated runs to provide the same solution, which is the optimal solution but is rigorous, time-consuming and quite often needs the gradient information about the objective function. On the other hand, the **stochastic or non-deterministic methods** provide different solutions in different runs because of the inherent randomness in the algorithm. The advantage of these methods is that they can provide near-optimal solutions to complex problems in a reasonable time as they explore different regions of solution space simultaneously.

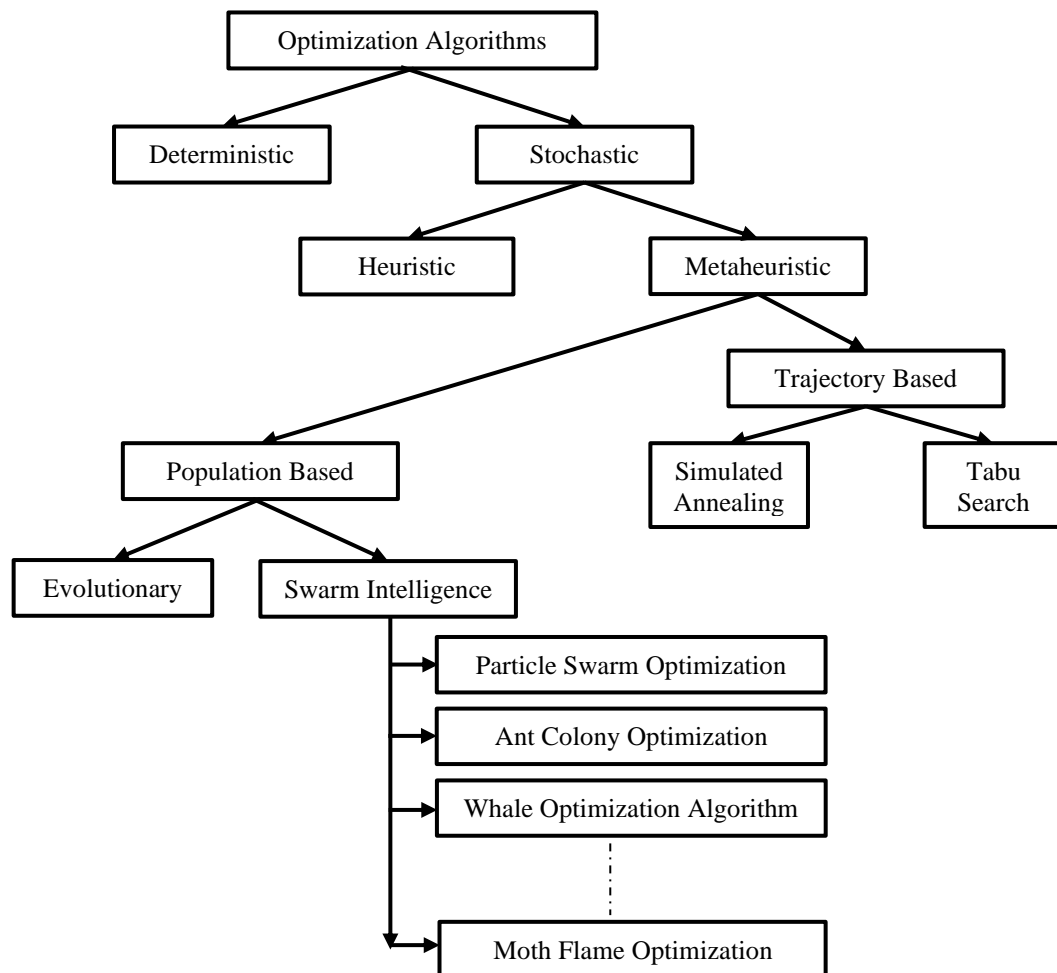


Fig. 1.9 Classification of optimization algorithms.

There are two types of derivative-free stochastic optimization algorithms: heuristic algorithms and metaheuristic algorithms (Nikolic, 2015; Ramachandran *et al.*, 2019; Siddique & Adeli, 2015). **Heuristic algorithms** find a solution by trial and error. The general problem with heuristic algorithms is that the obtained solution is not guaranteed to be the optimal one.

The second derivative-free stochastic algorithms are **metaheuristic algorithms** that can be used to solve more complex problems and very often provides better solutions than heuristic algorithms. Metaheuristic algorithms are based on principles of exploration and exploitation (X.-S. Yang, 2012). Exploration seeks to explore the full search space in order to find the different solutions that are yet to be refined. Exploitation focuses on local region search by exploiting information that the region comprises an optimal solution. These algorithms have the ability not to get stuck in local minima. The meta-heuristic algorithms can be classified as population-based and neighborhood or trajectory-based, as seen in Fig. 1.9.

Neighborhood or Trajectory-based algorithms such as simulated annealing (Kirkpatrick *et al.*, 1983) and tabu search (Glover, 1989) evaluate only one potential solution at a time, and the solution moves across a trajectory to the next probable space and so on, creating a path towards the solution. In **population-based algorithms**, a set of random solutions move towards goals simultaneously. These algorithms work iteratively to identify a set of high-performing solutions among them.

Population-based algorithms are further classified into evolutionary and swarm-intelligence-based algorithms. **Evolutionary algorithms** are fully based on Darwin's concept of evolution and survival of the fittest. Adapting this method guarantees that superior species (or solutions in terms of algorithms) preserve their functional advantage over inferior ones. Various authors have used evolutionary algorithms (such as genetic algorithm, differential evaluation, etc.) in many fields (Alibakhshikenari *et al.*, 2020; Dahunsi *et al.*, 2020; Dangor *et al.*, 2014; Dey *et al.*, 2019; Sachdeva *et al.*, 2011; Singla & Arora, 2012, Garg *et al.*, 2021). **Swarm intelligence** is population-based algorithms or nature-inspired optimization algorithms that are the most popular and widely used in the area of engineering (Anaraki *et al.*, 2018; Aruna & Jaya Christa, 2020; Cairone *et al.*, 2018; Cedro & Wiczorkowski, 2019; Gupta *et al.*, 2021; Hermawanto *et al.*, 2020; Jha & Kumar, 2020; Kapri *et al.*, 2020; Karroum *et al.*, 2020;

Laddimath & Patil, 2019; Pedro *et al.*, 2018; Zeng *et al.*, 2020). The main reason for the popularity of swarm intelligence is that these algorithms often share information among various agents, allowing for self-organization and learning during the process to aid in high efficiency (T. K. Gupta & Raza, 2019). Hence, the swarm intelligence algorithms offer a good choice for optimizing the gains of the controller.

Swarm intelligence is developed on the basis of collective behavior in self-organized systems (Gaing, 2004). These types of systems usually comprise a population of basic agents that interact locally with each other and with their surroundings. Generally, there is no centralized control structure that dictates how each agent should act, while local interactions among these agents frequently lead to the formation of global behavior. It may be seen in nature, for example, animal herding, birds flocking, fish schooling, ant colonies, and bee swarming.

Different types of swarm intelligence-based algorithms have been proposed in the literature such as Ant colony optimization (Dorigo, 1992), Particle swarm optimization algorithm (Kennedy & Eberhart, 1995), Bacterial foraging (Passino, 2002), Fish swarm/school (Li *et al.*, 2002), Bee colony optimization (Teodorovic & Dell' Orco, 2005), Bees swarm optimization (Drias *et al.*, 2005), Cat swarm optimization (Chu *et al.*, 2006), Virtual ant algorithm (X.-S. Yang *et al.*, 2006), Good lattice swarm optimization (Su *et al.*, 2007), Artificial bee colony (Karaboga & Basturk, 2007), Accelerated PSO (X.-S. Yang, 2008), Cuckoo search (X.-S. Yang & Suash Deb, 2009), Bat algorithm (X.-S. Yang, 2010), Firefly algorithm (Ali & Ahn, 2015; X. S. Yang, 2010), Eagle strategy (X. S. Yang & Deb, 2010), Krill Herd (Gandomi & Alavi, 2012), Wolf search (Tang *et al.*, 2012), Spider monkey optimization (Bansal *et al.*, 2014), Moth flame optimization algorithm (Mirjalili, 2015), Whale optimization algorithm (Mirjalili & Lewis, 2016). In the last few years, nature-inspired optimization algorithms have attracted great attention from researchers in many fields (Cortés *et al.*, 2018; Cui, Li, *et al.*, 2019; Cui, Zhang, *et al.*, 2019; Maurya, *et al.*, 2017; Ning *et al.*, 2018; Pathak & Singh, 2017; Wang *et al.*, 2014; Xiong *et al.*, 2018). At present, a variety of these swarm intelligence optimization algorithms have been successively applied, for example, Particle Swarm Optimization (PSO) (Ahn *et al.*, 2012; Amini *et al.*, 2013; Cortés *et al.*, 2018; Figueiredo *et al.*, 2016; Goud & Swarnkar, 2019; Poddar *et al.*, 2017; Sharma & Chhabra, 2021; Sohane & Agarwal, 2022; Wu *et al.*, 2014), Cuckoo Search (CS) (Baset *et al.*, 2018; Bhardwaj &

Agarwal, 2022; Cui *et al.*, 2017; Jain *et al.*, 2017; Zhang *et al.*, 2018), Whale Optimization Algorithm (WOA) (Mosaad *et al.*, 2019), Moth Flame Optimization (MFO) (Acharyulu *et al.*, 2020), Ant Colony Optimization (ACO) (Ning *et al.*, 2018; Dorigo *et al.*, 2006; Forcael *et al.*, 2014), Artificial Bee Colony (ABC) (Cui, Zhang, *et al.*, 2019; Goud & Swarnkar, 2019; Wang *et al.*, 2014), Bat Algorithm (BA) (Cai *et al.*, 2016; Cui, Li, *et al.*, 2019), Bacteria Foraging Optimization Algorithm (BFOA) (Yang *et al.*, 2016), Crow Search Algorithm (CSA) (Ranjan & Chhabra, 2022), Glowworm Swarm Optimization (GSO) (Yu & Feng, 2018; Zshou & Dong, 2018), Krill Herd Algorithm (KHA) (Guha *et al.*, 2016), and Grey Wolf Optimization (GWO) (Sharma & Saikia, 2015), etc.

From the above discussion, keeping in view the importance of temperature control (Balaton *et al.*, 2013; Greeshma *et al.*, 2019; Mahmood *et al.*, 2018; Waghmare *et al.*, 2005) and the wide scope offered by the conventional controller (Dahunsi *et al.*, 2020, Dangor *et al.*, 2014; Goud & Swarnkar, 2019), the work undertaken here envisages investigating the performance of the temperature control of the MISO system using a PID controller in a split range control scheme. An investigation of the MISO system was considered for the proposed work because most of the practical systems comprise of numerous MISO systems. Further, the literature review on temperature control of MISO system, PID controller in the split range control scheme, controller tuning using the Z-N method, or nature-inspired optimization techniques has been reported in the next section.

1.2 Literature review

The purpose of this section is to present a review of various studies conducted in the field of temperature control using the split range controller and also, to examine the performance of the controller. The brief details about these studies are as follows:

Table 1.3 Literature survey on the temperature control, split range PID controller, and controller tuning methods.

Research group	Key points	Remarks
<p><i>Reyes-Lúa and Skogestad</i> (Reyes-Lúa & Skogestad, 2020)</p>	<ul style="list-style-type: none"> ➤ Proposed a generalized split range control structure using a baton strategy. ➤ This new strategy provided better performance by reducing approximately 54.88% in the integrated absolute error value and 18.62% to 22.90% in the value of settling time, as compared to the conventional split range controller. 	<ul style="list-style-type: none"> ➤ The authors used a generalized split range control for a particular temperature set-point and defined a definite range of the controller output (not automatically) for varying the manipulated variables. If the temperature set-point changes, then the system's performance may degrade in terms of settling time. ➤ The authors used the same type of manipulated variables (either cooling or heating) to reduce the error in one direction. Therefore, it can be said that they restricted the range of error correction.
<p><i>Reyes-Lúa and Skogestad</i> (Reyes-Lúa & Skogestad, 2019)</p>	<ul style="list-style-type: none"> ➤ The authors used mainly two structures (one is PI controller in a normal split range scheme, and the second used three conventional PI controllers) to control the room temperature. ➤ Compared the controller performance and energy performance for both the structures. It was found to a reduction of 7.66% in the energy cost, as compared to split range control, and also showed comparable performance in terms of settling time. 	<ul style="list-style-type: none"> ➤ The authors used a standard split range control strategy (divided controller output range into two ranges). If the temperature set-point changes, the performance of the system may be poor in terms of settling time.

<p><i>Soto and Hernandez-Riveros</i> (Soto & Hernandez-Riveros, 2019)</p>	<ul style="list-style-type: none"> ➤ Presented Multidynamics algorithm for global optimization (MAGO) algorithm (Evolutionary algorithm) for tuning the split range controller parameters to control the temperature in Vapour compression refrigeration system (VCRS). ➤ Enhanced the behavior of temperature (controlled variable) by reducing the settling time (approx. 2.38%), as compared to the PID controller, thereby improving the energy performance of the system. 	<ul style="list-style-type: none"> ➤ To improve the system performance, a variable range of split range control scheme for bifurcating the controller output range can be a different area of research.
<p><i>Reyes-Lúa et al.</i> (Reyes-Lúa et al., 2018)</p>	<ul style="list-style-type: none"> ➤ The performance of controllers (conventional split range control and valve position control) is tested for disturbance rejection in cooling water temperature of $+2^{\circ}\text{C}$ at $t=200\text{ sec.}$ and $+4^{\circ}\text{C}$ at $t=2000\text{ sec.}$ ➤ The temperature gets earlier settled (by 19.11% approx.) in the case of split range control, as compared to valve position control. 	
<p><i>Mahitthimahawong et al.</i> (Mahitthimahawong et al., 2016)</p>	<ul style="list-style-type: none"> ➤ Proposed the application of split range control for heat exchanger networks. ➤ The performance of split range control was found better as compared to conventional PI controller based on stability analysis and utility cost. ➤ The responses of outlet cold and hot temperature of streams with PI controller are settled late (by 1.59% and 2.69% approx., respectively) with more oscillations, as compared to split range control. 	<ul style="list-style-type: none"> ➤ Nature-inspired controller techniques can be investigated for better performance by optimizing the split range controller parameters.

<p><i>Arora and Gupta</i> (Arora & Gupta, 2013)</p>	<ul style="list-style-type: none"> ➤ Used split range controller and the conventional PID controller, to control the temperature of the reactor. ➤ Compared these two controller performances and found better results in the case of split range controller, reducing approximately 39.39% in settling time, and performance error criteria (ITAE, IAE, ISE, ITSE). 	
<p><i>Balaton et al.</i> (Balaton et al., 2013)</p>	<ul style="list-style-type: none"> ➤ Performed split range controller for handling the three temperature ranges in the case of a monofluid thermoblock, especially the medium temperature range with low energy consumption. ➤ The parameters of the controller were determined by a genetic algorithm. ➤ The simulation results using the split range control strategy showed that temperature reached the desired set-point with a small overshoot and low settling time (approx. 4.18%), and no oscillation, as compared to the PI controller. 	<ul style="list-style-type: none"> ➤ Authors used the offline tuning approaches for optimizing the parameters of the conventional controller. So, for the better performance, the controller gain parameters can be updated continuously online.
<p><i>Zhang et al.</i> (X. J. Zhang et al., 2012)</p>	<ul style="list-style-type: none"> ➤ Developed a new temperature and humidity independent control device and tested split range controller performance of the device in a constant temperature and humidity air conditioning system working in a storeroom in a museum. ➤ Both the temperature and the humidity were controlled in the required ranges. As a result, the temperature varied in the range of 21.9°C and 22.1°C, and humidity varied between 59.3% and 61.1%. ➤ The performance of energy of the device 	

	<p>was also tested using simulation tests on the TRNSYS platform, and compared with the conventional air conditioning system. The results demonstrated that the developed control system obtained energy savings of 30-50%.</p>	
<p><i>Králová and Doležel</i> (Králová & Doležel, 2009)</p>	<ul style="list-style-type: none"> ➤ Used two PID controllers and a split range PID controller to control the temperature of the thermostatic bath. The controller gain parameters are determined using the trial-error method. ➤ Compared the results of both in terms of settling time and cost. It was found comparable performance in terms of settling time and a reduction of 47.31% in total cost (in the case of split range PID controller). 	<ul style="list-style-type: none"> ➤ The nature-inspired optimization techniques using an online tuning approach for optimizing the controller parameters can be further area of research.

1.3 Research gaps

Based on the literature mentioned above, the scope for future investigations have been identified and listed as below:

- The authors have used the standard split range control strategy for dividing the controller output range into two ranges that affect the system performance. Therefore, for better performance, a variable range of split range control scheme for bifurcating the controller output range can be a further area of research.
- Nature-inspired controller techniques can be investigated to optimize the controller gain parameters of split range control which may provide better performance.
- Many authors used the offline tuning approaches for optimizing the parameters of the conventional controller. Hence, an online tuning approach for optimizing the controller

parameters can be presented. To further improve the performance of the system, the controller parameters can be tuned online using nature-inspired optimization techniques that can be a further area of research.

- Validation of the results in a real environment can be the area of future research.

1.4 Objectives

The research gaps established above provide the motivation for the work undertaken here. The objectives of the work are listed as follows:

1. Investigation of MISO system for temperature control.
2. Implementation and Comparison of different nature inspired controller techniques for temperature control in MISO system.
3. Validation of the above investigations in real time environment.

1.5 Dissertation outline

The studies conducted and the results obtained are presented as chapters of this dissertation, the outline of which is provided as follows:

Table 1.4 Dissertation outline based on chapters.

S. No.	Description	Remarks
Chapter 1	Introduction	This chapter covers a background of Process Control, details about the process variables, control schemes, and tuning approaches being explored are discussed. To attain improved system performance, various studies on the process, controllers, tuning methods, and nature-inspired optimization techniques have been reported in the past. The survey of the literature is structured in chronological order (year-wise). The chapter also reports a list of observed research gaps which lay the foundation for the objectives of the dissertation and provides details about the outline of the dissertation.

Chapter 2	Modelling and Control Strategies	This chapter reports system description and modelling in detail. It presents the PID controller in a conventional scheme, a standard split range scheme, and a variable range split range scheme. This chapter also includes the description of the variable range of split range-based PID controller for the temperature control of the MISO (multiple input single output) water tank system (blending process) in detail.
Chapter 3	Controller: Tuning and Performance Evaluation	This chapter presents the use of nature-inspired algorithms for tuning the PID controller used in the variable range split range control scheme for temperature control of a mixing process. A comparison of the performances of the controllers tuned using these algorithms and the conventional Z-N method is also presented. Further, an improved nature-inspired algorithm is proposed to enhance the performance of the controller.
Chapter 4	Performance Evaluation of Online Tuned Variable Range SR-PID Controller	This chapter presents an online tuning approach using the original and improved versions of the MFO algorithms for optimizing the parameters of variable range split range PID controller to control the temperature of the mixing process. The performances of the controllers are investigated for the various temperature setpoints in terms of settling time and compared with performances obtained using the offline tuning approach with the MFO algorithm. Further, the performance of the online tuned controller using the proposed algorithm (EMFO) is investigated with respect to the effect of system dynamics and the effect of process disturbance.
Chapter 5	Validation	This chapter investigates the performances of the online tuned controllers in the simulated real environment. The effect of using the EMFO algorithm for online tuning of the controller in the simulated real environment is studied. An electrical analogous model of the practical environment is simulated for investigation by considering several effects (imperfect insulation, density, viscosity, and compressibility) found in real-time conditions. Further, the system is also investigated with the effect of system dynamics, and process disturbance. Moreover, a comparison of the performances of variable range SR-PID controller tuned online

using EMFO algorithm in case of the ideal environment and the practical environment is also studied.

Chapter 6	Conclusion and Future Scope	This chapter summarizes the research outcomes and the significant contributions of this dissertation. It also provides the future scope for improvement of the current research work.
------------------	------------------------------------	---

Modelling and Control Strategies

This chapter reports system description and modelling in detail. It presents the PID controller in a conventional scheme, a standard split range scheme and a variable range of split range scheme. This chapter also includes the description of variable range of split range-based PID controller for the temperature control of the MISO (multiple input single output) water tank system (blending process) in detail.

In Chapter 1, the brief introduction of process control, details about its variables, schemes, tuning approaches, and optimization techniques were discussed. This chapter describes the system, its modelling, and control methods in depth. It also investigates the PID controller's performance in conventional scheme, a standard split range scheme and the variable range of split range scheme, to control the temperature of the mixing process.

2.1 System description and modelling

The present work considered a mixing process as most industrial processes consist of the mixing process in process control industries. There are two inputs, namely, cold water (flow rate Q_1 , temperature T_1) and hot water (flow rate Q_2 , temperature T_2). Both the constituents are mixed thoroughly, and water is discharged from the outlet (flow rate Q , temperature T). The schematic diagram of the mixing process of two distinct flows in the water tank system is depicted in Fig. 2.1.

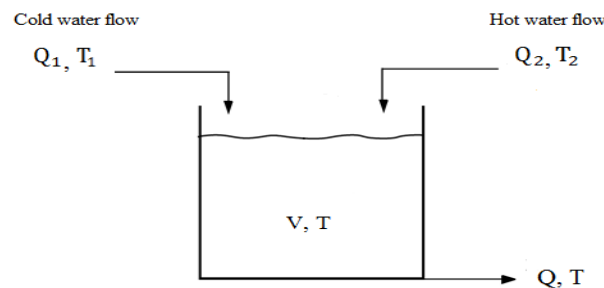


Fig. 2.1 Mixing process of two different water flows in water tank system.

Mathematical modelling of the process is done using the mass balance equation and the energy balance equation (Nagy, 2007). The assumptions for the development of the model are taken as follows:

- ❖ The volume of water in the tank is constant
- ❖ Input flow rate is equal to output flow rate, i.e., $Q = Q_1 + Q_2$
- ❖ The density and heat capacity of water are constant
- ❖ Perfect mixing
- ❖ Perfect insulation

On the basis of mass balance equation and energy balance equation, the differential equation for the development of the model can be written as follows:

$$V \frac{d(T)}{dt} = Q_1 T_1 + Q_2 T_2 - QT \quad (2.1)$$

The general solution of Eq. 2.1 is

$$T = T_\infty * \left(1 - e^{-\frac{t}{\tau}}\right) + T_0 * e^{-\frac{t}{\tau}} \quad (2.2)$$

where τ is the time constant of the water tank, T_0 is the temperature of water at the starting of measurement and T_∞ is the temperature after the measurement of transient characteristics, and can be calculated as follows:

$$\tau = \frac{V}{Q_1 + Q_2} \quad (2.3)$$

$$T_0 = \frac{Q_1 T_1 + Q_2 T_2}{Q_1 + Q_2}, \quad \text{when } t = 0, \quad (2.4)$$

$$T_\infty = \frac{Q_1 T_1 + Q_2 T_2}{Q_1 + Q_2}, \quad \text{when } t > 0, \quad (2.5)$$

In this work, the temperature of water in the tank is controlled by manipulating the flow rates of the cold and hot water flowing into the tank. Hence the temperature of water in the tank (T) is the controlled variable and the flow rates of the cold and hot water (Q_1 and Q_2 , respectively) are the manipulated variables. The temperature of the inflowing cold and hot water is taken as 20°C and 35°C, respectively. The volume of the water in the tank is maintained at 30 litres. Considering

pipes of 0.5 inches diameter for inflowing water, the range of flow rates resulting into a laminar (steady) flow through the pipes was determined by inspection in the laboratory. The minimum and the maximum flow rates thus obtained were 0.0115 litre/sec. and 0.1 litre/sec., respectively. From this range of flow rates obtained, flow rates corresponding to 30% (0.015 litre/sec.) and 75% (0.075 litre/sec.) of the stem position of the valve were taken as the lower and upper saturation limits of the working range of inflows in this work. The minimum and maximum temperature that can be obtained with the working range of flows is 22.5°C and 32.5°C, respectively. Hence, the working range of the temperature in this work is 22.5°C to 32.5°C.

In most of the practical cases, the performance of the closed-loop control system is specified in terms of transient response as it often exhibits damped oscillations before reaching the steady state. To study this transient performance, in this work, the performance of the controller is tested for the different temperature setpoints in terms of steady-state error (E_{ss}) and settling time (T_s). Some of the factors that affect the performance of the closed-loop system (Seborg *et al.*, 2008) are described below:

System dynamics

System dynamics is an approach to understand the nonlinear behaviour of the system over time. In the closed-loop system, nonlinearities can exist in the sensor, controller, process, and valve. However, nonlinearity in the valve is most important due to manipulation in manipulated variable. Hence, keeping in view the importance of changing the manipulated variable, it was decided to study the system dynamics through this factor (nonlinearity in the valve). The nonlinearity in the valve is generally considered as dead time. Thus, to study the effect of dead time, initially, the performance of the system is checked using the controller with the valve with no dead time associated. Further, it is investigated by considering the dead time of 0.25 and 0.5 seconds. As higher dead time leads to the reduction of ultimate gain of the system and makes it more prone to even low-frequency disturbances, dead time is taken in the lower half of the range, i.e., 0.25–0.5 seconds.

Disturbance rejection

In actual operation of the system along with closed-loop control, disturbances exist naturally as effects of the environment. So, the modelling and simulation of the effects of disturbances are necessary during design because the goal of the control system is to drive the physical system to follow reference commands in the presence of disturbances. In this work, a disturbance is introduced only in the forward path in the process, not in the feedback path as no sensors are considered, and the controller performance is investigated with the effect of the process disturbance to maintain the desired value by changing the manipulated variables (considered as flow rates). The disturbances in the process can exist as the environmental effect, random noises present, practical error (due to the machinist's fault, effect of loading), abnormal behaviour of the actuator (manipulation of the flow rate), etc. Since the majority of process disturbances are typically represented by impulse function, hence, disturbance as impulse function is introduced in the process to examine the controller's disturbance rejection capability (Luo & Lee, 1999). Therefore, in this work, the impulse function is used to examine the effect of disturbance on the system.

In this work, these above factors are used to investigate the closed-loop system performance. The performance of the controller is also tested in terms of utility consumption.

Utility consumption

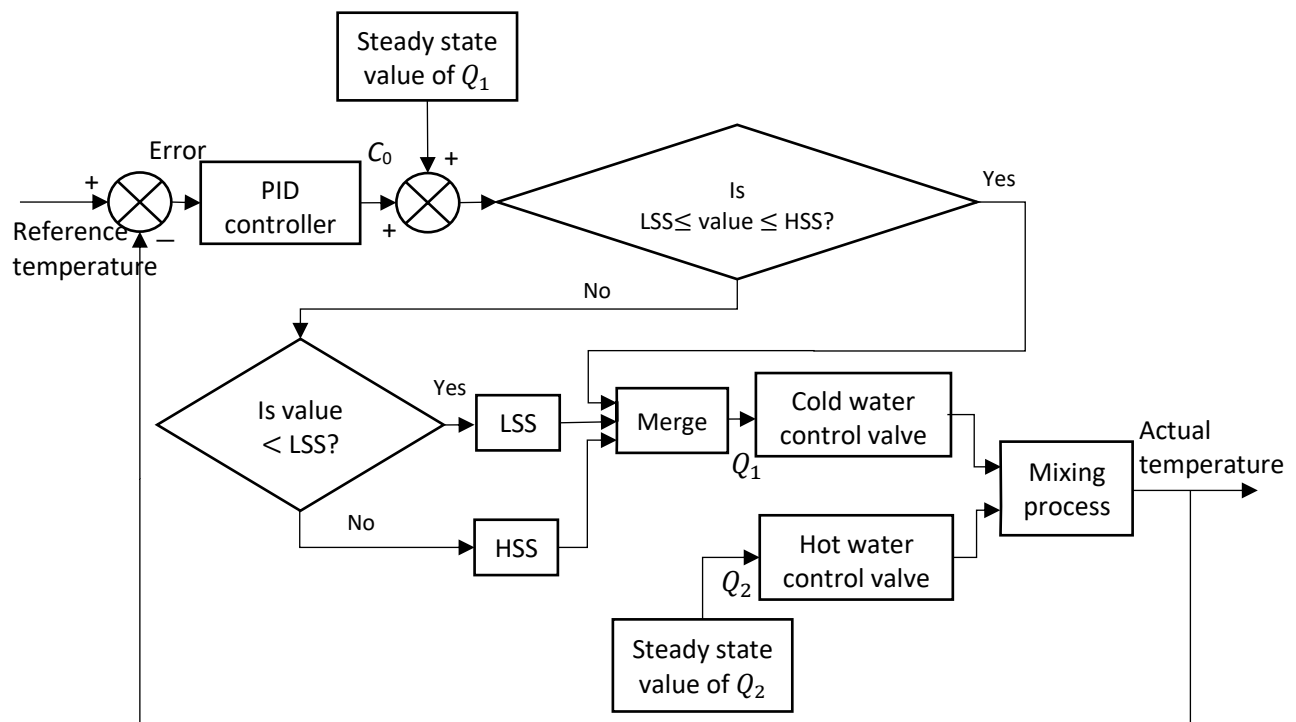
The utility can be understood easily in terms of economics as it directly impacts the demand and consequently, cost. In this work, cold water and hot water are considered as utilities. To study the consumption of utility (Fonseca *et al.*, 2013), the controller performance is checked for the various temperature setpoints and amount of the utilities (cold and hot water) are noted.

2.2 Control strategies

In this work, PID controller is used in different schemes to control the temperature of the mixing process, as described below.

2.2.1 PID controller in the conventional scheme

The conventional PID controller is used for both SISO and MISO systems. In SISO system, a single PID controller is used (Khare & Singh, 2010; Mahmood *et al.*, 2018), while in case of MISO system, multiple PID controllers (one for each manipulated variable) are used for controlling the controlled variable of the process (Králová & Doležel, 2009; Reyes-Lúa & Skogestad, 2019). The use of PID controller for each manipulated variable requires the setpoint for each manipulated variable to be known beforehand. In this work, since various combinations of flow rates are possible for a temperature, the setpoints for the manipulated variables will also vary accordingly for a given rise/fall in temperature. This makes the task of finding setpoints for each manipulated variable very difficult. Therefore, only one PID controller was used for controlling the temperature of the mixing process. In this strategy, only one manipulated variable is used to control the controlled variable while the other one is constant at its steady-state value. The schematic diagram of controlling the temperature of the mixing process using a PID controller is shown in Fig. 2.2.



(a)

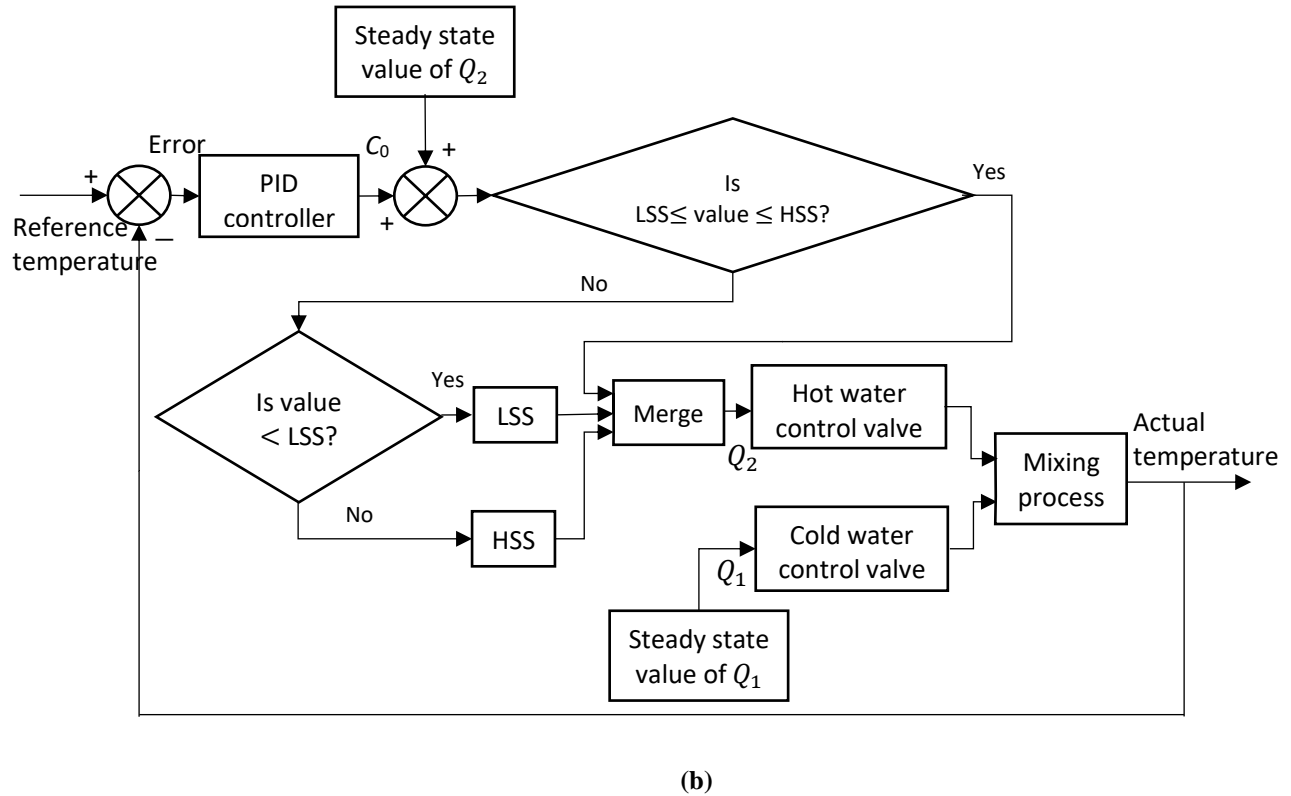


Fig. 2.2 Schematic diagram of temperature control using the conventional PID controller (a) for increasing temperature setpoint, (b) for decreasing temperature setpoint.

The use of only one PID controller allows the system to manipulate the flow rate of only one inflow instead of both the inflows. As a result, the settling time increases and even some steady state error is also observed.

In this work, the performance of the PID controller in different control schemes for temperature control of a mixing process is investigated. A desired increase/decrease of 10% and 30% of the working range (22.5°C to 32.5°C) of temperature is considered in different portions of the working range. For this, the five initial setpoints (extremities, midpoint, one each in the lower and upper half) considered are 22.5°C, 25°C, 27.5°C, 29°C, 32.5°C. The performances of the controllers are compared for 1°C and 3°C rise/ fall from these setpoints within the working range.

Each temperature setpoint has a number of possible combinations of the flow rates of the two inflows. For the best performance, the PID controller's gain parameters should be different for each combination of flow rates. However, it is very difficult to tune the PID controller for all

combinations of flow rates for a particular temperature. Therefore, PID controller is tuned for one combination of flow rates for a particular temperature. Later on, the performance of the PID controller tuned at one combination of flow rates is investigated for other combinations of flow rates at the same temperature. In the current work, a combination of steady-state flow rates (Q_{10} Q_{20}) obtained using Eq. 2.4 are (0.075 0.015), (0.04 0.02), (0.045 0.045), (0.02 0.03), (0.015 0.075) for temperature setpoints 22.5°C, 25°C, 27.5°C, 29°C, 32.5°C, respectively.

In this work, the tuning is done by the classical Ziegler-Nichols (Z-N) tuning method, as discussed in section 1.1.3.1. The controller gain parameters were tuned for all the cases (mentioned in Table 2.1) within the working range of temperature, and the gain parameters for all the cases were obtained, as shown in Table 2.1.

Table 2.1 Controller gains of PID controller tuned with Z-N method for all the cases under consideration.

Cases	Setpoints (°C)	Error	Controller gains		
			K_p	K_i	K_d
1	22.5	+1	-46.65	-0.09	-5.75
2	22.5	+3	-48.18	-0.08	-6.63
3	25	+1	-33.58	-0.002	-5.78
4	25	+3	-36.87	-0.008	-8.51
5	25	-1	32.75	0.029	3.44
6	25	-2	50.34	0.044	5.09
7	27.5	+1	-28.39	-0.051	-6.28
8	27.5	+3	-34.94	-0.025	-3.12
9	27.5	-1	28.19	0.031	5.26
10	27.5	-3	34.45	0.069	3.86
11	29	+1	-24.26	-0.009	-7.18
12	29	+3	-46.75	-0.017	-6.18
13	29	-1	29.49	0.009	5.73
14	29	-3	28.63	0.018	2.74
15	32.5	-1	28.88	0.076	4.97
16	32.5	-3	30.57	0.064	6.77

For the increasing temperature setpoints, corresponding to each set of parameters (K_p , K_i , and K_d) obtained, the performance of the controller for all the cases within the entire range was checked. The best set of parameters was selected on the basis of the algebraic sum to be minimum, as shown in bold in Table 2.2. The same procedure was adopted for the calculation of optimal gains for the decreasing temperature setpoints, as shown in bold in Table 2.3.

Table 2.2 Performance of PID controller corresponding to each set of parameters (for increasing setpoints).

Setpoints (°C)	Error	Case 1	Case 2	Case 3	Case 4	Case 7	Case 8	Case 11	Case 12
22.5	+3	789	617	756	767	842	673	785	1168
22.5	+1	211	388	211	322	257	242	357	283
25	+3	1161	1306	987	737	1757	1606	1018	2333
25	+1	346	420	279	344	393	357	363	421
27.5	+3	2142	3582	1673	2124	3353	1229	2240	3845
27.5	+1	567	687	439	568	191	576	1032	832
29	+3	7075	6014	5285	7470	9060	7917	7004	3255
29	+1	1209	1394	784	798	999	880	294	1480
Total Sum		13500	14408	10414	13130	16852	13480	13093	13617

Table 2.3 Performance of PID controller corresponding to each set of parameters (for decreasing setpoints).

Setpoints (°C)	Error	Case 5	Case 6	Case 9	Case 10	Case 13	Case 14	Case 15	Case 16
25	-1	922	1267	2063	1402	1449	1297	1571	1266
25	-2	6471	5989	13406	7734	8909	12911	8760	9243
27.5	-1	554	941	125	585	407	791	247	526
27.5	-3	2271	3949	1747	774	3527	3609	3832	1098
29	-1	666	1168	1012	492	304	989	361	382
29	-3	1990	2619	2391	2215	1469	923	1334	1823
32.5	-1	298	349	429	322	223	263	234	287
32.5	-3	744	781	724	722	708	963	801	521
Total Sum		13916	17063	21897	14246	16996	21746	17140	15146

The set of the gain parameters for two cases (Case 3 and Case 5) was selected as discussed above, and the performance for all the individual setpoints on the entire range was evaluated. The best set of parameters was chosen (Case 3) on the basis of the best settling time obtained during various simulations, as shown in [Table 2.4](#). The values were observed as K_p (proportional gain) = -33.58, K_i (integral gain) = -0.002, and K_d (derivative gain) = -5.78. These gain values

were then used to obtain the simulation results for increasing temperature setpoints (with the same signs) and decreasing temperature setpoints (with opposite signs).

Table 2.4 Performance of PID controller for the various temperature setpoints using the best set of parameters (Case 3 and Case 5).

Cases	Setpoints (°C)	Error	Case 3	Case 5
1	22.5	+3	756	774
2	22.5	+1	211	296
3	25	+3	987	1044
4	25	+1	279	317
5	25	-1	1149	922
6	25	-2	6641	6471
7	27.5	+3	1673	1945
8	27.5	+1	439	573
9	27.5	-1	442	554
10	27.5	-3	1678	2271
11	29	+3	5285	7484
12	29	+1	784	805
13	29	-1	349	666
14	29	-3	1509	1990
15	32.5	-1	205	298
16	32.5	-3	752	744
Total Sum			23139	27154

However, since the scheme in which PID controller was used, has an inherent limitation as discussed earlier. The results have a scope for further improvement. This motivated to investigate the performance of the split-range control scheme for temperature control of the mixing process.

2.2.2 PID controller in split range scheme

In this work, first, the standard split range control scheme is used in which the controller output range is divided into two ranges (0 to 50% and 50 to 100%) (Balaton *et al.*, 2013). However, dividing the complete range into two equal halves does not lead to desirable results always as LSS is reached well before the lower limit of the first half range. Hence, a variable split range control scheme is considered in which the range of controller output is bifurcated, depending on the steady-state input flow rates (Q_{10} and Q_{20}), lower saturation state (LSS), and higher saturation state (HSS) (Vishnoi *et al.*, 2021).

The split range configuration for controller output (C_0) can be defined when the temperature increases/decreases from its setpoint. The steady-state values of Q_1 and Q_2 and temperature setpoint will determine how much portions of the controller output will be used for the cold water valve V_1 and hot water valve V_2 (Vishnoi *et al.*, 2021a). Schematic diagram of controlling the temperature of the mixing process by manipulating the flow rates Q_1 and Q_2 is shown in Fig. 2.3. If the temperature increases or decreases from its setpoint, the split range configuration of the PID controller output in Fig. 2.3 can be specified as shown in Fig. 2.4.

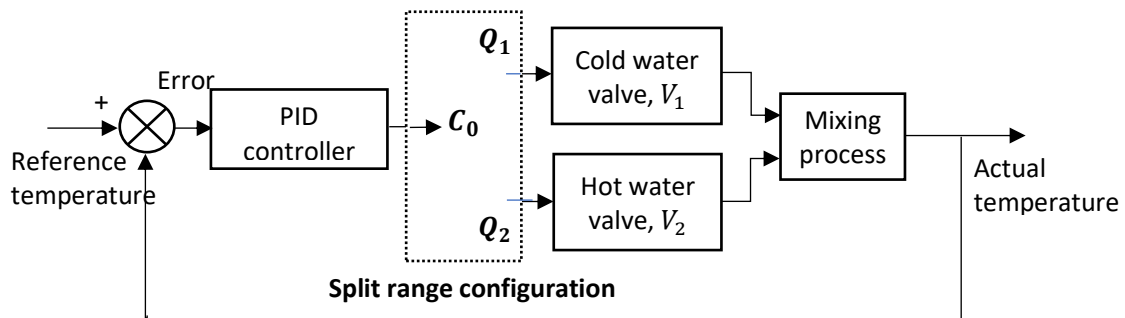
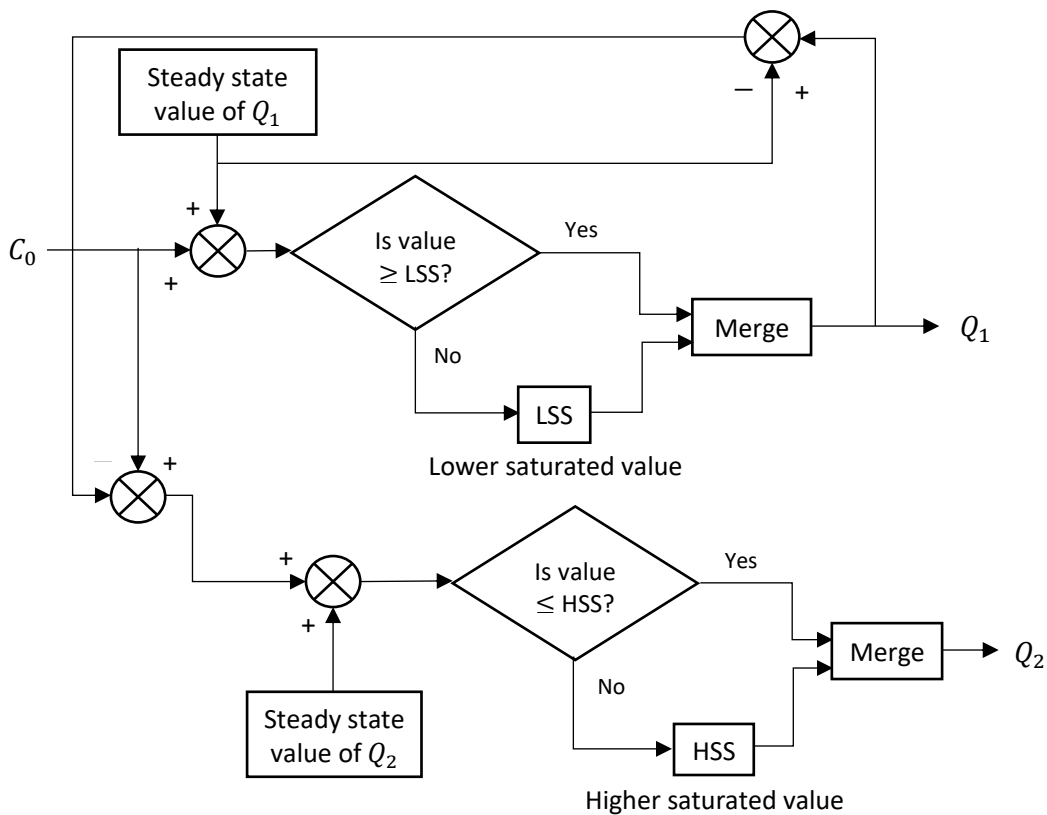


Fig. 2.3 Schematic diagram of temperature control using the variable range of split range PID controller.



(a)

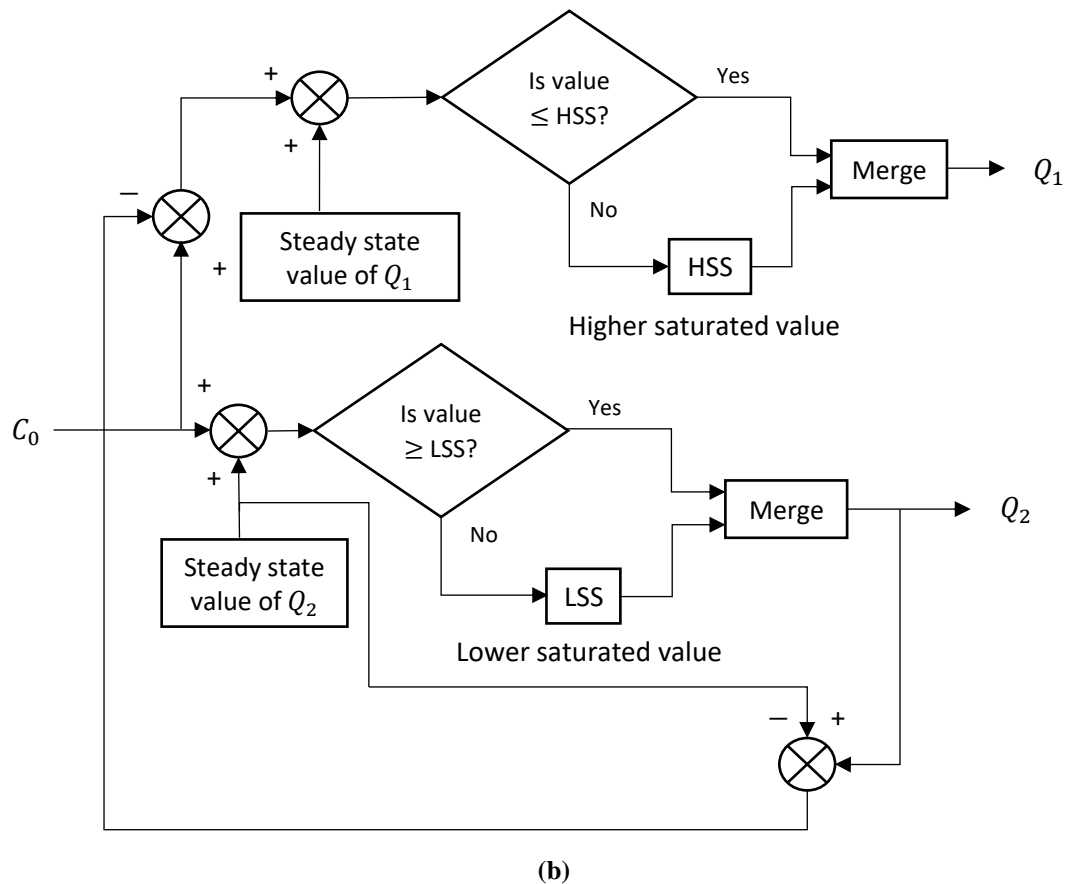


Fig. 2.4 Split range configuration of the controller output: (a) for increasing temperature setpoint, (b) for decreasing temperature setpoint.

The split range based PID controller was used to control the temperature of the mixing process by manipulating cold and hot water flow rates (Q_1 and Q_2 , respectively) with the help of an error signal. From Fig. 2.4, it can be seen that when the algebraic sum of controller output and initial steady-state flow rate of Q_1 or Q_2 is more than the lower steady-state value [$(C_0 + (Q_{10} \text{ or } Q_{20})) \geq LSS$], the value of Q_1 or Q_2 is calculated as $[(Q_{10} \text{ or } Q_{20}) + C_0]$ keeping Q_2 or Q_1 constant, respectively. However, if the algebraic sum is less than the lower steady-state value [$(C_0 + (Q_{10} \text{ or } Q_{20})) < LSS$], the controller action is bifurcated as follows: Q_1 or Q_2 approaches lower steady-state value, and the remaining part of the controller action is used for the operation of Q_2 or Q_1 , respectively, in the reverse mode (not exceeding the value of higher steady-state, HSS).

The action of the two valves taken as cold-water valve V_1 and hot water valve V_2 corresponding to Q_1 and Q_2 input streams, respectively, with respect to deviation in flow (d) is shown in Fig.

2.5, which shows the behaviour of the split range strategy. In the figure, the valve stem positions are considered as a minimum at 30% (fully close) and maximum at 75% (fully open), discussed in section 2.1.

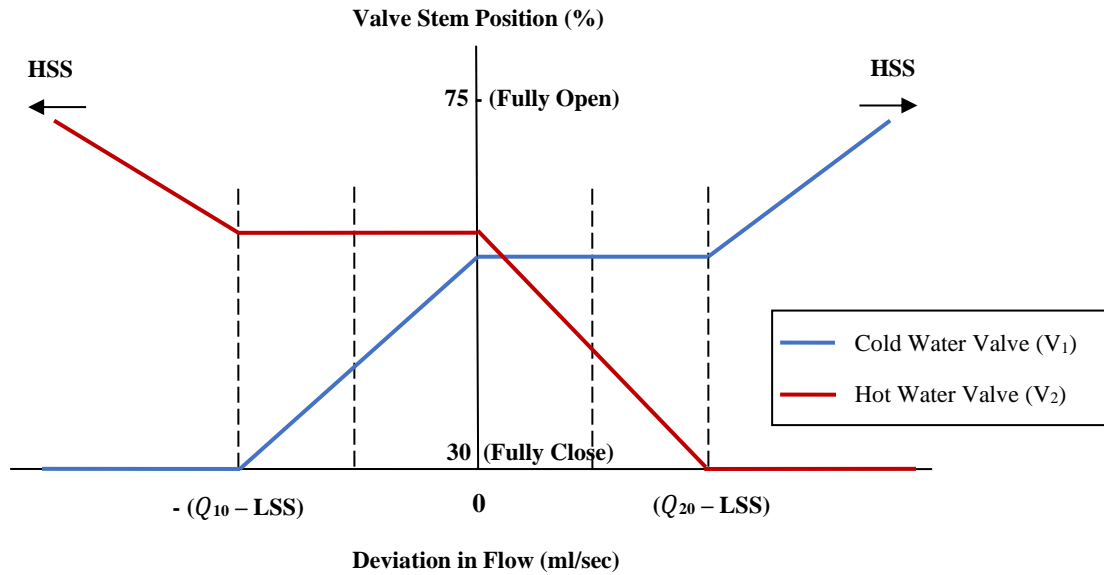


Fig. 2.5 Plot of action of two valves V_1 and V_2 .

A plot of action of two valves (in Fig. 2.5) can be understood by Tables 2.5 (a) and 2.5 (b) as follows:

Table 2.5 (a) Action of valves V_1 and V_2 , when $d \leq 0$ and $Q_{10} \neq LSS$.

Deviation in flow (d)	Action of valves	
	V_1	V_2
$d = 0$	Constant at steady state	Constant at steady state
$-(Q_{10} - LSS) < d < 0$	Decrease	Constant
$d = -(Q_{10} - LSS)$	Lower steady state	Constant
$d < -(Q_{10} - LSS)$	Lower steady state	Increase (maximum limit HSS)

Table 2.5 (b) Action of valves V_1 and V_2 , when $d \geq 0$ and $Q_{20} \neq LSS$.

Deviation in flow (d)	Action of valves	
	V_1	V_2
$d = 0$	Constant at steady state	Constant at steady state
$0 < d < (Q_{20} - LSS)$	Constant	Decrease
$d = (Q_{20} - LSS)$	Constant	Lower steady state
$d > (Q_{20} - LSS)$	Increase (maximum limit HSS)	Lower steady state

As discussed in Section 2.2.1, the same procedure was adopted for the calculation of the best set of the gain parameters in case of PID controller in a standard split range scheme (Tables 2.6–2.9) as well as the variable range of split range scheme (Tables 2.10–2.13) for the same above said setpoints.

Table 2.6 Controller gains of the standard SR-PID controller tuned with the Z-N method for all the cases under consideration.

Cases	Setpoints (°C)	Error	Controller gains		
			K_p	K_i	K_d
1	22.5	+1	-49.64	-1.25	-5.20
2	22.5	+3	-51.26	-1.80	-5.09
3	25	+1	-36.17	-0.93	-2.89
4	25	+3	-39.23	-1.25	-4.98
5	25	-1	36.42	0.83	3.80
6	25	-2	55.97	1.17	5.62
7	27.5	+1	-36.57	-0.53	-6.24
8	27.5	+3	-38.85	-1.07	-3.87
9	27.5	-1	31.35	1.79	5.23
10	27.5	-3	38.31	1.75	4.83
11	29	+1	-26.98	-1.16	-7.14
12	29	+3	-51.98	-0.35	-6.14
13	29	-1	32.79	1.07	6.14
14	29	-3	31.84	0.98	3.01
15	32.5	-1	32.11	0.91	5.46
16	32.5	-3	33.99	1.16	4.37

Table 2.7 Performance of the standard SR-PID controller corresponding to each set of parameters (for increasing setpoints).

Setpoints (°C)	Error	Case 1	Case 2	Case 3	Case 4	Case 7	Case 8	Case 11	Case 12
22.5	+3	552	460	548	537	589	474	552	817
22.5	+1	148	250	155	229	180	171	268	194
25	+3	812	712	734	523	1229	1131	901	1598
25	+1	242	254	213	244	275	252	290	289
27.5	+3	1498	1567	1365	1507	2345	866	2471	2634
27.5	+1	397	722	340	402	134	406	474	570
29	+3	4948	4898	4395	5298	6336	5576	4148	2230
29	+1	846	1115	573	566	699	620	203	1014
Total Sum		9443	9978	8323	9306	11787	9496	9307	9346

Table 2.8 Performance of the standard SR-PID controller corresponding to each set of parameters (for decreasing setpoints).

Setpoints (°C)	Error	Case 5	Case 6	Case 9	Case 10	Case 13	Case 14	Case 15	Case 16
25	-1	654	899	1495	1039	1043	940	1084	938
25	-2	5301	4128	9715	5729	6410	9157	6042	6847
27.5	-1	393	668	91	434	293	561	171	390
27.5	-3	1611	2801	1266	574	2538	2560	2643	814
29	-1	473	829	734	365	219	702	249	283
29	-3	1412	1858	1733	1641	1057	655	920	1351
32.5	-1	212	248	311	239	161	187	162	213
32.5	-3	528	554	525	535	510	683	553	386
Total Sum		10584	11985	15870	10556	12231	15445	11824	11222

Table 2.9 Performance of the standard SR-PID controller for the various temperature setpoints using the best set of parameters (Case 3 and Case 10).

Cases	Setpoints (°C)	Error	Case 3	Case 10
1	22.5	+3	548	557
2	22.5	+1	155	270
3	25	+3	734	910
4	25	+1	213	293
5	25	-1	862	1039
6	25	-2	5482	5729
7	27.5	+3	1365	2492
8	27.5	+1	340	478
9	27.5	-1	343	434
10	27.5	-3	1372	574
11	29	+3	4395	4189
12	29	+1	573	505
13	29	-1	267	365
14	29	-3	1126	1641
15	32.5	-1	151	239
16	32.5	-3	545	535
Total Sum			18471	20250

Table 2.10 Controller gains of the variable range of SR-PID controller tuned with Z-N method for all the cases under consideration.

Cases	Setpoints (°C)	Error	Controller gains		
			K_p	K_i	K_d
1	22.5	+1	-55.2	-1.4	-4.7
2	22.5	+3	-57	-0.9	-4.6
3	25	+1	-40.8	-1	-3.7
4	25	+3	-43.62	-1.4	-4.5
5	25	-1	40.5	0.8	4.2
6	25	-2	62.23	0.9	6.4
7	27.5	+1	-35.1	-0.6	-6.2
8	27.5	+3	-43.2	-1.2	-4.8
9	27.5	-1	34.86	2	5.2
10	27.5	-3	42.6	1.6	4.8
11	29	+1	-30	-1.3	-7.1
12	29	+3	-57.8	-0.4	-6.1
13	29	-1	30.9	1.2	6.1
14	29	-3	35.4	0.9	3.3
15	32.5	-1	35.7	0.9	6
16	32.5	-3	37.8	1.3	4.8

Table 2.11 Performance of the variable range of SR-PID controller corresponding to each set of parameters (for increasing setpoints).

Setpoints (°C)	Error	Case 1	Case 2	Case 3	Case 4	Case 7	Case 8	Case 11	Case 12
22.5	+3	376	322	369	392	412	334	381	560
22.5	+1	104	175	116	163	126	121	185	133
25	+3	568	498	565	371	860	797	622	1095
25	+1	171	178	166	172	193	178	200	198
27.5	+3	1124	1166	1100	1134	1850	610	1912	2010
27.5	+1	282	505	265	282	94	286	327	391
29	+3	3705	3705	3563	3793	4431	3927	2861	1528
29	+1	396	780	400	742	489	437	140	695
Total Sum		6726	7329	6544	7049	8455	6690	6628	6610

Table 2.12 Performance of the variable range of SR-PID controller corresponding to each set of parameters (for decreasing setpoints).

Setpoints (°C)	Error	Case 5	Case 6	Case 9	Case 10	Case 13	Case 14	Case 15	Case 16
25	-1	464	638	1084	1042	751	737	748	695
25	-2	4469	3237	7765	7524	5331	4276	4305	5072
27.5	-1	279	474	52	416	211	308	118	289
27.5	-3	1143	1987	918	203	1826	1816	1823	603
29	-1	336	588	532	520	158	188	172	210
29	-3	1002	1318	1256	1216	761	465	635	1001
32.5	-1	151	176	226	139	116	170	112	158
32.5	-3	375	393	381	506	367	380	382	286
Total Sum		8219	8811	12214	11566	9521	8340	8295	8314

Table 2.13 Performance of the variable range of SR-PID controller for the various temperature setpoints using the best set of parameters (Case 3 and Case 5).

Cases	Setpoints (°C)	Error	Case 3	Case 5
1	22.5	+3	369	384
2	22.5	+1	116	165
3	25	+3	565	387
4	25	+1	166	169
5	25	-1	628	464
6	25	-2	4496	4769
7	27.5	+3	1100	1132
8	27.5	+1	265	279
9	27.5	-1	267	279
10	27.5	-3	1107	1143
11	29	+3	3563	3700
12	29	+1	400	296
13	29	-1	196	336
14	29	-3	957	1002
15	32.5	-1	101	151
16	32.5	-3	364	375
Total Sum			14660	15031

The values were observed as $K_p = -36.17$, $K_i = -0.93$, and $K_d = -2.89$ (for the case of PID controller in standard split range scheme), and $K_p = -40.8$, $K_i = -1$, and $K_d = -3.7$ (for the case of PID controller in a variable range of split range scheme). These respective gain values were then used to obtain the simulation results for increasing temperature setpoints (with the same signs) and decreasing temperature setpoints (with opposite signs).

2.3 Results and discussion

In this chapter, the conventional PID controller and SR-PID controller (in standard and variable ranges) were used for controlling the temperature of the mixing process. The control strategy was simulated in MATLAB/Simulink environment. The performance of the controller was tested for different set points within the working range of temperature (22.5°C to 32.5°C). It was discussed in section 2.2.1 that the PID controller in all the three schemes was tuned for the following combination of flow rates (Q_{10} Q_{20}): (0.075 0.015), (0.04 0.02), (0.045 0.045), (0.02 0.03), (0.015 0.075) for temperature setpoints 22.5°C, 25°C, 27.5°C, 29°C, 32.5°C, respectively. The combination of flow rates for the extreme ends of the working range, i.e. 22.5°C and 32.5°C

are unique. However, for all other temperatures in the range, there are many possible combinations of the flow rates. Since it is not possible to tune the PID controller for each of the possible combination of flow rates for a particular temperature, it is important to investigate the performance of a controller tuned for one combination of the flow rates when this combination changes for the same temperature.

Table 2.14 lists some possible combinations of the flow rates (combination I, II and III) for same temperature setpoints. The controllers were tuned for combination II. It is clear from the table that the flow rates in combination I are lower than the flow rates for which the controller has been tuned. Further, the flow rates in combination III are higher than the flow rates for which the controllers have been tuned. However, for a given temperature setpoints, the different combinations of flow rates were possible within the entire range (0.015 0.075), of which some combinations of $(Q_{10} Q_{20})$ were tried as shown in Table 2.14.

Table 2.14 Combinations of steady-state flow rates.

Setpoints (°C)	Combination I ($Q_{10} Q_{20}$)	Combination II ($Q_{10} Q_{20}$)	Combination III ($Q_{10} Q_{20}$)
25	(0.035 0.0175)	(0.04 0.02)	(0.06 0.03)
27.5	(0.03 0.03)	(0.045 0.045)	(0.06 0.06)
29	(0.018 0.027)	(0.02 0.03)	(0.04 0.06)

The performances of the controllers were investigated with the combinations of $(Q_{10} Q_{20})$ for the setpoints shown in Table 2.14 and compared in terms of settling time, as shown in Table 2.15. In Table 2.15, the first, second and third rows (in all the cases) used combination I, combination II and combination III, respectively.

It can be inferred from Table 2.15 that irrespective of the combination of flow rates for a given temperature, a variable range of SR-PID controller provided better results in all the conditions (temperature rises/ falls of 10% or 30% within the entire range), as compared to the standard SR-PID and the conventional PID controllers.

Table 2.15 Performances of PID controller (in all the schemes) with different combinations of flow rates.

Cases	Setpoints (°C)	Error	PID Controller	SR-PID Controller (in Standard range)	SR-PID Controller (in Variable range)	Improvements in T_s (%)	
			T_s	T_s	T_s	SR-PID (standard) over PID	SR-PID (variable) over PID
1	25	+3	1002	744	573	25.75	42.81
	25	+3	987	734	565	25.82	42.96
	25	+3	782	554	389	26.84	49.25
2	27.5	+3	2172	1857	1588	18.5	33.89
	27.5	+3	1673	1365	1100	18.61	34.25
	27.5	+3	1223	921	660	24.69	46.03
3	29	+3	5310	4413	3576	16.81	31.66
	29	+3	5285	4395	3563	16.94	32.58
	29	+3	4560	3677	2852	19.36	37.46
4	25	-2	6668	5502	4511	17.39	32.15
	25	-2	6641	5482	4496	17.48	32.3
	25	-2	5736	4588	3610	20.01	37.06
5	27.5	-3	2186	1873	1604	18.32	33.62
	27.5	-3	1678	1372	1107	18.44	34.03
	27.5	-3	1232	929	668	24.58	45.78
6	29	-3	1524	1138	967	25.33	36.55
	29	-3	1509	1126	957	25.39	36.69
	29	-3	1061	814	598	26.28	43.64
7	25	+1	284	217	169	23.59	40.49
	25	+1	279	213	166	23.66	40.55
	25	+1	256	195	151	23.84	41.02
8	27.5	+1	465	363	286	21.94	38.49
	27.5	+1	439	340	265	22.55	39.64
	27.5	+1	412	318	245	22.82	40.53
9	29	+1	796	581	405	20.41	35.12
	29	+1	784	573	400	21.52	36.45
	29	+1	591	456	315	21.84	38.41
10	25	-1	1162	871	634	21.04	34.44
	25	-1	1149	862	628	22.09	36.08
	25	-1	963	682	452	22.18	37.06
11	27.5	-1	469	367	289	21.75	38.38
	27.5	-1	442	343	267	22.39	39.59
	27.5	-1	416	320	246	23.08	40.87
12	29	-1	356	271	199	23.38	44.1
	29	-1	349	267	196	23.5	44.84
	29	-1	322	243	174	24.53	45.96

It is also clear from the results that the controller tuned for one combination of flow rates at a temperature performs satisfactorily when the combination of the flow rates changes for the same

temperature. Therefore, further investigations in this work shall be made only for the combination of flow rates for which the controllers have been tuned.

The performance of the controller in all the schemes using the best set of gain parameters was evaluated on the basis of steady-state error (E_{ss}) and settling time (T_s), as shown in Tables 2.16–2.21. The investigations consist of studying the transient performance, the effect of system dynamics (dead time in the valve), and the effect of disturbance in the process. The performance of the controller is also examined in terms of utility consumptions.

2.3.1 Effect of dead time in the valve

Initially, the system performance was checked with zero dead time (Table 2.16). The simulation results were obtained for the different temperature setpoints. Based on these simulation results, a comparative study was made for the performance of the controller (in all the schemes) on the basis of steady-state error, E_{ss} and settling time, T_s (second), as shown in Table 2.16.

Table 2.16 Comparative study of the transient response of PID controller (in all the schemes) with no dead time.

Cases	Setpoints (°C)	Error	PID Controller		SR-PID Controller (in Standard range)		SR-PID Controller (in Variable range)		Improvements in T_s (%)	
			T_s	E_{ss}	T_s	E_{ss}	T_s	E_{ss}	SR-PID (standard) over PID	SR-PID (variable) over PID
1	22.5	+3	756	0.056	548	0.005	369	0.001	27.51	51.19
2	22.5	+1	211	0.031	155	0.003	116	0	26.54	45.02
3	25	+3	987	0.055	734	0.007	565	0.001	25.63	42.96
4	25	+1	279	0.034	213	0.004	166	0	23.66	40.55
5	25	-1	1149	0.061	862	0.009	628	0.003	22.09	36.08
6	25	-2	6641	0.073	5482	0.016	4496	0.008	17.45	32.3
7	27.5	+3	1673	0.054	1365	0.0098	1100	0.004	18.41	34.25
8	27.5	+1	439	0.037	340	0.006	265	0.001	22.55	39.64
9	27.5	-1	442	0.038	343	0.006	267	0.001	22.39	39.59
10	27.5	-3	1678	0.054	1372	0.0096	1107	0.004	18.24	34.03
11	29	+3	5285	0.071	4395	0.02	3563	0.009	16.84	32.58
12	29	+1	784	0.042	573	0.007	400	0.002	21.52	36.45
13	29	-1	349	0.033	267	0.004	196	0	23.5	44.84
14	29	-3	1509	0.056	1126	0.0072	957	0.001	25.38	36.69
15	32.5	-1	205	0.03	151	0.003	101	0	26.34	48.56
16	32.5	-3	752	0.057	545	0.006	364	0.001	27.53	51.6

It is clear from the above table that the system took the longest to settle in case of the conventional PID control scheme as only one flow rate was manipulated in this scheme. A decrease in settling time was observed in case of standard split range control scheme as both the flow rates were manipulated. However, as the controller output was effectively split in the variable range split range control resulting in simultaneous manipulation of both the flow rates, minimum settling times were observed in this case.

The table also shows the percentage improvement in settling time in case of the SR (standard) and SR (variable) over PID control scheme. It is clear that the percentage improvement is more in case of SR (variable) in all the cases. However, this percentage improvement decreases for a desired rise in temperature when we move from the lower bound of the working range towards the higher bound. This is because for higher temperatures, the inflow rate of the hot water is higher. For a further rise in temperature, the controller output is split in a manner that tends to increase the inflow rate of hot water even more. However, the flow rates can be varied only within the permitted limits. Therefore, the desired control action cannot be implemented fully in the higher ranges of temperature. The gap between the desired and actual control action implemented increases as we move upwards in the working range. As a result, the system takes comparatively longer to settle to the desired setpoint as we move upwards in the working range. This also results in decrease in the percentage improvement in the performance of the SR (variable) scheme over the conventional PID control scheme as we move upwards in the working range. However, the performance is still better than the standard SR strategy. For similar reason, a decrease in percentage improvement in the performance of SR (variable) over the conventional PID control scheme is observed for falling temperatures as we move from higher to lower bounds of the working range of temperatures.

By considering the dead time of 0.25 second and 0.5 second in the valve, the performances of the controllers were further investigated. The comparison between the performances of the controllers on the basis of simulation studies are presented in [Tables 2.17](#) and [2.18](#).

Table 2.17 Comparative study of the transient response of PID controller (in all the schemes) with a dead time of 0.25 sec.

Cases	Setpoints (°C)	Error	PID Controller		SR-PID Controller (in Standard range)		SR-PID Controller (in Variable range)		Improvements in T_s (%)	
			T_s	E_{ss}	T_s	E_{ss}	T_s	E_{ss}	SR-PID (standard) over PID	SR-PID (variable) over PID
1	22.5	+3	922	0.058	657	0.007	441	0.002	28.74	52.17
2	25	-1	1401	0.064	1034	0.011	736	0.003	26.2	47.47
3	27.5	+3	2041	0.057	1638	0.012	1311	0.005	19.75	35.77
4	27.5	-3	2047	0.057	1646	0.012	1321	0.004	19.59	35.47
5	29	+1	956	0.044	687	0.009	468	0.002	28.14	51.05
6	32.5	-3	917	0.059	654	0.009	435	0.002	28.68	52.56

Table 2.18 Comparative study of the transient response of PID controller (in all the schemes) with a dead time of 0.5 sec.

Cases	Setpoints (°C)	Error	PID Controller		SR-PID Controller (in Standard range)		SR-PID Controller (in Variable range)		Improvements in T_s (%)	
			T_s	E_{ss}	T_s	E_{ss}	T_s	E_{ss}	SR-PID (standard) over PID	SR-PID (variable) over PID
1	22.5	+3	1171	0.061	818	0.01	532	0.002	30.15	54.57
2	25	-1	1780	0.068	1287	0.017	875	0.004	27.7	50.84
3	27.5	+3	2593	0.061	2039	0.02	1586	0.005	21.37	38.84
4	27.5	-3	2600	0.06	2049	0.019	1595	0.005	21.19	38.65
5	29	+1	1215	0.047	856	0.012	555	0.003	29.55	54.32
6	32.5	-3	1165	0.063	814	0.011	527	0.003	30.13	54.76

On the basis of the simulation results, it can be said from [Tables 2.16–2.18](#) that the variable range of SR-PID controller provided better results as compared to the standard SR-PID and the conventional PID controllers. With the introduction of dead time in the valve, the manipulated variables will take longer to actually change. Therefore, the system will give a sluggish response, resulting in higher settling times in all the cases, as compared to a system with no dead time. It can also be inferred that the settling times increase with an increase in dead time of the valve.

2.3.2 Effect of process disturbance

The system under consideration was further checked for the effect of disturbance (considered in the forward path in the process at $t=30$ seconds) on the performance of the controller using impulse function as disturbance. The performance of the controller (in all the three schemes) was tested at the various temperature setpoints considered earlier, and the results are provided in Tables 2.19–2.21.

Table 2.19 Comparative study of the transient response of PID controller (in all the schemes) with disturbance and no dead time.

Cases	Setpoints (°C)	Error	PID Controller		SR-PID Controller (in Standard range)		SR-PID Controller (in Variable range)		Improvements in T_s (%)	
			T_s	E_{ss}	T_s	E_{ss}	T_s	E_{ss}	SR-PID (standard) over PID	SR-PID (variable) over PID
1	22.5	+3	781	0.057	569	0.008	387	0.002	27.14	50.45
2	22.5	+1	228	0.031	168	0.004	126	0.001	26.32	44.74
3	25	+3	1013	0.056	751	0.009	579	0.002	25.86	42.84
4	25	+1	309	0.035	239	0.007	189	0.001	22.65	38.83
5	25	-1	1324	0.064	1031	0.011	791	0.006	20.13	33.26
6	25	-2	7857	0.076	6742	0.017	5705	0.01	14.19	27.39
7	27.5	+3	1742	0.057	1427	0.012	1157	0.006	18.08	33.58
8	27.5	+1	503	0.037	401	0.007	322	0.003	20.28	35.98
9	27.5	-1	506	0.038	403	0.007	324	0.003	20.36	35.97
10	27.5	-3	1745	0.057	1433	0.011	1163	0.005	17.88	33.35
11	29	+3	6577	0.074	5733	0.021	4848	0.009	12.83	26.29
12	29	+1	891	0.045	675	0.008	501	0.003	20.04	33.77
13	29	-1	388	0.034	302	0.004	228	0.001	22.16	41.24
14	29	-3	1539	0.057	1147	0.009	975	0.002	25.47	36.65
15	32.5	-1	229	0.031	171	0.004	118	0.001	25.33	48.47
16	32.5	-3	771	0.059	560	0.009	377	0.002	27.37	51.1

The results showed from Tables 2.19 that even in case of disturbance, the variable range of split range PID controller gave a better performance as compared to the conventional PID controller and a standard split range PID controller. It can be said from Tables 2.16 (results with no disturbance) and 2.19 (results with disturbance) that when a disturbance is introduced in the

process, the performance of the control system will be affected in terms of settling time, steady-state error.

Table 2.20 Comparative study of the transient response of PID controller (in all the schemes) with a dead time of 0.25 sec. and disturbance.

Cases	Setpoints (°C)	Error	PID Controller		SR-PID Controller (in Standard range)		SR-PID Controller (in Variable range)		Improvements in T_s (%)	
			T_s	E_{ss}	T_s	E_{ss}	T_s	E_{ss}	SR-PID (standard) over PID	SR-PID (variable) over PID
1	22.5	+3	949	0.059	680	0.008	460	0.002	28.35	51.53
2	25	-1	1578	0.066	1205	0.012	899	0.004	25.84	43.03
3	27.5	+3	2112	0.058	1702	0.013	1369	0.006	19.41	35.18
4	27.5	-3	2116	0.058	1709	0.012	1378	0.005	19.23	34.88
5	29	+1	1065	0.045	791	0.009	570	0.003	25.73	46.48
6	32.5	-3	938	0.061	671	0.01	449	0.002	28.46	52.13

Table 2.21 Comparative study of the transient response of PID controller (in all the schemes) with a dead time of 0.5 sec. and disturbance.

Cases	Setpoints (°C)	Error	PID Controller		SR-PID Controller (in Standard range)		SR-PID Controller (in Variable range)		Improvements in T_s (%)	
			T_s	E_{ss}	T_s	E_{ss}	T_s	E_{ss}	SR-PID (standard) over PID	SR-PID (variable) over PID
1	22.5	+3	1199	0.062	841	0.011	551	0.002	29.86	54.05
2	25	-1	1958	0.069	1458	0.018	1039	0.005	26.54	46.94
3	27.5	+3	2665	0.062	2103	0.021	1644	0.006	21.09	38.31
4	27.5	-3	2670	0.063	2112	0.02	1652	0.006	20.9	38.14
5	29	+1	1325	0.047	960	0.012	657	0.004	27.55	50.42
6	32.5	-3	1187	0.065	831	0.013	541	0.003	29.99	54.42

It can also be observed that in the presence of disturbance, the system will take more time to attain the desired state of temperature as compared to the system without disturbance. This time increases further with an increase in dead time of the valve (Table 2.19–2.21).

2.3.3 Utility consumption

The performance of the controller was also investigated on the basis of utility consumption, i.e., cold utility consumption (u_c) and hot utility consumption (u_h). The utility consumption achieved using the controller in all the schemes is presented in [Tables 2.22–2.27](#).

Table 2.22 Comparative study of the amount of Q_1 and Q_2 flows using PID controller (in all the schemes) with no dead time.

Cases	Setpoints (°C)	Error	PID Controller		SR-PID Controller (in Standard range)		SR-PID Controller (in Variable range)	
			u_c	u_h	u_c	u_h	u_c	u_h
1	22.5	+3	23.52	11.34	16.71	28.83	8.27	25.28
2	22.5	+1	18.12	3.17	10.23	12.64	7.18	11.26
3	25	+3	32.45	19.74	18.11	55.82	10.09	40.8
4	25	+1	19.75	5.58	13.37	16.07	9.02	14.46
5	25	-1	45.96	34.16	52.19	14.04	46.41	12.14
6	25	-2	265.64	189.26	334.94	107.29	296.4	68.68
7	27.5	+3	48.32	75.29	29.94	97.96	17.03	82.38
8	27.5	+1	21.46	19.76	14.28	23.96	10.11	22.13
9	27.5	-1	19.89	22.05	24.05	14.67	21.39	11.17
10	27.5	-3	75.51	48.64	98.26	30.04	82.89	17.41
11	29	+3	139.34	158.55	88.32	233.27	54.36	208.62
12	29	+1	22.77	23.52	16.13	27.68	12.57	25.75
13	29	-1	6.98	20.37	17.71	12.35	15.47	9.88
14	29	-3	30.18	38.92	56.22	26.76	45.45	15.17
15	32.5	-1	3.07	17.79	12.93	10.01	10.95	7.03
16	32.5	-3	11.28	23.42	28.67	16.57	25.21	8.07

It can be inferred from [Tables 2.22](#) that in case of variable range SR-PID controller, the total utility consumption (cold water plus hot water) was observed to be lesser in all the cases as compared to the conventional PID controller and a standard SR-PID controller. It can also be seen from table that hot utility consumption (u_h)/ cold utility consumption (u_c) increases by rising/ falling the temperature setpoint with the same value of error. This is because of higher temperature which requires more heating, whereas lower temperature requires more cooling.

Table 2.23 Comparative study of the amount of Q_1 and Q_2 flows using PID controller (in all the schemes) with disturbance and no dead time.

Cases	Setpoints (°C)	Error	PID Controller		SR-PID Controller (in Standard range)		SR-PID Controller (in Variable range)	
			u_c	u_h	u_c	u_h	u_c	u_h
1	22.5	+3	25.21	11.72	17.91	30.65	8.87	26.82
2	22.5	+1	19.57	3.42	11.04	13.49	7.68	11.94
3	25	+3	35.05	20.26	19.47	59.56	10.8	43.25
4	25	+1	21.33	6.18	14.37	17.15	9.65	15.33
5	25	-1	52.96	38.31	55.95	14.92	49.66	11.82
6	25	-2	314.28	204.21	360.06	114.48	327.15	72.8
7	27.5	+3	53.27	78.39	30.86	98.45	18.22	86.51
8	27.5	+1	23.18	22.64	15.35	25.57	10.82	23.46
9	27.5	-1	22.77	23.79	25.85	15.65	22.89	11.84
10	27.5	-3	78.53	53.84	99.26	31.93	87.98	18.87
11	29	+3	150.49	197.31	94.94	248.9	58.17	221.14
12	29	+1	24.41	26.73	17.29	29.42	12.98	27.32
13	29	-1	7.76	21.98	19.04	13.18	16.55	10.47
14	29	-3	30.78	41.99	60.44	28.55	48.63	16.08
15	32.5	-1	3.44	19.2	13.9	10.68	11.72	7.45
16	32.5	-3	11.57	24.9	30.73	17.61	26.21	8.58

Table 2.24 Comparative study of the amount of Q_1 and Q_2 flows using PID controller (in all the schemes) with a dead time of 0.25 sec.

Cases	Setpoints (°C)	Error	PID Controller		SR-PID Controller (in Standard range)		SR-PID Controller (in Variable range)	
			u_c	u_h	u_c	u_h	u_c	u_h
1	22.5	+3	27.42	13.83	18.28	38.23	8.73	31.76
2	25	-1	56.04	38.07	59.08	16.63	57.13	14.19
3	27.5	+3	58.69	91.85	35.52	119.28	20.49	98.21
4	27.5	-3	92.12	58.90	119.64	35.71	98.84	20.63
5	29	+1	26.69	28.68	19.65	31.39	11.97	29.37
6	32.5	-3	13.76	27.29	38.18	18.15	31.17	8.38

Table 2.25 Comparative study of the amount of Q_1 and Q_2 flows using PID controller (in all the schemes) with a dead time of 0.5 sec.

Cases	Setpoints (°C)	Error	PID Controller		SR-PID Controller (in Standard range)		SR-PID Controller (in Variable range)	
			u_c	u_h	u_c	u_h	u_c	u_h
1	22.5	+3	36.07	17.57	24.82	46.21	10.67	37.67
2	25	-1	71.2	40.82	76.74	18.95	73.44	16.19
3	27.5	+3	74.91	116.69	44.97	148.38	24.66	119.05
4	27.5	-3	117	75.18	149.04	45.26	118.69	25.37
5	29	+1	30.53	36.45	23.29	44.46	13.28	38.74
6	32.5	-3	17.48	35.92	46.15	24.64	37.59	10.30

Table 2.26 Comparative study of the amount of Q_1 and Q_2 flows using PID controller (in all the schemes) with a dead time of 0.25 sec. and disturbance.

Cases	Setpoints (°C)	Error	PID Controller		SR-PID Controller (in Standard range)		SR-PID Controller (in Variable range)	
			u_c	u_h	u_c	u_h	u_c	u_h
1	22.5	+3	29.39	14.24	19.56	40.56	9.36	33.76
2	25	-1	63.12	40.47	63.22	17.64	61.24	15.08
3	27.5	+3	62.92	95.04	38.01	126.5	21.97	104.4
4	27.5	-3	95.22	62.61	128.0	37.89	105.9	21.93
5	29	+1	28.61	31.95	21.03	33.3	12.83	31.22
6	32.5	-3	14.07	29.01	40.85	19.26	33.41	8.91

Table 2.27 Comparative study of the amount of Q_1 and Q_2 flows using PID controller (in all the schemes) with a dead time of 0.5 sec. and disturbance.

Cases	Setpoints (°C)	Error	PID Controller		SR-PID Controller (in Standard range)		SR-PID Controller (in Variable range)	
			u_c	u_h	u_c	u_h	u_c	u_h
1	22.5	+3	38.67	17.98	26.56	49.03	11.44	40.04
2	25	-1	72.32	43.39	82.11	20.11	78.73	17.21
3	27.5	+3	80.3	110.9	48.12	157.4	26.44	126.6
4	27.5	-3	111.3	79.92	159.4	48.02	127.2	26.97
5	29	+1	32.73	39.75	24.92	47.17	14.24	41.18
6	32.5	-3	17.81	38.18	49.38	26.14	40.3	10.95

It can also be said that in case of conventional PID controller, the hot utility consumption for rising setpoints and the cold utility consumption for falling setpoints are found to be lower, as compared to the SR-PID controller in standard and variable range schemes. It is because of the consideration of one manipulated variable (Q_1 or Q_2 , respectively) in case of conventional PID controller to control the controlled variable, keeping other one (Q_2 or Q_1 , respectively) constant at its steady-state value.

It can also be inferred from [Tables 2.24–2.25](#) that hot and cold utility consumptions increase with an increase in dead time of the valve. These utility consumptions (u_h and u_c) increase further in case of introduction of disturbance and dead times ([Tables 2.26–2.27](#)).

This chapter mainly highlighted the conventional PID controller, a standard split range PID controller, and the variable range of split range PID controller for temperature control of a mixing process. For the tuning purpose, the Z-N tuning method was applied to find the controller gains. The simulation results obtained were compared in terms of settling time and steady-state error. It was observed that the variable range of split range PID controller outperformed conventional PID controller as well as a standard split range PID controller in all the scenarios, i.e., the effect of dead time in the valve, the effect of disturbance in the process, and utility consumption. The steady-state error in the case of the variable range of the SR-PID controller

was found to be minimum (equivalent to zero) in all the scenarios (Tables 2.16–2.21). It was also observed that the use of a variable range of SR-PID controller with the Z-N tuning method produced a response that led to a significant overshoot and settling time. Furthermore, this work can be extended by using nature-inspired optimization techniques for enhancing the performance of the variable range of the SR-PID controller.

Controller: Tuning and Performance Evaluation

This chapter presents the use of nature- inspired algorithms for tuning the PID controller used in the variable range split range control scheme for temperature control of a mixing process. A comparison of the performances of the controllers tuned using these algorithms and the conventional Z-N method is also presented. Further, an improved nature inspired algorithm is proposed to enhance the performance of the controller.

The variable range split range controller exhibited the best performance in controlling the temperature of a mixing process in the last chapter. However, the said controller was tuned using Z-N method. As the nature-inspired optimization algorithms have been proved to be efficient tools for tuning PID controllers, it is proposed to further enhance the performance of the variable range split range controller by tuning the controller using nature-inspired optimization algorithms.

The particle swarm optimization (PSO) algorithm is one of the most used swarm intelligent-based techniques (nature-inspired techniques), which utilizes the swarming behavior of birds. The main advantage of the PSO algorithm is that it is easy to implement, does not require gradient information (a detailed mathematical description of the process), and determines the optimal gain parameters (K_p , K_i , and K_d) depending on the objective function. PSO converges fast because this algorithm is easily employed with few parameters to be adjusted. Various researchers are reported their work for optimizing the controller using the PSO algorithm (Safarzadeh & Noori-kalkhoran, 2021; Mien *et al.*, 2020; Rajesh, 2019; Thamallah *et al.*, 2019; Greeshma, 2019; Gao *et al.*, 2018).

Since most of the earlier developed optimization algorithms such as PSO, gravitational search algorithm (GSA), and differential evolution (DE) do not have parameters to determine specific

iterations for the exploitation or the exploration phase as they utilize only one format for updating search agents' positions, so the chances of entrapment into local optima is typically increased (Ho-Huu *et al.*, 2015).

Hence, to improve the system performance, a Whale optimization algorithm is used that shows the hunting behavior of humpback whales in nature. It takes lesser iterations to reach the optimal global solution in comparison to the previously developed algorithms. The whale optimization algorithm leads PSO, GSA, DE, and other optimization techniques in terms of solutions accuracy and stability (Fan *et al.*, 2021). Various authors have used the WOA algorithm for optimizing the controller parameters (Karam & Awad, 2020; Mosaad *et al.*, 2019; Loucif *et al.*, 2019; Ding *et al.*, 2018).

In the WOA algorithm, convergence and speed depend on the control parameter 'a', which is used in the encircling prey mechanism. This parameter has a major effect on WOA's performance (Zhong & Long, 2017). As a result, this algorithm performs with poor convergence speed in both the exploitation and the exploration stages (Pecora & Carroll, 2015; Saidala & Devarakonda, 2018). As a result, an effective improvement is required for balancing exploitation and exploration (Abdel-Basset *et al.*, 2018). Moreover, the whale optimization algorithm utilizes the encircling process in the search space, which is less capable to jump out from local optima. As a result, it leads to poor performance (Nagaraj *et al.*, 2015). This algorithm also has another limitation when improving the best solution after every iteration (Xu Z. *et al.*, 2018).

Further, to enhance the system performance, the Moth flame optimization algorithm is used that simulates the navigational behavior of the moths in nature. It has highly opted for optimization involving high exploration, exploitation, and avoidance of local optima, and also combines capabilities of global exploration and local search (Acharyulu *et al.*, 2020; Mohanty, 2019; Dhyani *et al.*, 2018; Ng Shin Mei *et al.*, 2017). This makes it a very promising algorithm for finding applications in various fields (Acharyulu *et al.*, 2020; Mohanty, 2019; Dhyani *et al.*, 2018; Mohanty *et al.*, 2018; Ng Shin Mei *et al.*, 2017; Allam *et al.*, 2016; Jangir *et al.*, 2016).

These nature-inspired optimization techniques have been discussed in detail below:

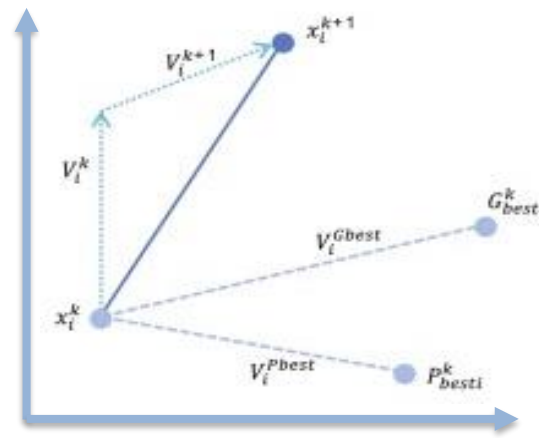
3.1 Particle swarm optimization

PSO is the population-based algorithm that was proposed by Eberhart and Kennedy in 1995 (Kennedy & Eberhart, 1995). It simulates the social behavior of bird flocking. The PSO algorithm works by having a population (known as a swarm) of particles. Each particle represents the candidate solution to the optimization problems. The movement of the particles is directed by their own best position in the search space as well as the global best position of the population. Each particle's performance is evaluated on the basis of a fitness function that also varies depending on the optimization problem. This performance shows how near the particle is to the global optimum.

The two main operators in the PSO algorithm are velocity update and position update. Each particle is moved toward its prior best position and the global best position throughout each iteration. A new velocity for each particle is determined at each iteration on the basis of its current velocity, distance from its prior best position, and distance from the best global position. The updated velocity value is then utilized to compute the particle's next position in the search space. This procedure is then repeated a certain number of times or until a minimal error is obtained (Eberhart *et al.*, 2001). Fig. 3.1 shows the bird flocking behavior and the displacement of the i^{th} particles in the solution space during the k and $(k+1)$ iterations (Eberhart *et al.*, 2001).



(a)



(b)

Fig. 3.1 PSO inspiration: (a) Bird flocking behavior, (b) Movement of the i^{th} particles in the swarm space. (Source: Zeng *et al.*, 2014)

The evolution of displacement of the particles is influenced by the particle's best position (P_{best}), and the global best position (G_{best}). Each particle modifies its path in order to reach the best fitting solution achieved thus far. This value is known as P_{best} . Each particle also adjusts its path towards the best prior position achieved by any member of its neighborhood. This is referred to as G_{best} .

The procedure of the PSO algorithm is given step by step as follows (Eberhart *et al.*, 2001):

Step 1: Initialization: The position and velocity of all the particles are set at random within pre-defined limits.

Step 2: Velocity and position updating: The velocity and position of each particle are updated according to the following equations:

$$V_i^{k+1} = w \cdot V_i^k + c_1 \cdot r_1 (P_{best\ i} - x_i^k) + c_2 r_2 (G_{best} - x_i^k) \quad (3.1)$$

$$x_i^{k+1} = x_i^k + V_i^{k+1} \quad (3.2)$$

where V_i^k and x_i^k are the current velocity and the position of the i^{th} particle at iteration k , respectively; V_i^{k+1} represents the new velocity of the i^{th} particle at iteration k ; w is a population controlling the flying dynamics; c_1 and c_2 are factors controlling the related weighting of corresponding terms; $P_{best\ i}$ is personal best of i^{th} particle and G_{best} is global best of the population, respectively; r_1 and r_2 specify the random numbers in the range $[0, 1]$; x_i^{k+1} denotes the position of the i^{th} particle at the next iteration $k+1$.

The use of random variables in the PSO algorithm provides the capacity to search in a randomized manner. The factors c_1 and c_2 indicate the weighting of the random acceleration that accelerate each particle toward P_{best} and G_{best} positions. As a result, adjusting these constants alters the amount of stress in the system. Low values enable particles to wander far away from target locations before being pulled back, whereas high values allow rapid movement toward target locations (Eberhart & Shi, 2001). The weighting factors minimize the unavoidable tradeoff between exploration and exploitation.

Step 3: Memory updating: Update $P_{best\ i}$ and G_{best} when the condition is met.

$$P_{best\ i} = x_i^k \quad \text{when } f(x_i^k) < f(P_{best\ i}) \quad (3.3)$$

$$G_{best} = x_i^k \quad \text{when } f(x_i^k) > f(G_{best}) \quad (3.4)$$

where $f(t)$ denotes the objective function to be maximized.

Step 4: Termination checking: This algorithm repeats Steps 2 to 4 until the end criteria are satisfied. Once the algorithm is ended, it provides the best optimal solution as $P_{best\ i}$ and G_{best} .

The velocities of particles in each dimension are restricted to a maximum velocity V_{max} . Therefore, V_{max} is a crucial parameter that defines the fineness, or resolution, with which areas between the current position and the targeted (best so far) position are explored. If V_{max} is too high, particles may fly through suitable solutions, whereas particles may not explore adequately beyond locally favorable areas if V_{max} is too low (Eberhart *et al.*, 2001). Indeed, they may fall in local optima, not able to accelerate in order to achieve a better position in the search space (Eberhart & Shi, 2001). The flowchart for the basis PSO algorithm is shown in Fig. 3.2.

The following parameters were considered for the PSO algorithm that is listed in Table 3.1.

Table 3.1 PSO algorithm parameters.

S. No.	Parameters	Values
1	Number of search agents	20
2	Dimension	3
3	Maximum no. of iterations	50
4	Lower bounds (for increasing setpoints)	[-70 -2 -10]
5	Lower bounds (for decreasing setpoints)	[0 0 0]
6	Upper bounds (for increasing setpoints)	[0 0 0]
7	Upper bounds (for decreasing setpoints)	[70 2 10]

The number of iterations (K) was set at 50 to achieve rapid convergence and good results (Gaing, 2004). The initial population of 20 particles was generated randomly in search space that may attain the various final solutions. The result may or may not be the optimal solution if only one trial was performed. Therefore, to solve this problem, multiple trials were performed, and the best solution was identified between all trials. It involved running the PSO algorithm 10 times (trial number =10) and finding the optimum gains of the SR-PID controller associated with the minimum fitness value. The lower and upper bounds for increasing setpoints and decreasing setpoints were taken, as shown in Table 3.1. These values were taken on the basis of the

controller gain values obtained from the Ziegler-Nichols method for different set points within the working range. The values of lower and upper bounds were decided in a way to ensure that the solution space not only includes all the gain values obtained using the Z-N method but also provides the opportunity to the PSO algorithm to explore probable solutions beyond the range obtained by the Z-N method.

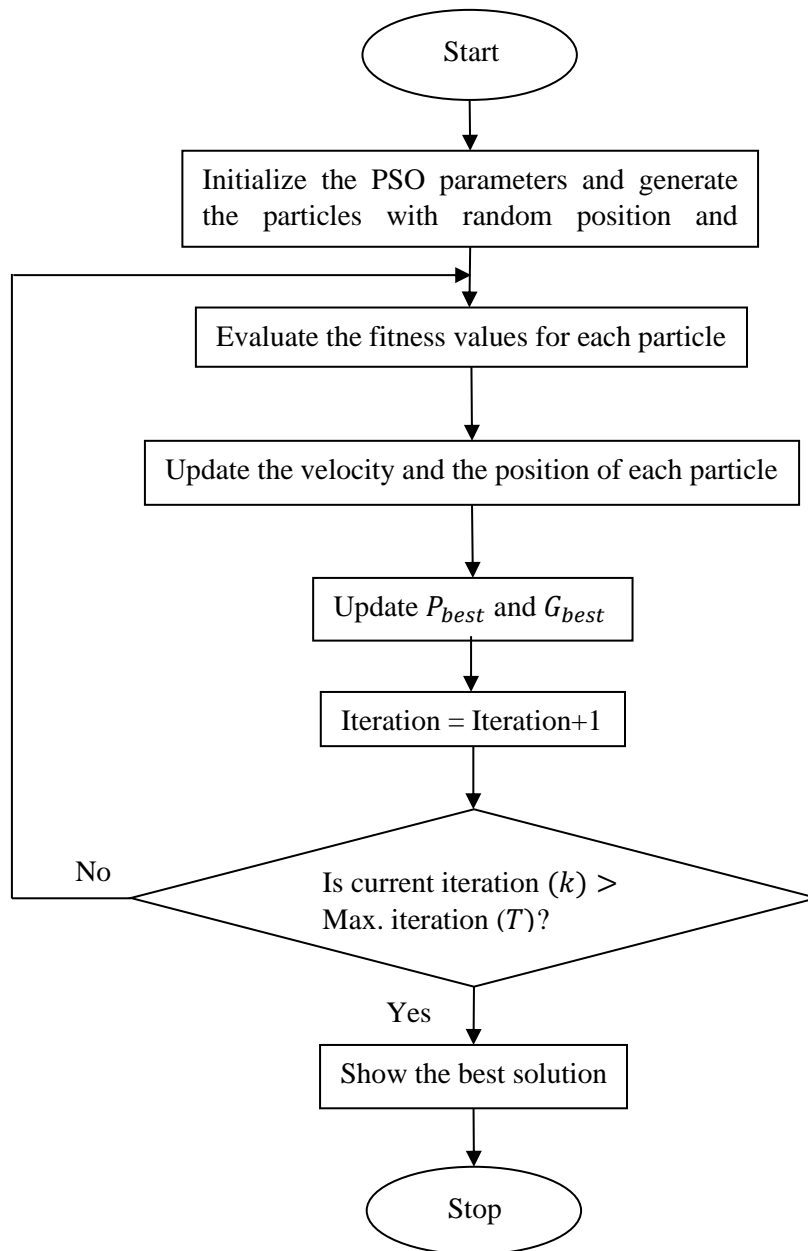


Fig. 3.2 Flow chart of PSO algorithm.

3.2 Whale optimization algorithm

The Whale Optimization Algorithm (WOA) was proposed by Mirjalili and Lewis (Mirjalili & Lewis, 2016). It is a metaheuristic algorithm based on the hunting behavior of humpback whales. According to the humpback whales' hunting strategy, once prey (such as small fishes, krills) is discovered, humpback whales dive deeply and create bubble nets in a spiral shape around the prey. This special hunting strategy used by the humpback whales is known as the bubble-net attacking method. The bubble-net attacking behavior of humpback whales is shown in Fig. 3.3.

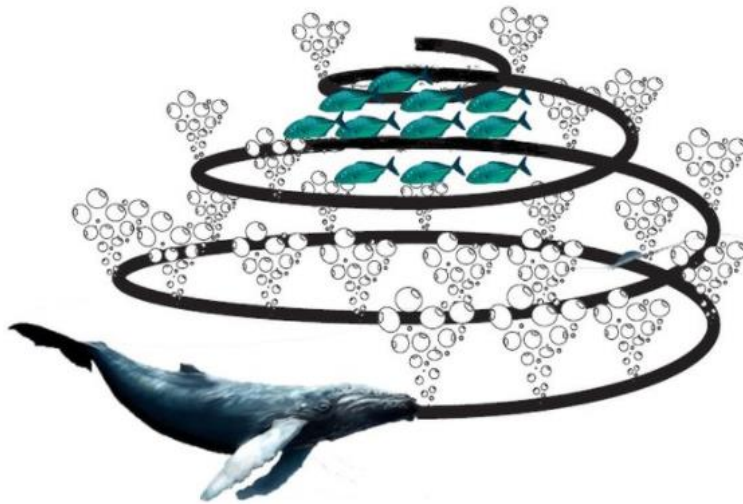


Fig. 3.3 Bubble-net attacking behavior of humpback whales. (Source: Mirjalili & Lewis, 2016)

The mathematical model of the WOA algorithm can be explained in the following phases (Mirjalili & Lewis, 2016):

- ❖ Encircling Prey
- ❖ Bubble-net attacking method
- ❖ Search for prey

3.2.1 Encircling prey

In this phase, the humpback whale identifies the position of prey and encircles them. Since initialization of the search agents is random in the search-space, hence, this algorithm treats the current best position as the position of the prey. The other agents will update their positions according to the following equations:

$$D = |C \cdot X^*(k) - X(k)| \quad (3.5)$$

$$X(k + 1) = X^*(k) - A \cdot D \quad (3.6)$$

where, $X(k)$ represents the whale's position vector, k denotes the current iteration, $X^*(k)$ is the prey position vector, and A and C are coefficient vectors, which can be determined as follows:

$$A = 2a \cdot r - a \quad (3.7)$$

$$C = 2 \cdot r \quad (3.8)$$

where a signifies a variable linearly decreased from 2 to 0 during iterations, and r represents a random vector in $[0 \ 1]$.

3.2.2 Bubble-net attacking method (Exploitation phase)

The humpback whales use a bubble-net method to capture the prey, as described previously. Two mechanisms are adopted to explain the bubble-net behavior of humpback whales for capturing prey. These mechanisms are mathematically represented as follows:

Shrinking encircling mechanism

This approach is dependent on the value of a . In order to achieve the shrinking behavior, the value of a is decreased from 2 to 0 throughout iterations, and $|A| < 1$. The shrinking encircling mechanism can be represented as shown in [Fig. 3.4 \(a\)](#).

Spiral updating position

This approach calculates the distance D' between the whale's position (X, Y) and the prey position (X^*, Y^*) as shown in [Fig. 3.4 \(b\)](#). Following this, a spiral equation can be formed to update the position of the whales as given below:

$$X(k + 1) = D' \cdot e^{bt} \cdot \cos(2\pi t) + X^*(k) \quad (3.9)$$

where, $D' = |X^*(k) - X(k)|$ that represents the distance of the i^{th} whale to the prey, b indicates a constant that specifies the logarithmic spiral's shape, t denotes a random number in $[-1 \ 1]$.

During hunting, humpback whales utilize both of the approaches described above, swimming around the prey in a shrinking circle while also swimming along a spiral path at the same time.

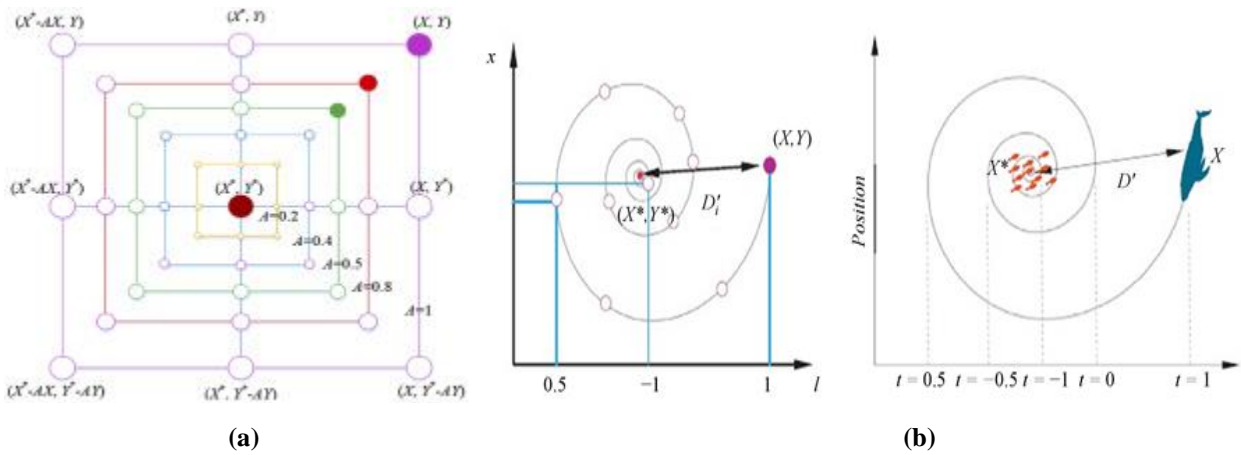


Fig. 3.4 Bubble-net search: (a) Shrinking mechanism, and (b) Spiral updating position. (Source: Mirjalili & Lewis, 2016)

In order to take into consideration this behavior, it is supposed that they use a probability of 50% for the shrinking approach and the same probability for the spiral model to update their position, which may be described as follows (Mirjalili & Lewis, 2016):

$$X(k + 1) = \begin{cases} X^*(k) - A \cdot D, & \text{if } p < 0.5 \\ D' \cdot e^{bt} \cdot \cos(2\pi t) + X^*(k), & \text{if } p \geq 0.5 \end{cases} \quad (3.10)$$

where p is the probability number that lies in the range of 0 and 1.

3.2.3 Search for prey (Exploration phase)

In this process, it is assumed that the humpback whales search for prey on a random basis, depending on their position each other. The whales' position is updated based on a randomly chosen whale rather than the best whale computed so far. The exploration phase can be described mathematically as follows (Mirjalili & Lewis, 2016):

$$D = X_{rand}(k) - X(k) \quad (3.11)$$

$$X(k + 1) = X_{rand}(k) - A \cdot D \quad (3.12)$$

A flow chart of the WOA algorithm is shown in Fig. 3.5, which conveys the mechanism of the algorithm.

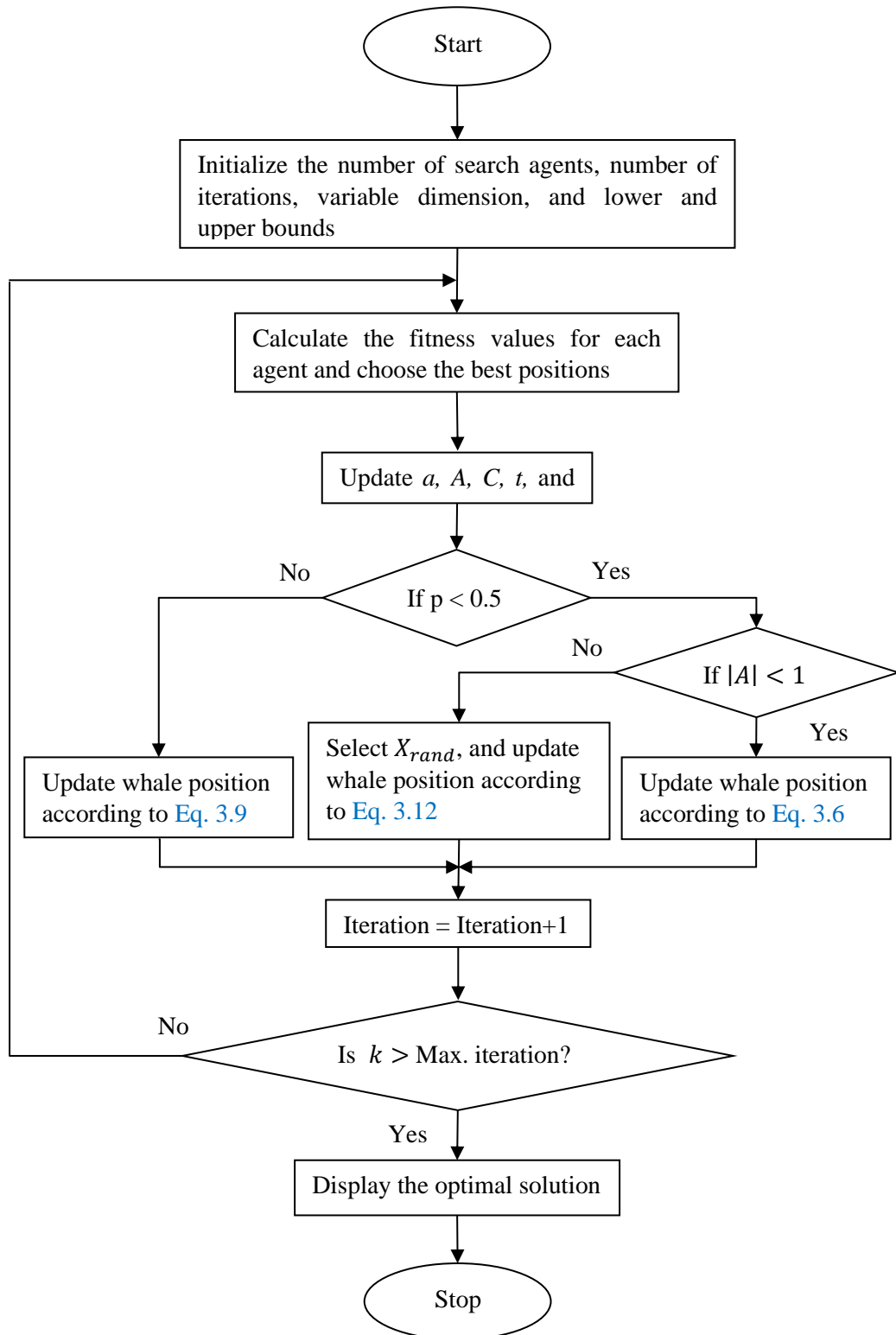


Fig. 3.5 Flowchart of WOA.

The same parameters were considered in the WOA algorithm as mentioned in Table 3.1

3.3 Moth flame optimization

Moth flame optimization is a population-based algorithm developed by Mirjalili in 2015 (Mirjalili, 2015). It is a nature-inspired technique for providing an effective and appropriate solution to the problem of optimization (Mirjalili, 2015). This algorithm is inspired by the navigation of moths in nature. Moths set up their path by keeping a fixed angle with respect to the moon. This notion of moths is highly useful for their straight flight, especially when the light source is away. However, when the light source is nearby, moths move in a spiral path around it. Fig. 3.6 demonstrates the conceptual model for the transverse orientation and the moths' spiral flying path.

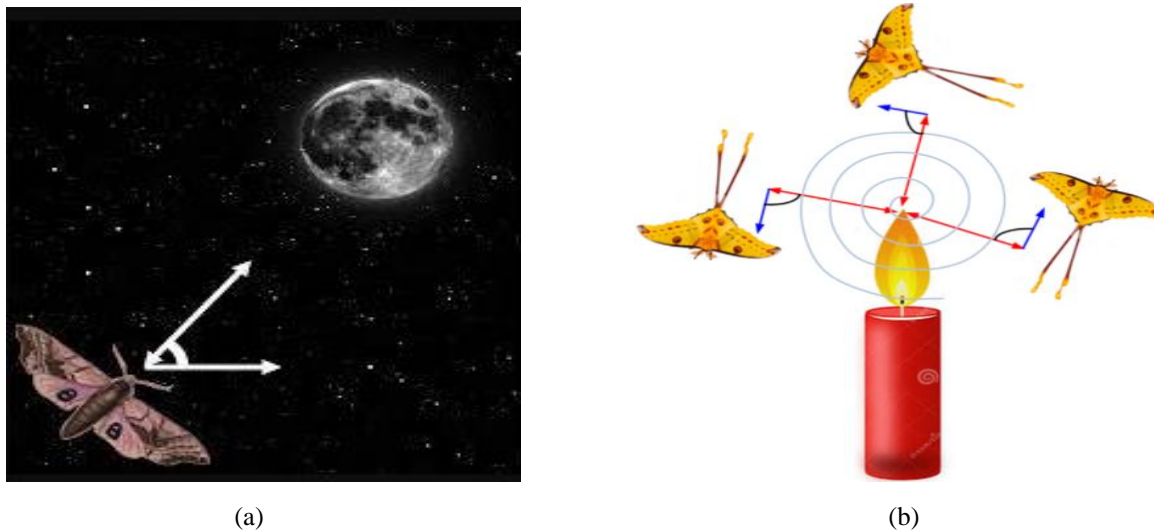


Fig. 3.6 MFO inspiration: (a) Transverse orientation, (b) Spiral flying path around a close light source. (Source: Mirjalili, 2015)

In this algorithm, moths are the search agents, and flames are the most recent promising positions of the moths obtained during the search for the solution space. Initially, the matrices of moths M and flames F have the same dimension (Mirjalili, 2015). In the beginning, each moth moves around its corresponding flame in F along a spiral path to exploit the space around the flame. The moth positions are updated as follows:

$$S(M_i, F_j) = D_i \cdot e^{bt} \cdot \cos(2\pi t) + F_j \quad (3.13)$$

where S signifies the spiral function, M_i and F_j signify the i^{th} moth and the j^{th} flame, b indicates a constant that shows spiral movement shape, t represents a random number within the range from

-1 to 1, and D_i indicates the distance from i^{th} moth to j^{th} flame calculated as given below:

$$D_i = |F_j - M_i| \quad (3.14)$$

Fig. 3.7 shows the logarithmic spiral path, space around the flame, and the position considering at different values of t on the curve. It is clear from the figure that $t = -1$ shows the closest position to the flame, while $t = 1$ represents the farthest.

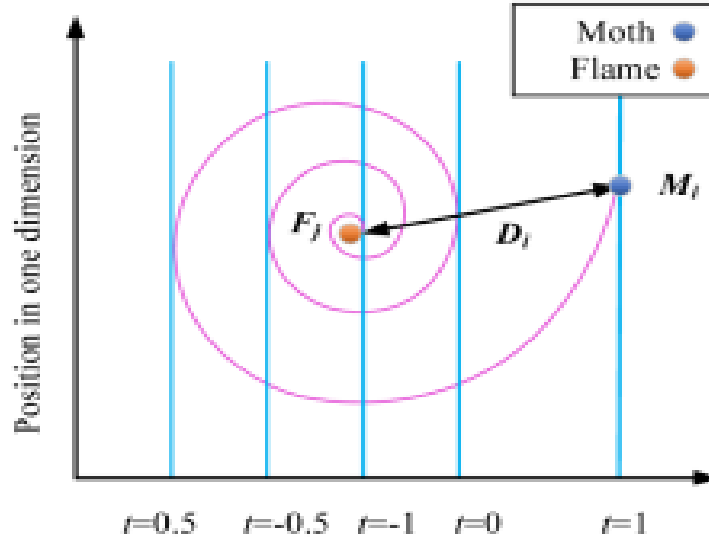


Fig. 3.7 Logarithmic spiral and space around a flame. (Source: Mirjalili, 2015)

According to Eq. 3.13, moths update their position to the local optimal position with respect to flames. After updating each moth corresponding to flame, the first and last positions of the moth are allotted to the best fitness flame and the worst flame, respectively.

At the beginning of the algorithm, there was N number of flames. During the iterations, the number of flames was then decreased progressively as per Eq. 3.15 to balance exploitation and exploration in the search space (Mirjalili, 2015).

$$C_k = \text{round}\left(N - k * \frac{N-1}{K}\right) \quad (3.15)$$

where k is the current iteration, N signifies the maximum number of flames, and K is the maximum number of iterations. The culmination of the iterations leads to the positioning of moths with respect to the best flame.

MFO flow chart in Fig. 3.8 conveys more understanding of the detailed computational process

and mechanism of the algorithm.

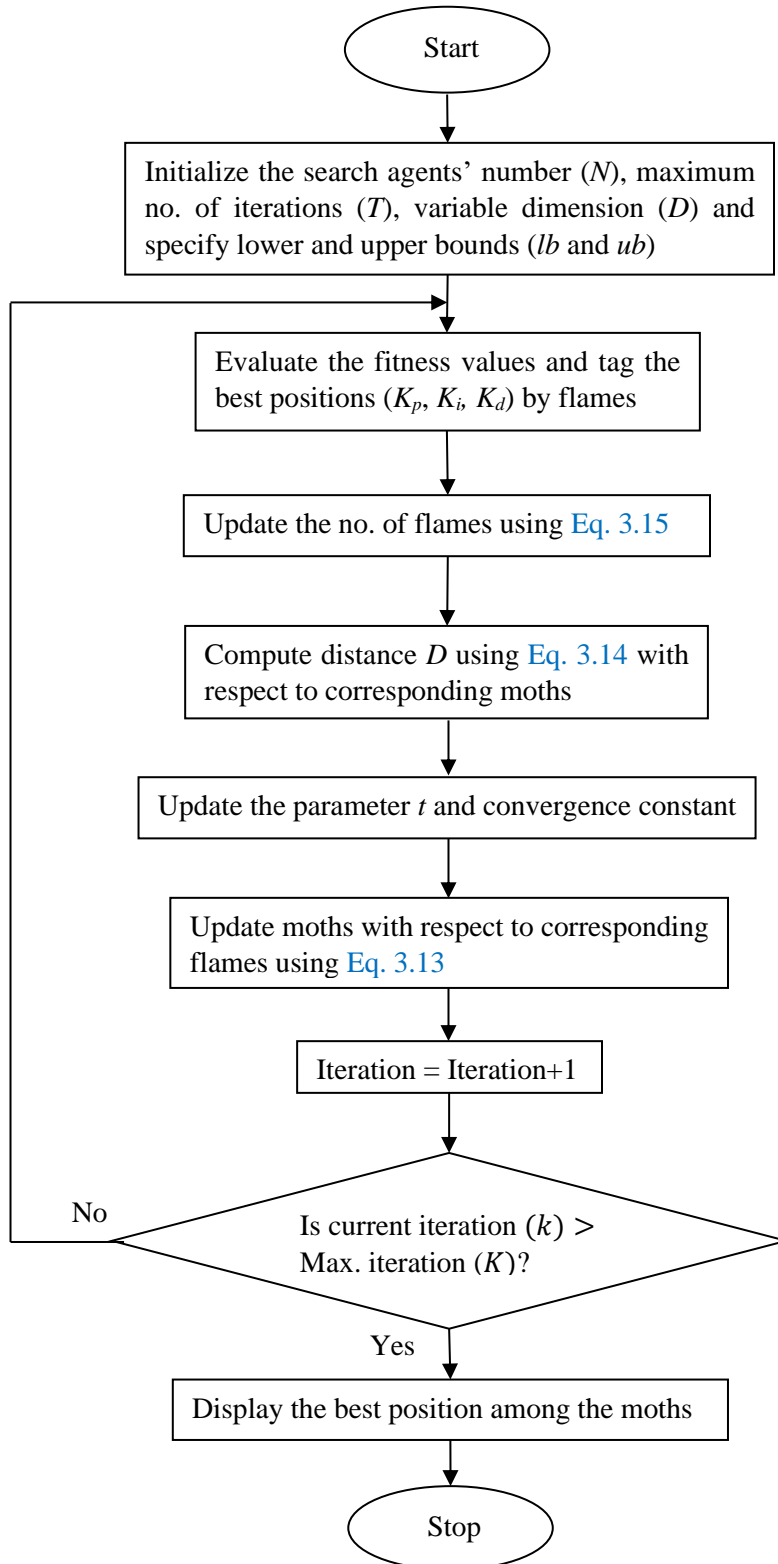


Fig. 3.8 Flowchart of MFO algorithm.

The MFO algorithm considered the same parameters as mentioned in [Table 3.1](#).

Initially, the controller's gain parameters were tuned using the Z-N classical method as discussed in Chapter 2. It was observed from Chapter 2 that in all the control schemes (conventional PID controller, standard SR-PID controller, and variable range SR-PID controller), the best set of the gain parameters was obtained for Case 3. Therefore, in view of this, in this chapter, the variable range SR-PID controller gain parameters were tuned for Case 3 using PSO, WOA, and MFO algorithms. The values of optimal gain parameters for PSO were observed as $K_p = -39.56$, $K_i = -8.472 \times 10^{-3}$, and $K_d = -3.98$, for WOA, $K_p = -38.11$, $K_i = -2.214 \times 10^{-3}$, and $K_d = -4.38$, and for MFO, $K_p = -37.72$, $K_i = -5.7128 \times 10^{-4}$, and $K_d = -4.69$. These gain values were then used to obtain the simulation results for increasing temperature setpoints (with the same signs) and decreasing temperature setpoints (with opposite signs).

To find the fitness value of the search agents, in this work, the objective function (fitness function) used in the above-said algorithms is as follow:

$$J = w_1 * T_s + (1 - w_1) * E_{ss} \quad (3.16)$$

where w_1 is the weighing factor, T_s and E_{ss} are setting time and steady-state error, respectively. The weighting factor for each parameter is taken as 0.5 (taken equal weightage for each parameter).

3.4 Comparison of the nature-inspired algorithms for tuning the controller

The performance of the controller using PSO, WOA, and MFO methods was tested for various temperature setpoints within the working range (22.5°C to 32.5°C).

The system was now investigated using the set of gain parameters obtained using nature-inspired algorithms mentioned above, and the Z-N method (in Chapter 2). The simulation results were obtained for the various temperature setpoints. Based on these simulation results, a comparative study was made for the controller performance using the Z-N method, PSO, WOA, and MFO algorithms on the basis of settling time, i.e., T_s (second), as shown in [Table 3.2](#).

Table 3.2 Comparative study of the transient response of SR-PID controller using classical Z-N and nature-inspired techniques with no dead time.

Cases	Setpoints (°C)	Error	Valve (with no dead time)				Improvement (%)		
			Z-N	PSO	WOA	MFO	PSO over Z-N	WOA over Z-N	MFO over Z-N
1	22.5	+3	369	197	180	176	46.61	51.22	52.30
2	22.5	+1	116	87	77	75	25.01	33.62	35.34
3	25	+3	565	278	254	245	50.80	55.04	56.64
4	25	+1	166	99	89	85	40.36	45.39	47.49
5	25	-1	628	251	218	206	60.03	65.29	67.21
6	25	-2	4496	1004	941	907	77.67	79.07	79.83
7	27.5	+3	1100	462	426	419	58.01	61.27	61.91
8	27.5	+1	265	155	143	138	41.51	46.04	47.92
9	27.5	-1	267	157	144	139	41.20	46.07	47.94
10	27.5	-3	1107	465	428	419	57.99	61.34	62.15
11	29	+3	3563	909	849	818	74.49	76.17	77.04
12	29	+1	400	177	163	156	55.75	59.25	61.01
13	29	-1	196	109	97	93	40.39	45.51	46.55
14	29	-3	957	412	391	382	56.95	59.14	60.08
15	32.5	-1	101	85	75	73	15.84	25.74	27.72
16	32.5	-3	364	194	175	170	46.70	51.92	53.30

It can be inferred from [Table 3.2](#) that simulation results obtained by optimized gains of SR-PID controller using the MFO algorithm were better as compared to those tuned using Z-N, PSO, and WOA methods in terms of settling time. This is due to the optimized controller gains achieved through better global convergence using the MFO algorithm. In nature-inspired algorithms such as PSO, WOA, and MFO, the optimal gains are obtained by optimizing the settling time and steady-state error of the system observed during simulation of the performance of the system, whereas the Z-N tuning method makes use of rules based on certain approximations. Moreover, all the search agents in the MFO algorithm do not search around a single promising solution as this would increase the probability of the local optima stagnation. Instead, each search agent searches the solution space around only one of the recent promising solutions assigned to it from a set of the continuously updated set of promising solutions. This results in a higher exploration of the considered solution space. The spiral path of movement of the search agent around the promising solution assigned to it leads to better exploitation of the search space around that promising solution. Further, the MFO algorithm implemented in this work gradually decrements the number of promising solutions in the set so that an increasing

number of search agents search the space around the more promising solutions as the iterations progress. This leads to better global convergence.

Further, the investigations were carried out with respect to the effect of dead time in the valve, the effect of disturbance into the process, and utility consumption.

3.4.1 Effect of dead time in the valve

Initially, the performance of the system was checked using PSO, WOA, and MFO algorithms with the valve with zero dead time (Table 3.2). Further, it was investigated with the effect of dead time in the valve by considering the dead time of 0.25 sec. and 0.5 sec. A comparative study of the transient response (in terms of T_s) of SR-PID controller for various temperature set points using Z-N method, PSO, WOA, and MFO algorithms with the dead time of 0.25 sec. and 0.5 sec. is shown in Tables 3.3–3.4, respectively.

Table 3.3 Performances of SR-PID controller using classical Z-N and nature-inspired techniques with a dead time of 0.25 sec.

Cases	Setpoints (°C)	Error	Valve (with dead time of 0.25 second)				Improvement (%)		
			Z-N	PSO	WOA	MFO	PSO over Z-N	WOA over Z-N	MFO over Z-N
1	22.5	+3	441	223	204	199	49.43	53.74	54.88
2	22.5	+1	136	102	90	87	26.15	33.82	36.03
3	25	+3	662	320	293	282	51.66	55.74	57.40
4	25	+1	196	116	104	99	40.82	45.94	47.49
5	25	-1	736	304	268	254	58.70	63.59	65.49
6	25	-2	5093	1215	1148	1110	76.14	77.46	78.21
7	27.5	+3	1311	560	521	511	57.28	60.26	61.02
8	27.5	+1	307	178	164	157	42.02	46.58	48.86
9	27.5	-1	310	180	165	158	41.94	46.77	49.03
10	27.5	-3	1321	565	524	512	57.23	60.33	61.24
11	29	+3	4250	1117	1051	1016	73.72	75.27	76.09
12	29	+1	468	196	180	171	58.12	61.54	63.46
13	29	-1	228	126	112	107	40.74	44.88	48.07
14	29	-3	1126	496	472	461	55.95	58.08	59.06
15	32.5	-1	122	101	89	86	17.21	27.05	29.51
16	32.5	-3	435	226	204	198	48.05	53.10	54.48

Based on the simulation results, it can be seen from Tables 3.2–3.4 that the variable range SR-PID controller using the MFO algorithm provided better results as compared to the controller using Z-N, PSO, and WOA methods in terms of settling time.

Table 3.4 Performances of SR-PID controller using classical Z-N and nature-inspired techniques with a dead time of 0.5 sec.

Cases	Setpoints (°C)	Error	Valve (with dead time of 0.5 second)				Improvement (%)		
			Z-N	PSO	WOA	MFO	PSO over Z-N	WOA over Z-N	MFO over Z-N
1	22.5	+3	532	270	250	244	49.25	53.01	54.14
2	22.5	+1	164	113	101	98	31.10	38.41	40.24
3	25	+3	799	385	357	345	51.81	55.32	56.82
4	25	+1	228	126	114	108	44.74	48.03	50.63
5	25	-1	875	368	330	314	57.94	62.29	64.11
6	25	-2	5836	1469	1398	1358	74.83	76.05	76.73
7	27.5	+3	1586	682	640	629	56.99	59.65	60.33
8	27.5	+1	362	199	184	176	45.03	49.17	51.38
9	27.5	-1	367	202	186	180	44.96	49.32	50.95
10	27.5	-3	1595	687	645	632	56.93	59.56	60.38
11	29	+3	5117	1368	1299	1262	73.27	74.61	75.34
12	29	+1	555	240	222	212	56.76	60.02	61.80
13	29	-1	267	142	128	122	43.82	48.06	49.31
14	29	-3	1320	594	568	556	55.01	56.97	57.88
15	32.5	-1	149	109	97	94	26.85	34.90	36.91
16	32.5	-3	527	271	249	243	48.58	52.75	53.89

It is also clear from the above tables that MFO over Z-N has more improvement than other used nature-inspired algorithms over Z-N. The presence of dead time in the valve causes the manipulated variables to take longer to change. As a result, the system will respond slowly, resulting in longer settling times in all cases as compared to a system with no dead time. It can also be inferred that the settling times increase with an increase in dead time of the valve.

3.4.2 Effect of disturbance in the process

The performances of the controllers using the Z-N method, PSO, WOA, and MFO algorithms for tuning were investigated in terms of T_s for the same temperature setpoints. A comparative analysis of the transient response of the SR-PID controller for the various temperature setpoints using the above-said methods, considering the effect of disturbance (considered as impulse function in the forward path in the process at $t=30$ seconds) is shown in [Table 3.5](#). The table shows that even in the presence of disturbance, MFO outperforms not only the Z-N method but also the PSO and WOA.

Table 3.5 Performances of SR-PID controller using classical Z-N and nature-inspired techniques with disturbances.

Cases	Setpoints (°C)	Error	Z-N	PSO	WOA	MFO	Improvement (%)		
							PSO over Z-N	WOA over Z-N	MFO over Z-N
1	22.5	+3	387	212	194	189	45.22	49.87	51.16
2	22.5	+1	126	99	88	85	21.43	30.16	32.54
3	25	+3	579	297	272	262	48.70	53.02	54.75
4	25	+1	189	113	102	97	40.21	46.03	48.68
5	25	-1	791	306	273	259	61.31	65.49	67.26
6	25	-2	5705	1077	1012	976	81.12	82.26	82.89
7	27.5	+3	1157	492	454	445	57.47	60.76	61.54
8	27.5	+1	322	177	164	159	45.03	49.07	50.62
9	27.5	-1	324	178	165	160	45.06	49.07	50.62
10	27.5	-3	1163	493	455	442	57.61	60.88	61.99
11	29	+3	4848	961	898	864	80.18	81.48	82.18
12	29	+1	501	216	201	192	56.89	59.88	61.68
13	29	-1	228	123	110	106	44.05	48.75	49.51
14	29	-3	975	438	415	405	55.08	57.44	58.46
15	32.5	-1	118	96	85	82	18.64	27.97	30.51
16	32.5	-3	377	208	187	181	44.83	50.40	51.99

It can be said from [Table 3.5](#) that when a disturbance is introduced into the process, the performance of the control system will be influenced in terms of settling time. In the presence of disturbance, the system will take longer to reach the required temperature as compared to the system in the absence of a disturbance.

3.4.3 Utility consumption

The performance of the controller was also investigated on the basis of utility consumption, i.e., cold utility and hot utility, considering the valve with zero dead time. As the simulation results, it was found that the utility consumption of cold water and hot water using a controller with MFO was lesser than that of Z-N, PSO, and WOA methods. The utility consumptions obtained using the SR-PID controller with Z-N, PSO, WOA, and MFO are presented in [Tables 3.6–3.9](#).

Table 3.6 Comparative study of the amount of Q_1 and Q_2 flows using SR-PID controller with Z-N and nature-inspired optimization techniques in case of valve with no dead time.

Cases	Setpoints (°C)	Error	Z-N		PSO		WOA		MFO	
			u_c	u_h	u_c	u_h	u_c	u_h	u_c	u_h
1	22.5	+3	8.27	25.28	7.02	18.69	5.82	15.26	5.11	13.95
2	22.5	+1	7.18	11.26	6.21	10.98	5.13	8.97	4.25	8.20
3	25	+3	10.09	40.8	8.04	26.54	6.66	21.67	5.85	19.81
4	25	+1	9.02	14.46	7.16	12.25	5.93	10.36	5.21	9.65
5	25	-1	46.41	12.14	25.35	9.68	21.02	7.63	18.45	7.24
6	25	-2	296.4	68.68	95.50	18.63	79.17	15.21	69.49	13.91
7	27.5	+3	17.03	82.38	10.14	46.17	8.40	37.69	7.38	34.46
8	27.5	+1	10.11	22.13	7.27	16.86	6.02	13.77	5.29	12.59
9	27.5	-1	21.39	11.17	18.66	8.12	15.47	6.62	13.58	6.06
10	27.5	-3	82.89	17.41	45.71	9.56	37.89	7.81	33.26	7.14
11	29	+3	54.36	208.62	17.29	84.16	14.33	68.71	12.58	62.81
12	29	+1	12.57	25.75	9.41	18.43	7.65	15.23	6.21	14.01
13	29	-1	15.47	9.88	12.26	7.55	10.51	6.17	9.47	5.64
14	29	-3	45.45	15.17	40.53	7.94	33.60	6.48	29.49	5.93
15	32.5	-1	10.95	7.03	10.51	6.12	8.53	5.06	7.97	4.14
16	32.5	-3	25.21	8.07	18.66	6.59	15.47	5.38	13.58	4.92

Table 3.7 Comparative study of the amount of Q_1 and Q_2 flows using SR-PID controller with Z-N and nature-inspired optimization techniques in case of valve with a dead time of 0.25 sec.

Cases	Setpoints (°C)	Error	Z-N		PSO		WOA		MFO	
			u_c	u_h	u_c	u_h	u_c	u_h	u_c	u_h
1	22.5	+3	8.73	31.76	7.63	21.48	6.27	17.39	5.45	15.76
2	22.5	+1	7.88	13.28	6.75	12.62	5.52	10.22	4.53	9.26
3	25	+3	11.07	48.12	8.74	30.51	7.17	24.69	6.24	22.37
4	25	+1	9.89	17.05	7.78	14.08	6.39	11.8	5.56	10.9
5	25	-1	57.13	14.19	27.55	11.13	22.64	8.69	19.69	8.18
6	25	-2	325.13	81	103.8	21.41	85.26	17.33	74.14	15.71
7	27.5	+3	20.49	98.21	11.02	53.07	9.05	42.95	7.87	38.92
8	27.5	+1	11.09	26.1	7.9	19.38	6.48	15.69	5.64	14.22
9	27.5	-1	23.46	13.17	20.28	9.33	16.66	7.54	14.49	6.84
10	27.5	-3	98.84	20.63	49.68	10.99	40.81	8.9	35.49	8.06
11	29	+3	59.63	246.05	18.79	96.74	15.43	78.29	13.42	70.94
12	29	+1	11.97	29.37	10.23	21.18	8.24	17.35	6.63	15.82
13	29	-1	16.97	11.65	13.33	8.68	11.32	7.03	10.1	6.37
14	29	-3	49.86	17.89	44.05	9.13	36.19	7.38	31.46	6.7
15	32.5	-1	12.01	7.56	11.42	7.03	9.19	5.77	8.5	4.68
16	32.5	-3	31.17	8.38	20.28	7.57	16.66	6.13	14.49	5.56

Table 3.8 Comparative study of the amount of Q_1 and Q_2 flows using SR-PID controller with Z-N and nature-inspired optimization techniques in case of valve with a dead time of 0.5 sec.

Cases	Setpoints (°C)	Error	Z-N		PSO		WOA		MFO	
			u_c	u_h	u_c	u_h	u_c	u_h	u_c	u_h
1	22.5	+3	10.67	37.67	8.45	25.12	6.82	19.99	5.98	18.27
2	22.5	+1	9.29	15.8	7.47	14.76	6	11.75	4.97	10.74
3	25	+3	13.06	57.24	9.67	35.68	7.79	28.38	6.84	25.94
4	25	+1	11.66	20.28	8.61	16.47	6.95	13.56	6.1	12.64
5	25	-1	73.44	16.19	30.5	13.02	24.61	9.99	21.6	9.48
6	25	-2	383.47	96.34	114.9	25.04	92.67	19.92	81.33	18.21
7	27.5	+3	24.66	119.05	12.2	62.06	9.84	49.37	8.63	45.13
8	27.5	+1	13.08	31.04	8.74	22.66	7.04	18.03	6.19	16.49
9	27.5	-1	27.67	15.66	22.45	10.91	18.11	8.67	15.89	7.93
10	27.5	-3	118.69	25.37	54.99	12.85	44.36	10.23	38.93	9.35
11	29	+3	70.33	292.66	20.8	113.13	16.77	89.99	14.72	82.25
12	29	+1	13.28	38.74	11.32	24.77	8.96	19.94	7.27	18.34
13	29	-1	20.02	13.86	14.76	10.15	12.3	8.08	11.08	7.39
14	29	-3	58.81	21.28	48.76	10.68	39.34	8.48	34.51	7.77
15	32.5	-1	14.17	8.99	12.64	8.22	9.99	6.63	9.32	5.43
16	32.5	-3	37.59	10.30	22.45	8.85	18.11	7.05	15.89	6.45

Table 3.9 Comparative study of the amount of Q_1 and Q_2 flows using SR-PID controller with Z-N and nature-inspired optimization techniques in case of valve with disturbance and no dead time.

Cases	Setpoints (°C)	Error	Z-N		PSO		WOA		MFO	
			u_c	u_h	u_c	u_h	u_c	u_h	u_c	u_h
1	22.5	+3	8.87	26.82	7.61	20.26	6.37	16.73	5.64	15.51
2	22.5	+1	7.68	11.94	6.73	11.9	5.61	9.83	4.69	9.12
3	25	+3	10.8	43.25	8.72	28.77	7.29	23.75	6.46	22.03
4	25	+1	9.65	15.33	7.76	13.28	6.49	11.36	5.75	10.73
5	25	-1	49.66	11.82	27.49	10.51	23.01	8.36	20.38	8.05
6	25	-2	327.15	72.8	106.43	20.22	89.02	16.67	80.92	15.47
7	27.5	+3	18.22	86.51	11.69	50.06	9.19	41.32	8.15	38.32
8	27.5	+1	10.82	23.46	7.88	18.28	6.59	15.09	5.84	14.03
9	27.5	-1	22.89	11.84	20.24	8.82	16.93	7.26	15.03	6.74
10	27.5	-3	87.98	18.87	49.57	10.36	41.47	8.56	36.73	7.94
11	29	+3	58.17	221.14	18.75	93.77	15.68	77.38	13.89	72.35
12	29	+1	12.98	27.32	10.24	19.98	8.37	16.7	6.86	15.58
13	29	-1	16.55	10.47	13.31	8.19	11.5	6.76	10.46	6.27
14	29	-3	48.63	16.08	43.95	8.61	36.77	7.1	32.57	6.59
15	32.5	-1	11.72	7.45	11.42	6.64	9.34	5.55	8.81	4.62
16	32.5	-3	26.21	8.58	20.24	7.14	16.93	5.9	15.18	5.47

It can be observed from the above [Tables 3.6–3.9](#) that the utility consumption of cold and hot water using SR-PID controller tuned with MFO algorithm was lesser as compared to the controller tuned with Z-N method, PSO, and WOA algorithms.

The scope for further improvement in the original MFO algorithm has also been explored. Various researchers have reported modifications in the original MFO algorithm (Li et al., 2021; Ma et al., 2021; Mohanty & Panda, 2021; Sapre & Mini, 2021; Shehab et al., 2021; Suja, 2021; Xia et al., 2021; Bandopadhyay & Roy, 2020; Zhiling Cui et al., 2020; Dash et al., 2020; Dash et al., 2020a; Elattar & Elsayed, 2020; Elaziz et al., 2020; Fei et al., 2020; Helmi & Alenany, 2020; Kaur et al., 2020; Korashy et al., 2020; Li Yu et al., 2020; Lin et al., 2020; Pelusi et al., 2020; Reddy & Bojja, 2020; Sayed et al., 2020; R. Sharma & Saha, 2020; Yu et al., 2020; Zhang Z. et al., 2020; Zhang H. et al., 2020; Zhao et al., 2020; Buch & Trivedi, 2019; Hongwei et al., 2019; Jain & Saxena, 2019; Jia et al., 2019; Khalilpourazari & Khalilpourazary, 2019; Li et al., 2019; Luo et al., 2019; Rashid et al., 2019; Sapre & Mini, 2019; Sheng et al., 2019; Singh et al., 2019; Taher et al., 2019; Wu et al., 2019; Xu Y. et al., 2019; Xu Y. et al., 2019a; Elsakaan et al., 2018; Jangir & Trivedi, 2018; Kamalpathi et al., 2018; Li C. et al., 2018; Li W. K. et al., 2018; Reddy et al., 2018; Sayed & Hassanien, 2018; Xu L. et al., 2018; Anfal & Abdelhafid, 2017; Aziz et al., 2017; Bhesdadiya et al., 2017; Gholizadeh et al., 2017; Hassanien et al., 2017; Jangir P., 2017; Savsani & Tawhid, 2017; M. Wang et al., 2017; W. Yang et al., 2017; Li et al., 2016; Soliman et al., 2016; Vikas & Nanda, 2016; Zhang *et al.*, 2016) and also shown in a table in **Appendix I**. Most of these modifications involve additional parameters and computational complexity. Hence, in the current work, it was decided to explore only such modifications that do not add parameters and complex computations to the existing algorithm.

The current work investigates the effect of the following three modifications and their combination in improving the exploitation and exploration capabilities along with improvement in convergence: (i) modifying the spiral path for improved exploitation of the search space, (ii) use of opposition theory to create the initial population consisting of search agents with better prospects as compared to the original MFO algorithm, (iii) change in the flame matrix to include better flames to improve exploration and convergence and (iv) combination of these three modifications. The effect of these modifications on the performance of the variable range SR-PID controller is studied.

3.5 Improvements in MFO

In light of the superior performance of the MFO algorithm for tuning the PID controller, some improvements in the MFO algorithm were proposed to further enhance the performance of the

MFO algorithm for tuning the PID controller. It was decided to explore the solution space more efficiently using a variable spiral path, opposition-based learning, and modification in the flame selection. The incorporation of the three proposed improvements in the original MFO algorithm resulted into three improved MFO algorithms namely, IMFO 1, IMFO 2, and IMFO 3, respectively. Further, the above three were combined to form the Enhanced MFO (EMFO) algorithm (Vishnoi *et al.*, 2021b).

3.5.1 Improved moth flame optimization 1 (IMFO 1) algorithm

In the original MFO algorithm, the logarithmic spiral path was used, as shown in Fig. 3.9 (a), in which moths move around the flames from infinite to origin with a varying pitch of the spiral path. This results in non-uniform exploitation of the search space around the flames. As a result, the final global solution may not be optimum. To further explore the search space properly, it was thought to change the spiral path. Sun *et al.* presented an improved whale optimization algorithm based on eight different search paths for optimizing 23 functions, where the Archimedean spiral path provided superior performance as compared to the other seven spiral paths used (Sun *et al.*, 2018). This motivated us to consider the Archimedean spiral path in the MFO algorithm, as described in Eq. 3.17.

$$S(M_i, F_j) = D_i \cdot t \cdot \cos(2\pi t) + F_j \quad (3.17)$$

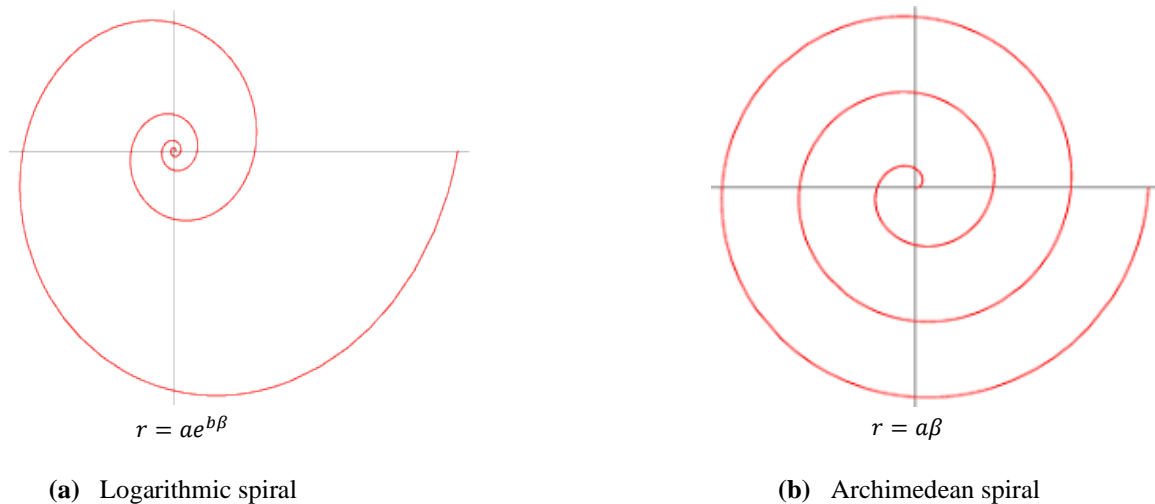


Fig. 3.9 Different spirals path. (Source: Sun *et al.*, 2018)

The Archimedean spiral path is shown in Fig. 3.9 (b), which initiates from its origin and generates a curve with a constant pitch, due to which the potential solutions may be found uniformly in the complete solution space leading to faster convergence toward a better optimum solution.

3.5.2 Improved moth flame optimization 2 (IMFO 2) algorithm

This section represents an improved version of the MFO algorithm based on the opposition learning theory. The concept of this theory, first introduced by Tizhoosh in 2005 (Tizhoosh, 2005), can be used to enhance the basic meta-heuristic methods to obtain the optimal solution to any problem of optimization. Generally, optimization algorithms start with a random initial population due to the lack of prior knowledge about the solution, which increases the chance of visiting unproductive regions of the search space. Hence, to overcome such inherent problems, the opposition learning based initialization has been reported in the literature (Choi *et al.*, 2019; Dinkar & Deep, 2018, 2019; Gaidhane & Nigam, 2018; Gupta & Deep, 2019; Rahnamayan *et al.*, 2008; Sapre & Mini, 2019; Wang, 2015). Mathematically, it has been proved that opposite numbers are more likely to be closer to the optimal solution than random ones (Rahnamayan *et al.*, 2008a). This motivated us to employ the opposition learning theory in the MFO algorithm for boosting the performance of the algorithm.

The following steps show the procedure to select the initial population based on the opposite population.

Step-1 Initialize the random population P_{ij} , $i = 1, 2, \dots, N$; $j = 1, 2, 3$.

Step-2 Evaluate its opposite population Q_{ij} based on the opposition learning theory:

$$Q_{ij} = ub_i + lb_i - P_{ij} \quad (3.18)$$

Step-3 Pick the N fittest candidates based on the fitness values from the set of these two populations, i.e., $(P_{ij} \cup Q_{ij})$.

By doing this, N fittest candidates (random or its opposite solution) can be selected as a starting population that leads to faster convergence towards the best solution using the Archimedean spiral path.

3.5.3 Improved moth flame optimization 3 (IMFO 3) algorithm

In the original MFO algorithm, flames are updated by sorting the updated moths, as discussed in section 3.3. This method, however, suffers from the inherent problem of non-exploration of complete solution space (Kaur *et al.*, 2020; Pelusi *et al.*, 2020; Reddy *et al.*, 2018; Sheng *et al.*, 2019). This motivated us to devise a new search strategy that explores the solution space fully, probably leading to a better optimal solution. This strategy proposes to update the current flame matrix by introducing better new flames from the solution space. The method of selection of the new better flames from the solution space is random in nature. The steps for the proposed strategy for k^{th} iterations are as follows:

- Step-1** Select the random flames S_{ij} within the boundary, $i = 1, 2, \dots, N; j = 1, 2, 3$.
- Step-2** Merge these random flames S_{ij} with the current flames F_{ij} as evaluated initially in the original MFO algorithm.
- Step-3** Select the N fittest flames FF_{ij} based on their fitness value from a set of two ($S_{ij} \cup F_{ij}$).
- Step-4** Update the position of the moths M_{ij} in respect of corresponding N fittest flames FF_{ij} .
- Step-5** Update the remaining last $(N - C_k)$ moths' position of M_{ij} with respect to the fittest flame position FF_{1j} .

In this algorithm, the moths are updated using the Archimedean spiral path.

3.5.4 Enhanced moth flame optimization (EMFO) algorithm

This approach proposed a new version of the MFO algorithm by combining all the above-said algorithms, namely IMFO 1, IMFO 2, and IMFO 3, to merge the advantages of all the above-mentioned algorithms. The steps for the proposed EMFO algorithm are as follows:

- Step-1** Initialize the same number of search agents (N), the variable dimension (D), the maximum number of iterations (K), and the upper and lower bounds (ub and lb), as mentioned in Table 3.1.
- Step-2** Generate N most suitable search agents from two populations ($P_{ij} \cup Q_{ij}$) on the basis of their fitness values, where P_{ij} is the random population within the range (bounds), $i = 1, 2, \dots, N; j = 1, 2, 3$, and Q_{ij} is the opposite population.

Step-3 Calculate the moths' fitness values as per Eq. 3.16 and identify the best positions as flames.

Step-4 Update flame number as per Eq. 3.19:

$$C_k = \text{round}(N - k * \frac{N-1}{K}) \quad (3.19)$$

where N indicates the maximum number of flames, k denotes the current iteration, and K signifies the maximum number of iterations.

Step-5 Find new flames FF_j as the N most suitable flames on the basis of their fitness values from $(S_{ij} \cup F_{ij})$, where S_{ij} is the set of random flames, and F_{ij} is the set of existing flames as calculated first in the basic MFO algorithm.

Step-6 Compute the distance D_i between the new flames FF_j and the corresponding moths M_i , i.e.

$$D_i = |FF_j - M_i| \quad (3.20)$$

Step-7 Update the convergence constant r and parameter t as per Eq. 3.21 and 3.22:

$$r = -1 + k * (\frac{-1}{K}) \quad (3.21)$$

$$t = (r - 1) * \text{rand} + 1 \quad (3.22)$$

Step-8 Change the moths' position with respect to corresponding new flames FF_j as per the following equation:

$$S(M_i, FF_j) = D_i \cdot t \cdot \cos(2\pi t) + FF_j \quad (3.23)$$

where S indicates the spiral function, M_i represents the i^{th} moths corresponding to j^{th} new flames FF_j , and t denotes a random number in the range from -1 to 1.

Step-9 Iteration = Iteration+1.

Step-10 If current iteration (k) > max. iteration (K), then show the best moth position among the others; else, go to Step 3 and repeat.

The parameters for all the improved MFO algorithms were considered the same, as shown in Table 3.1.

3.6 Results and discussion

The performance of the controllers tuned using the improved versions of the MFO algorithm (IMFO 1, IMFO 2, IMFO 3, and EMFO) for the cases under consideration were simulated and compared with the performance of the controller tuned using the original MFO algorithm in terms of settling time, as presented in Table 3.10.

Table 3.10 Comparative study of the transient response of SR-PID controller using MFO and improved MFO algorithms.

Cases	Setpoints (°C)	Error	MFO	IMFO 1	IMFO 2	IMFO 3	EMFO
1	22.5	+3	176	168	163	156	148
2	25	-1	206	194	179	176	160
3	27.5	+3	419	392	383	367	329
4	27.5	-3	419	396	386	371	335
5	29	+1	156	151	144	136	127
6	32.5	-3	170	166	160	155	147

It can be inferred from the simulation results in Table 3.10 that each of the proposed modifications has led to improvements in the controller performance as compared to the original MFO algorithm. A comparison is presented in Fig. 3.10.

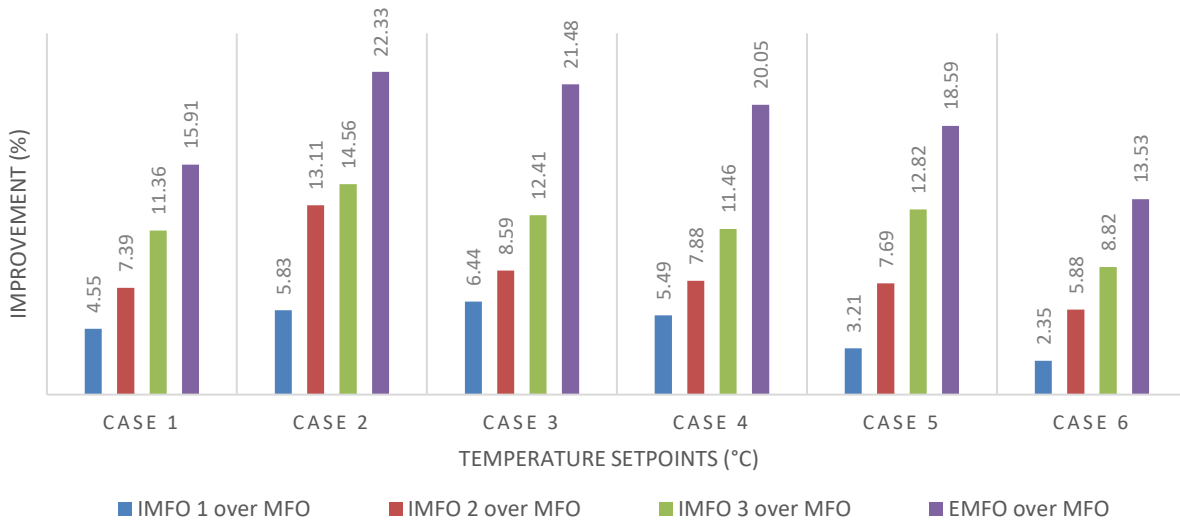


Fig. 3.10 Percentage improvement in the performances of SR-PID controller using MFO and improved MFO algorithms.

On the basis of simulation results in [Table 3.10](#) and the relative performance of the proposed algorithms shown in [Fig. 3.10](#), EMFO has been found to be the most efficient in improving the performance of the controller in terms of settling times.

It is observed that IMFO 1, where the spiral path was changed to ensure a uniform search in the solution space, has led to improved solutions indicating better exploitation of the solution space. The improved solutions (controller gains) have resulted in a considerable decrease in settling time. The improvement in the case of IMFO 2 is more than that of IMFO 1 as IMFO 2 combines the advantage of IMFO 1 and the benefit of using opposition-based learning for generating an optimal initial population. An optimal initial population converges to better solutions in fewer iterations along with a better spiral path in IMFO 2 has resulted in an improved search of the solution space. The original MFO algorithm exploits the solution space around probable fitter solutions in search of the global optima. In IMFO 3, a random search for even fitter probable solutions is made in each iteration to ensure a better exploration of the solution space. This improved exploration is combined with better exploitation due to the spiral path of IMFO 1 in IMFO 3, resulting in even better controller gains that further reduce the settling times. The last of the proposed algorithms (EMFO) demonstrates the best performance as it combines the advantages of the first three algorithms. It has an optimal initial population using a better spiral path to search around even fitter probable candidates for the global optima. This ensures that optimal controller gains are provided in case of the EMFO algorithm resulting in a significant reduction in the settling time for all the cases under consideration.

Further, the investigations were made to study the effect of dead time (considering a dead time of 0, 0.25, and 0.5 seconds in the valve) and process disturbance. A comparison was made for the transient performance of the SR-PID controller for various temperature setpoints using the original MFO and EMFO algorithms with and without dead time and disturbance as shown in [Tables 3.11–3.16](#).

Table 3.11 Performances of SR-PID controller using MFO and EMFO algorithms with no dead time.

Cases	Setpoints (°C)	Error	MFO	EMFO	Improvement (%)
					EMFO over MFO
1	22.5	+3	176	148	15.91
2	22.5	+1	75	46	38.67
3	25	+3	245	182	25.71
4	25	+1	85	53	37.65
5	25	-1	206	160	22.33
6	25	-2	907	545	39.91
7	27.5	+3	419	329	21.48
8	27.5	+1	138	103	25.36
9	27.5	-1	139	101	27.34
10	27.5	-3	419	335	20.05
11	29	+3	818	657	19.68
12	29	+1	156	127	18.59
13	29	-1	93	68	26.88
14	29	-3	382	217	43.19
15	32.5	-1	73	45	38.36
16	32.5	-3	170	147	13.53

Table 3.12 Performances of SR-PID controller using MFO and EMFO algorithms with a dead time of 0.25 sec.

Cases	Setpoints (°C)	Error	MFO	EMFO	Improvement (%)
					EMFO over MFO
1	22.5	+3	199	157	21.11
2	22.5	+1	87	52	40.23
3	25	+3	282	196	30.5
4	25	+1	99	60	39.39
5	25	-1	254	177	30.31
6	25	-2	1110	588	47.03
7	27.5	+3	511	351	31.31
8	27.5	+1	157	113	28.03
9	27.5	-1	158	112	29.11
10	27.5	-3	512	355	30.66
11	29	+3	1016	699	31.2
12	29	+1	171	139	18.71
13	29	-1	107	76	28.97
14	29	-3	461	230	50.11
15	32.5	-1	86	50	41.86
16	32.5	-3	198	155	21.72

Table 3.13 Performances of SR-PID controller using MFO and EMFO algorithms with a dead time of 0.5 sec.

Cases	Setpoints (°C)	Error	MFO	EMFO	Improvement (%)
					EMFO over MFO
1	22.5	+3	244	171	29.92
2	22.5	+1	98	60	38.71
3	25	+3	345	218	36.81
4	25	+1	108	69	36.11
5	25	-1	314	198	36.94
6	25	-2	1358	652	51.99
7	27.5	+3	629	379	39.75
8	27.5	+1	176	126	28.41
9	27.5	-1	180	125	30.56
10	27.5	-3	632	385	39.08
11	29	+3	1262	772	38.83
12	29	+1	212	155	26.89
13	29	-1	122	85	30.33
14	29	-3	556	248	55.4
15	32.5	-1	94	58	38.3
16	32.5	-3	243	167	31.28

Table 3.14 Performances of SR-PID controller using MFO and EMFO algorithms with disturbance and no dead time.

Cases	Setpoints (°C)	Error	MFO	EMFO	Improvement (%)
					EMFO over MFO
1	22.5	+3	189	155	17.99
2	22.5	+1	85	50	41.18
3	25	+3	262	191	27.1
4	25	+1	97	59	39.18
5	25	-1	259	178	31.27
6	25	-2	976	571	41.5
7	27.5	+3	445	342	23.15
8	27.5	+1	159	113	28.93
9	27.5	-1	160	112	30.01
10	27.5	-3	442	348	21.27
11	29	+3	864	681	21.18
12	29	+1	192	134	30.21
13	29	-1	106	76	28.3
14	29	-3	405	226	44.2
15	32.5	-1	82	48	41.46
16	32.5	-3	181	152	16.02

Table 3.15 Performances of SR-PID controller using MFO and EMFO algorithms with a dead time of 0.25 sec. and disturbance.

Cases	Setpoints (°C)	Error	MFO	EMFO	Improvement (%)
					EMFO over MFO
1	22.5	+3	213	166	22.07
2	22.5	+1	98	57	41.84
3	25	+3	301	208	30.9
4	25	+1	112	67	40.18
5	25	-1	308	196	36.36
6	25	-2	1180	615	47.88
7	27.5	+3	538	365	32.16
8	27.5	+1	179	124	30.73
9	27.5	-1	180	124	31.11
10	27.5	-3	536	369	31.16
11	29	+3	1063	724	31.89
12	29	+1	208	147	29.33
13	29	-1	121	85	29.75
14	29	-3	485	241	50.31
15	32.5	-1	96	55	42.71
16	32.5	-3	210	164	21.9

Table 3.16 Performances of SR-PID controller using MFO and EMFO algorithms with a dead time of 0.5 sec. and disturbance.

Cases	Setpoints (°C)	Error	MFO	EMFO	Improvement (%)
					EMFO over MFO
1	22.5	+3	260	182	30.38
2	22.5	+1	111	68	38.74
3	25	+3	365	234	36.99
4	25	+1	123	78	37.4
5	25	-1	370	227	38.65
6	25	-2	1430	697	52.52
7	27.5	+3	658	403	39.97
8	27.5	+1	200	138	31.01
9	27.5	-1	204	138	32.35
10	27.5	-3	658	407	39.36
11	29	+3	1311	816	38.98
12	29	+1	251	169	32.67
13	29	-1	138	95	31.16
14	29	-3	582	263	55.5
15	32.5	-1	106	66	38.68
16	32.5	-3	257	176	31.52

Based on the simulation results, it can be said from [Tables 3.11–3.16](#) that in all the scenarios, the SR-PID controller using EMFO algorithm outperforms the controller with the original MFO algorithm.

The performance of the controller is also studied in terms of utility consumption. The same is carried out for the above-considered setpoints. The consumption of utility using the controller with MFO and EMFO is presented in Tables 3.17–3.22.

Table 3.17 Comparative study of the amount of Q_1 and Q_2 flows using SR-PID controller with MFO and EMFO in case of valve with no dead time.

Cases	Setpoints (°C)	Error	MFO		EMFO	
			u_c	u_h	u_c	u_h
1	22.5	+3	5.11	13.95	4.51	11.48
2	22.5	+1	4.25	8.20	4.18	5.83
3	25	+3	5.85	19.81	5.43	16.42
4	25	+1	5.21	9.65	4.59	6.94
5	25	-1	18.45	7.24	15.53	5.39
6	25	-2	69.49	13.91	45.42	9.06
7	27.5	+3	7.38	34.46	5.98	27.21
8	27.5	+1	5.29	12.59	4.84	9.75
9	27.5	-1	13.58	6.06	9.62	4.81
10	27.5	-3	33.26	7.14	27.08	5.93
11	29	+3	12.58	62.81	11.07	54.36
12	29	+1	6.21	14.01	5.97	11.3
13	29	-1	9.47	5.64	6.19	4.36
14	29	-3	29.49	5.93	18.97	4.55
15	32.5	-1	7.97	4.14	5.69	4.11
16	32.5	-3	13.58	4.92	11.33	4.38

Table 3.18 Comparative study of the amount of Q_1 and Q_2 flows using SR-PID controller with MFO and EMFO in case of valve with a dead time of 0.25 sec.

Cases	Setpoints (°C)	Error	MFO		EMFO	
			u_c	u_h	u_c	u_h
1	22.5	+3	5.45	15.76	4.8	12.9
2	22.5	+1	4.53	9.26	4.45	6.55
3	25	+3	6.24	22.37	5.78	18.45
4	25	+1	5.56	10.9	4.89	7.8
5	25	-1	19.69	8.18	16.54	6.06
6	25	-2	74.14	15.71	48.36	10.18
7	27.5	+3	7.87	38.92	6.37	30.57
8	27.5	+1	5.64	14.22	5.15	10.95
9	27.5	-1	14.49	6.84	10.24	5.41
10	27.5	-3	35.49	8.06	28.83	6.66
11	29	+3	13.42	70.94	11.79	61.07
12	29	+1	6.63	15.82	6.36	12.7
13	29	-1	10.1	6.37	6.59	4.91
14	29	-3	31.46	6.7	20.2	5.11
15	32.5	-1	8.5	4.68	6.06	4.61
16	32.5	-3	14.49	5.56	12.06	4.92

Table 3.19 Comparative study of the amount of Q_1 and Q_2 flows using SR-PID controller with MFO and EMFO in case of valve with a dead time of 0.5 sec.

Cases	Setpoints (°C)	Error	MFO		EMFO	
			u_c	u_h	u_c	u_h
1	22.5	+3	5.98	18.27	5.25	14.83
2	22.5	+1	4.97	10.74	4.87	7.53
3	25	+3	6.84	25.94	6.33	21.21
4	25	+1	6.1	12.64	5.35	8.97
5	25	-1	21.6	9.48	18.11	6.97
6	25	-2	81.33	18.21	52.94	11.7
7	27.5	+3	8.63	45.13	6.97	35.14
8	27.5	+1	6.19	16.49	5.64	12.59
9	27.5	-1	15.89	7.93	11.21	6.21
10	27.5	-3	38.93	9.35	31.56	7.66
11	29	+3	14.72	82.25	12.91	70.2
12	29	+1	7.27	18.34	6.96	14.6
13	29	-1	11.08	7.39	7.21	5.63
14	29	-3	34.51	7.77	22.11	5.87
15	32.5	-1	9.32	5.43	6.63	5.31
16	32.5	-3	15.89	6.45	13.2	5.66

Table 3.20 Comparative study of the amount of Q_1 and Q_2 flows using SR-PID controller with MFO and EMFO in case of disturbance.

Cases	Setpoints (°C)	Error	MFO		EMFO	
			u_c	u_h	u_c	u_h
1	22.5	+3	5.64	15.51	4.93	12.68
2	22.5	+1	4.69	9.12	4.57	6.44
3	25	+3	6.46	22.03	5.94	18.14
4	25	+1	5.75	10.73	5.02	7.67
5	25	-1	20.38	8.05	16.99	5.96
6	25	-2	80.92	15.47	49.68	10.01
7	27.5	+3	8.15	38.32	6.54	30.06
8	27.5	+1	5.84	14.03	5.29	10.77
9	27.5	-1	15.03	6.74	10.52	5.31
10	27.5	-3	36.73	7.94	29.62	6.55
11	29	+3	13.89	72.35	12.11	60.06
12	29	+1	6.86	15.58	6.53	12.48
13	29	-1	10.46	6.27	6.77	4.82
14	29	-3	32.57	6.59	20.75	5.03
15	32.5	-1	8.81	4.62	6.22	4.53
16	32.5	-3	15.18	5.47	12.39	4.84

Table 3.21 Comparative study of the amount of Q_1 and Q_2 flows using SR-PID controller with MFO and EMFO in case of disturbance and valve with a dead time of 0.25 sec.

Cases	Setpoints (°C)	Error	MFO		EMFO	
			u_c	u_h	u_c	u_h
1	22.5	+3	6.07	17.6	5.28	14.34
2	22.5	+1	5.05	10.34	4.89	7.28
3	25	+3	6.95	24.99	6.36	20.51
4	25	+1	6.2	12.17	5.38	8.67
5	25	-1	21.94	9.14	18.19	6.74
6	25	-2	82.62	17.55	53.18	11.31
7	27.5	+3	8.77	43.47	7.01	33.98
8	27.5	+1	6.29	15.88	5.66	12.17
9	27.5	-1	16.15	7.64	11.26	6.32
10	27.5	-3	39.55	9.03	31.71	7.41
11	29	+3	14.96	79.24	12.97	67.88
12	29	+1	7.39	17.67	6.99	14.12
13	29	-1	11.26	7.11	7.25	5.45
14	29	-3	35.06	7.48	22.21	5.68
15	32.5	-1	9.47	5.23	6.66	5.12
16	32.5	-3	16.15	6.21	13.26	5.47

Table 3.22 Comparative study of the amount of Q_1 and Q_2 flows using SR-PID controller with MFO and EMFO in case of disturbance and valve with a dead time of 0.5 sec.

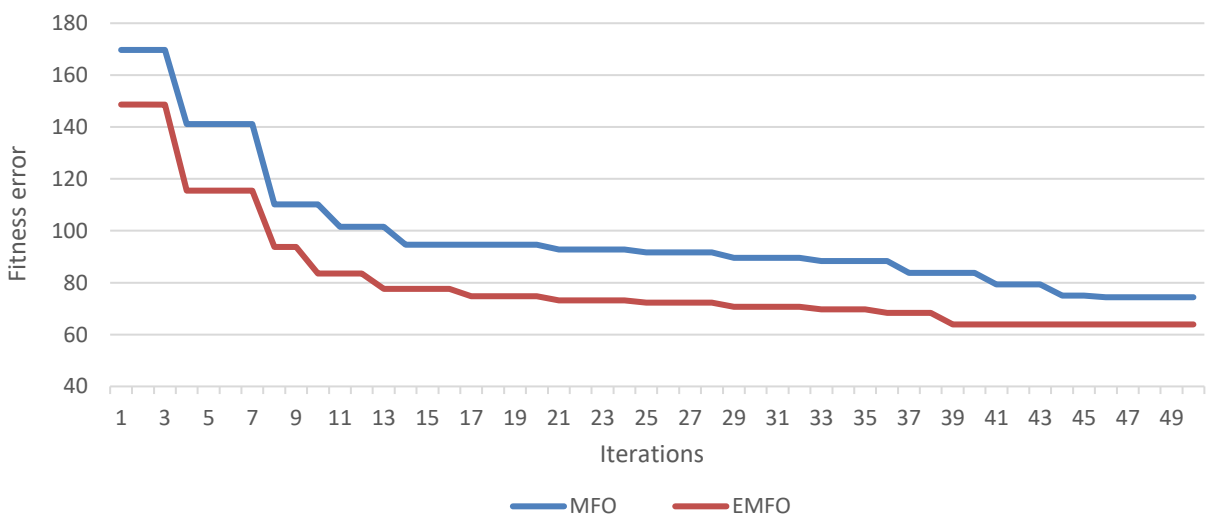
Cases	Setpoints (°C)	Error	MFO		EMFO	
			u_c	u_h	u_c	u_h
1	22.5	+3	6.72	20.48	5.84	16.05
2	22.5	+1	5.59	12.04	5.41	8.15
3	25	+3	7.69	29.08	7.03	22.96
4	25	+1	6.86	14.17	5.95	9.71
5	25	-1	24.29	10.63	20.12	7.55
6	25	-2	91.45	20.41	58.82	12.66
7	27.5	+3	9.7	50.59	7.75	38.04
8	27.5	+1	6.96	18.48	6.26	13.62
9	27.5	-1	17.87	8.89	12.45	6.72
10	27.5	-3	43.77	10.48	35.07	8.28
11	29	+3	16.55	92.2	14.34	75.99
12	29	+1	8.17	20.56	7.73	15.81
13	29	-1	12.46	8.28	8.02	6.11
14	29	-3	38.8	8.71	24.56	6.36
15	32.5	-1	10.48	6.09	7.37	5.73
16	32.5	-3	17.87	7.23	14.67	6.12

It was found from [Tables 3.17–3.22](#) that the utility consumption of cold and hot water (u_c and u_h) using the controller with the EMFO algorithm was lower when compared to the original MFO

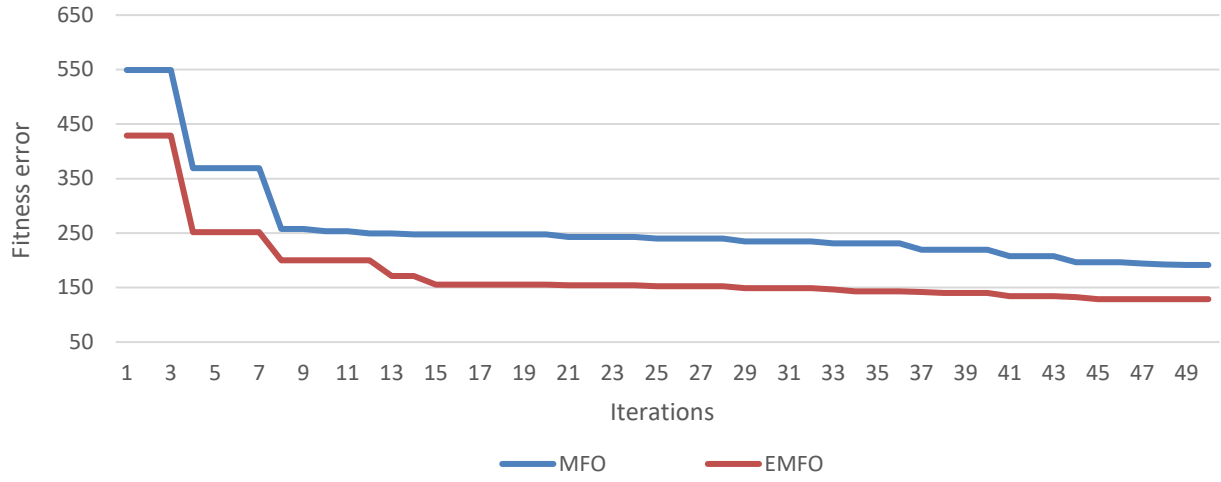
algorithm. The simulation results also show that hot utility consumption increases for the same desired rise as we move upward in the working range of the temperature. This is so because at higher temperatures, the difference between the temperature of the hot water inflowing and the tank temperature reduces. As a result, the amount of hot water flowing into the tank to bring about the same desired rise in the temperature increases. For similar reason, the cold water consumption increases for the same desired fall in the temperature as we move downward in the working range of temperature.

It can also be said from [Tables 3.17–3.19](#) that the hot and cold utility consumptions rise as the valve's dead time increases. These utility consumptions (u_h and u_c) rise even more when disturbances and dead times are introduced ([Tables 3.21–3.22](#)).

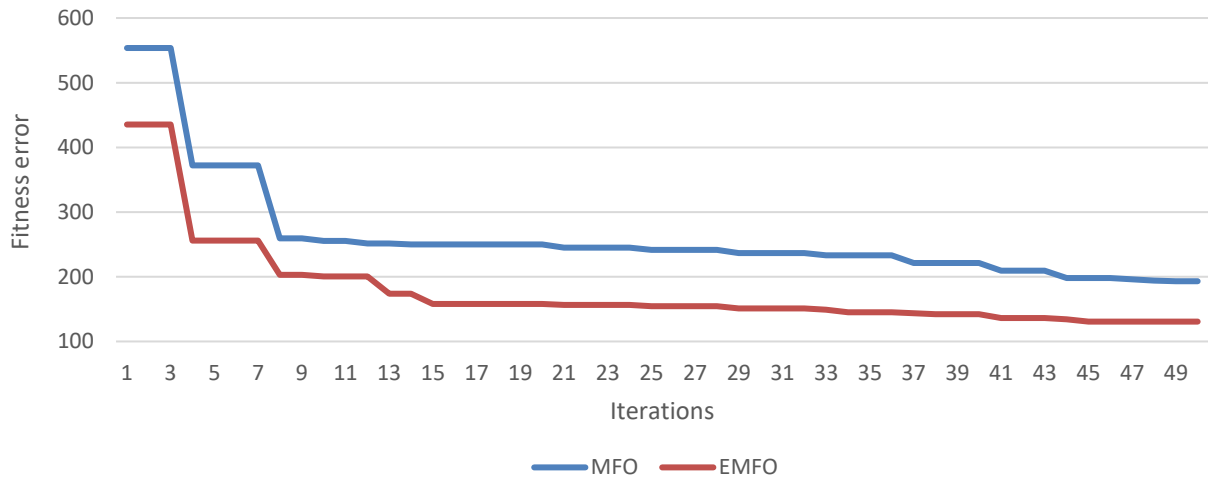
Moreover, a comparison between the original MFO and EMFO algorithms was made with regard to the values of fitness function and its convergence as the iterations progressed. It was observed that the value of the fitness function in the very first iteration was less in the case of the EMFO algorithm as compared to the MFO algorithm. It was also found that the fitness function converged faster to a comparatively smaller value in the case of the EMFO algorithm as compared to the original MFO algorithm for all the considered set points. [Fig. 3.11](#) shows the fitness function's convergence with iterations proceeded for Cases 5, 7, 10, and 12 listed in [Table 3.11](#) (cases without dead time). Similar trends were observed in all other cases.



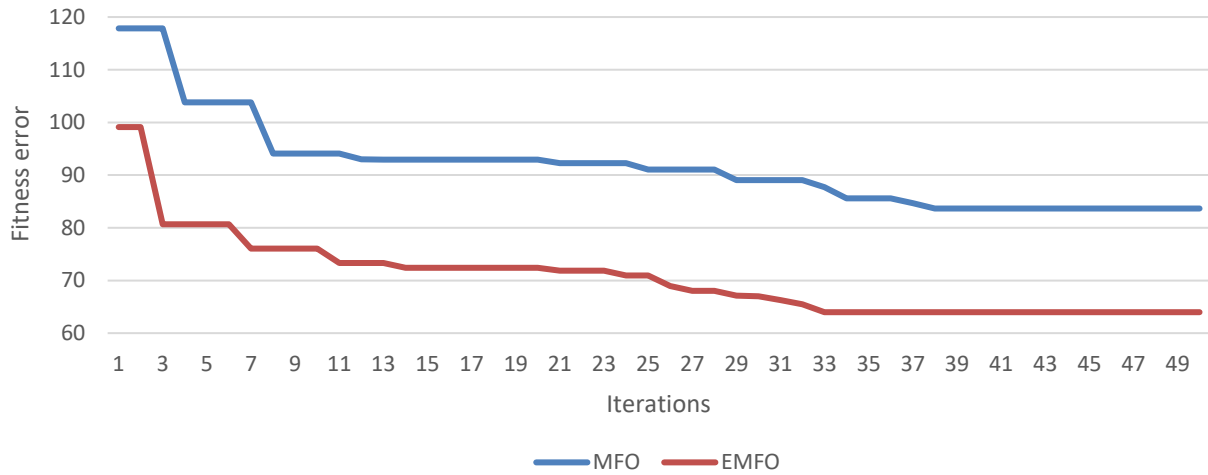
(a)



(b)



(c)



(d)

Fig. 3.11 Convergence diagram for MFO and EMFO for the cases: (a) Case 5, (b) Case 7, (c) Case 10, and (d) Case 12.

It can be inferred for all the scenarios that simulation results obtained by optimized gains of variable range SR-PID controller using EMFO algorithm were better as compared to those tuned using MFO method in terms of settling time (Tables 3.11–3.16) and utility consumptions (Tables 3.17–3.22). This is due to the optimized controller gains obtained through better global convergence using the EMFO algorithm.

Finally, it can be inferred from the above discussion that the EMFO algorithm provided the best results as compared to the original MFO, IMFO 1, IMFO 2, and IMFO 3 algorithms. It is because of better solutions found during the search process in solution space. Due to this, the algorithm converged fast into the best solution.

The chapter investigated the performance of the variable range SR-PID controller tuned offline using the different nature-inspired algorithms. The simulation results obtained from these algorithms were compared with the performance of the SR-PID controller tuned using the Z-N method in terms of settling time. It has been observed that as compared to both the PSO and WOA algorithms, MFO algorithm-based controller outperformed the Z-N method-based controller in all the scenarios, i.e., the effect of dead time in the valve, the effect of disturbance in the process, and utility consumption. To further improve the system's performance, this chapter also proposed modifications in the original MFO algorithm in three phases. A new version of the MFO (EMFO) algorithm was obtained by combining all the three phases of modification and used to tune the SR-PID controller in offline mode. It was found that the controller using the EMFO algorithm outperformed the controller using the original MFO algorithm in all the scenarios. Further, this work can be extended by updating the controller gain parameters online using the original and modified versions of the MFO algorithm for improving the performance of the variable range SR-PID controller.

Performance Evaluation of Online Tuned Variable Range SR-PID Controller

This chapter presents an online tuning approach using the original and improved versions of the MFO algorithms for optimizing the parameters of variable range split range PID controller to control the temperature of the mixing process. The performances of the controllers are investigated for the various temperature setpoints in terms of settling time and compared with performances obtained using the offline tuning approach with the MFO algorithm. Further, the performance of the online tuned controller using the proposed algorithm (EMFO) is investigated with respect to the effect of system dynamics and the effect of process disturbance.

Adaptive or continuously online tuned PID controllers are increasingly becoming popular as they outperform the conventional offline tuned PID controller in varying conditions (Hernández-Alvarado *et al.*, 2016; Kofinas & Dounis, 2019). The performance of the controller depends on the selection of the gains. Conventionally, the gains of the controllers are fixed and obtained by tuning the controller offline at an operating condition. However, the controller with fixed gains may fail to provide adequate control action due to (i) change in operating conditions, (ii) parameter drift in the system resulting from scaling of orifices, variations in instruments and sensors with time, (iii) sudden disturbances, and (iv) inherent nonlinearities of the system. Adaptive controllers overcome these difficulties as their gains are updated continuously online. This continuous updating of the controller gains enables the controller to handle the variations more effectively, resulting in improved performance of the plant (El-Gendy *et al.*, 2020; Memon & Shao, 2020; Colombino *et al.*, 2020; Kofinas & Dounis, 2018, 2019; Zhou *et al.*, 2019; Chen *et al.*, 2019; Davanipour *et al.*, 2018; Tamilselvan & Aarthy, 2017).

The use of metaheuristic algorithms (PSO, WOA, and MFO) for tuning the controller gains of the

PID controller used in a variable range of split range control scheme for regulating the temperature of a mixing process has been presented in Chapter 3. The performance of this offline (PSO, WOA, and MFO) tuned controller compared favorably with the performance of the conventional Ziegler Nichols (Z-N) tuned PID controller. However, since an online tuned controller provides a better control action as compared to an offline tuned controller in varying conditions, the current work presents a continuously online tuned PID controller employed in the variable range split range control scheme to provide an improved regulation of the temperature of a mixing process. The continuous updating of the gains of the controller is done using the MFO algorithm as it was found to be the best among the algorithms considered in Chapter 3.

The optimization algorithm used for the continuous online updating of the controller gains should converge fast to a solution that improves the system performance. With this aim, the enhanced MFO algorithm presented in Chapter 3 has been employed to continuously update the gains of the PID controller. This chapter also presents the effect of modifications in the algorithm on the performance of the online tuned PID controller.

4.1 Online tuning approach

This approach was used by considering the same gain parameters initially, which were obtained using the offline tuning method. During the cycle of the process, the controller gain parameters were updated continuously after a fixed interval of time, and the gain parameters were adjusted as per the current situation of the process (Vishnoi *et al.*, 2021b). The same objective function (used in Chapter 3) was considered in this approach for an appropriate comparison of the controller performance. The objective function is given below for reference.

$$J = w_1 * T_s + (1 - w_1) * E_{ss} \quad (4.1)$$

where w_1 denotes the weighting factor, T_s indicates setting time, and E_{ss} represents steady-state error. T_s and E_{ss} were assigned with equal weightage as 0.5 to calculate the fitness value.

4.2 Results and discussion

The mixing process using the variable range SR-PID controller for controlling the temperature of the water tank was modeled and simulated in MATLAB/Simulink. The controller gains were

initially obtained by the offline tuning approach, as described in Chapter 3. To further improve the performance of the controller, this work proposed to update the controller gains continuously during the cycle of the process. This updating of the controller gains was done using the MFO algorithm (observed as the best optimization algorithm among the others used) after fixed intervals of time to adjust the gain parameters to get optimum performance as per the current situation. A more frequent updating (shorter fixed interval) will give an improved performance but at the cost of the computational burden. In this work, the effect of the various duration of the fixed interval of time (after which the gains are to be updated) on the performance of the online tuned controller was investigated. The simulation results were obtained for different temperature setpoints within the working range (22.5°C to 32.5°C) and compared with the performance of the offline tuned controller in terms of settling time (T_s), as shown in [Table 4.1](#).

Table 4.1 Comparative study of the transient response of SR-PID controller tuned offline and online (with the various duration of the fixed interval of time) using the MFO algorithm.

Cases	Setpoints (°C)	Error	Offline	Duration of 3 sec.	Duration of 5 sec.	Duration of 8 sec.	Duration of 10 sec.	Duration of 20 sec.	Duration of 22 sec.
1	22.5	+3	176	129	131	138	142	168	177
2	22.5	+1	75	47	49	55	58	69	73
3	25	+3	245	181	187	201	209	236	247
4	25	+1	85	58	61	69	73	84	89
5	25	-1	206	174	180	188	191	204	208
6	25	-2	907	531	545	552	555	559	563
7	27.5	+3	419	309	319	326	331	369	378
8	27.5	+1	138	80	85	93	97	131	138
9	27.5	-1	139	81	85	92	96	130	137
10	27.5	-3	419	308	316	324	328	365	373
11	29	+3	818	647	657	668	673	692	696
12	29	+1	156	118	120	124	129	141	146
13	29	-1	93	64	70	74	77	88	92
14	29	-3	382	214	216	225	230	267	275
15	32.5	-1	73	45	47	49	53	66	72
16	32.5	-3	170	126	128	133	138	164	171

From [Table 4.1](#), it was found that the performance of the online tuned controller was better than the offline tuned controller for durations less than 20 seconds, beyond which the performances started deteriorating for some cases.

It was also observed from [Table 4.1](#) that the performance of the online tuned controller was enhanced with a reduction in the duration of the fixed interval. In the current work, this duration is taken as three seconds. The performance of the proposed online tuned controller with gain parameters updated after every three seconds was simulated for different temperature settings over the entire operating range. The simulation results thus obtained were compared with the performance of the offline tuned controller in terms of settling time. The simulation results obtained for the online and offline tuned SR-PID controller are presented in [Table 4.2](#).

Table 4.2 Comparative study of the transient response of SR-PID controller using MFO and improved MFO algorithms.

Cases	Setpoints (°C)	Error	MFO	MFO	IMFO 1	IMFO 2	IMFO 3	EMFO
			Offline	Online	Online	Online	Online	Online
1	22.5	+3	176	129	123	116	111	104
2	25	-1	206	174	163	156	142	105
3	27.5	+3	419	309	292	279	265	242
4	27.5	-3	419	308	294	282	269	245
5	29	+1	156	118	104	86	79	61
6	32.5	-3	170	126	121	115	112	107

The results clearly establish a superior performance of the online tuned controller over the offline tuned controller. The continuous up-gradation of the gain parameters enabled the online tuned controller to provide a more appropriate control action as per the prevailing conditions. As a result, settling time was reduced significantly.

In order to improve the performance of the online tuned controller, the modifications proposed in the original MFO algorithm in section 3.5 (Chapter 3) were incorporated. The performances of the controllers using IMFO 1, IMFO 2, IMFO 3, and EMFO for updating the controller gains during online tuning were simulated for the cases already under consideration. The simulation results in terms of settling times are presented in [Table 4.2](#). It is evident from the simulation results that each of the proposed modifications has led to improvements in the performance of the online tuned controller as compared to the original MFO algorithm. It is observed that IMFO 1 improved the exploitation by modifying the spiral path, IMFO 2 included a better initial population along with a better spiral path, resulting in a faster convergence to better solution in

the lesser iterations, whereas IMFO 3 improved exploration by including the better flames. The combination of these three modifications (EMFO) demonstrates the best performance as it combines the advantages of these three algorithms (IMFO 1, IMFO 2, IMFO 3). Therefore, the EMFO algorithm has an optimal initial population using a better spiral path to search around even fitter probable candidates for the global optima. This guarantees that optimal controller gains are provided at regular intervals to the controller throughout the cycle of the process resulting in a substantial reduction in the settling time for all the cases under consideration. A comparison of the relative performance of the controllers due to the modifications over the original MFO algorithm is presented in Fig. 4.1.

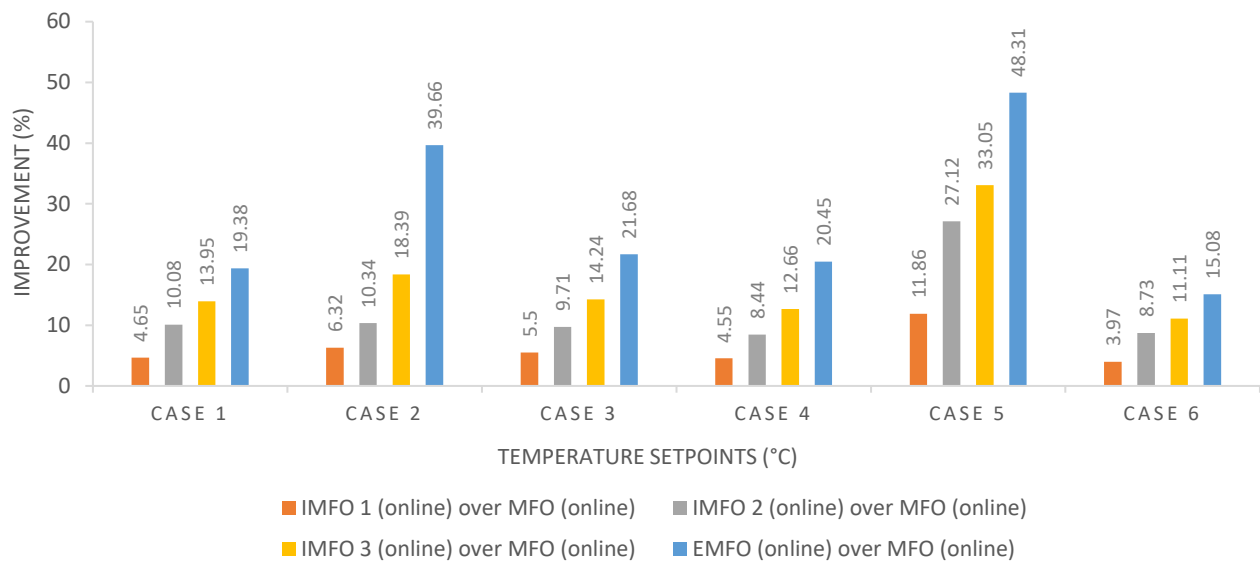
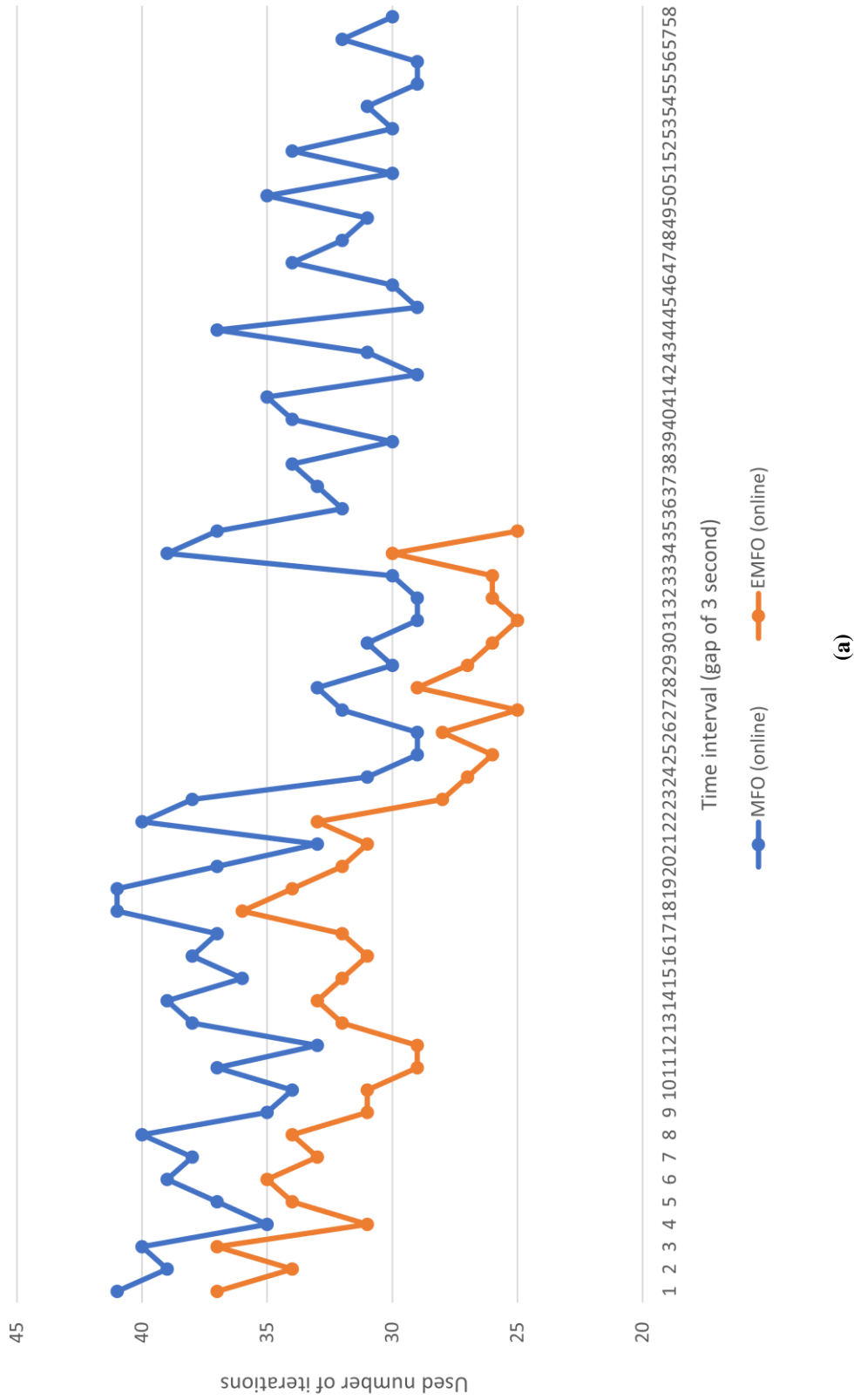
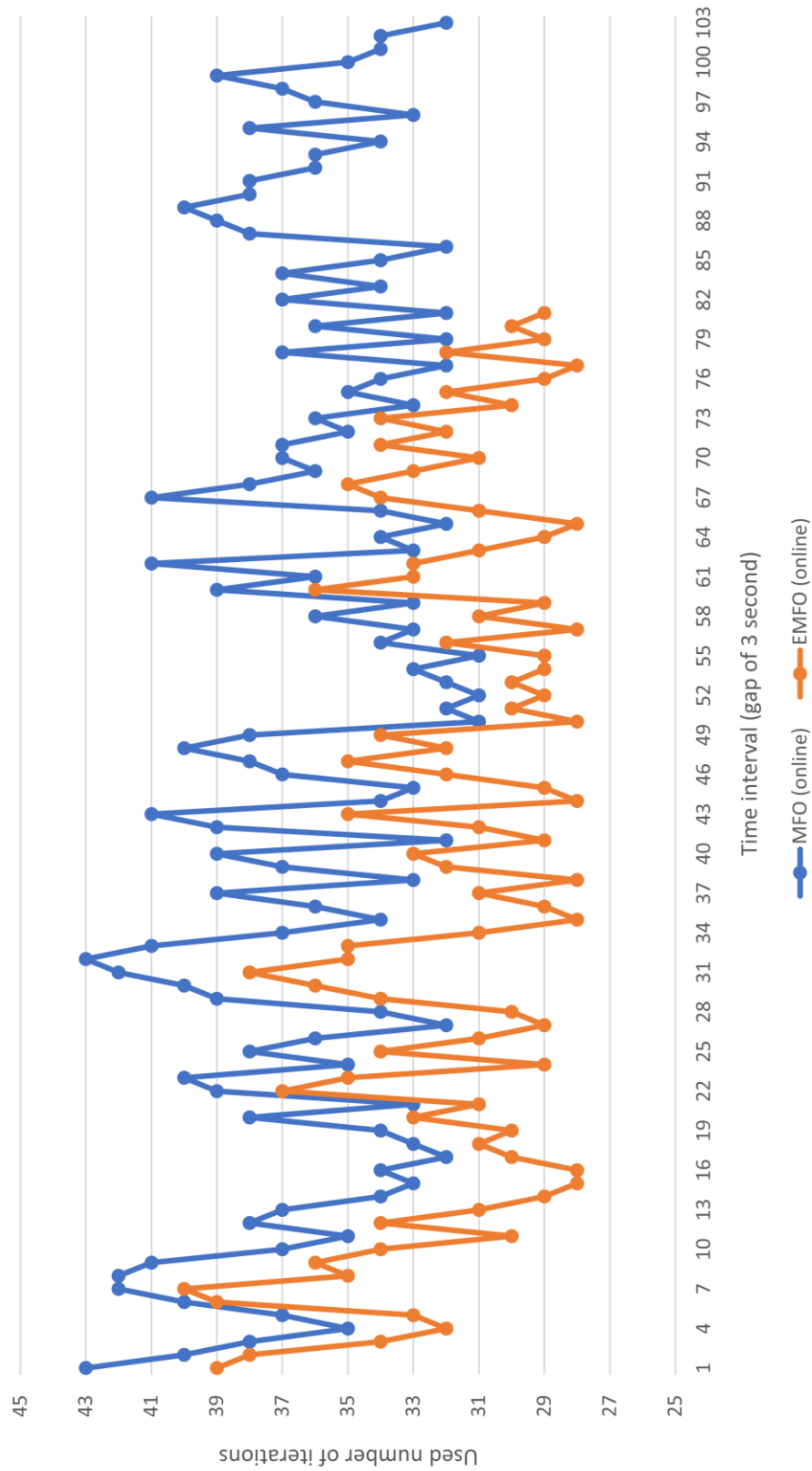


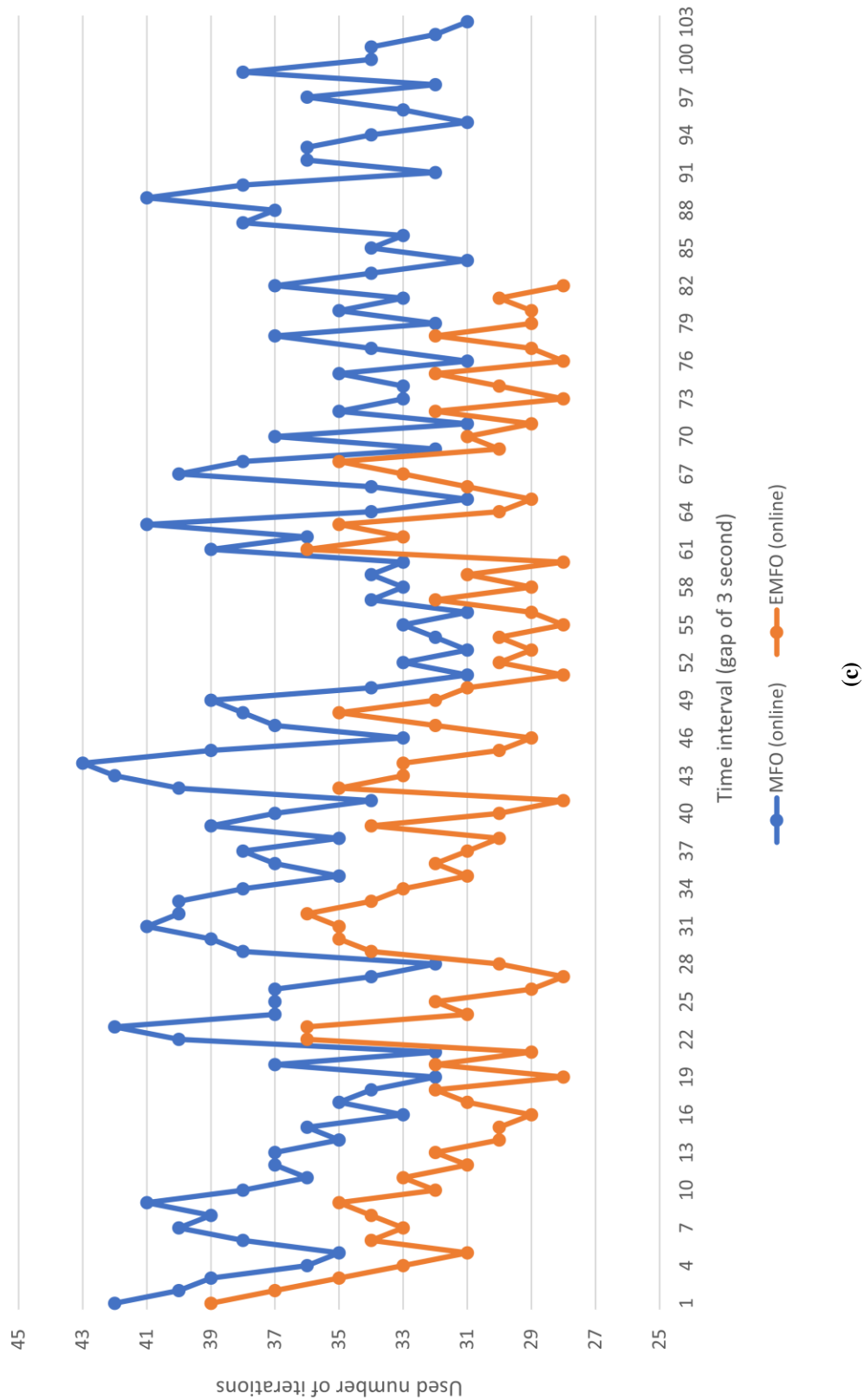
Fig. 4.1 Percentage improvement in the performances of SR-PID controller using MFO and improved MFO algorithms.

Based on the simulation results in Table 4.2 and the relative performance of the algorithms shown in Fig. 4.1, EMFO has been found to be the most efficient in improving the performance of the online tuned controller in terms of settling times. A comparison of the EMFO and the original MFO algorithms is then made in terms of the number of iterations required by these two algorithms to converge to the respective optimum solutions. The number of iterations required by these two algorithms after every three seconds to converge to the best controller gains for Cases 2, 3, 4, and 5 is shown in Fig. 4.2.





(b)



(c)

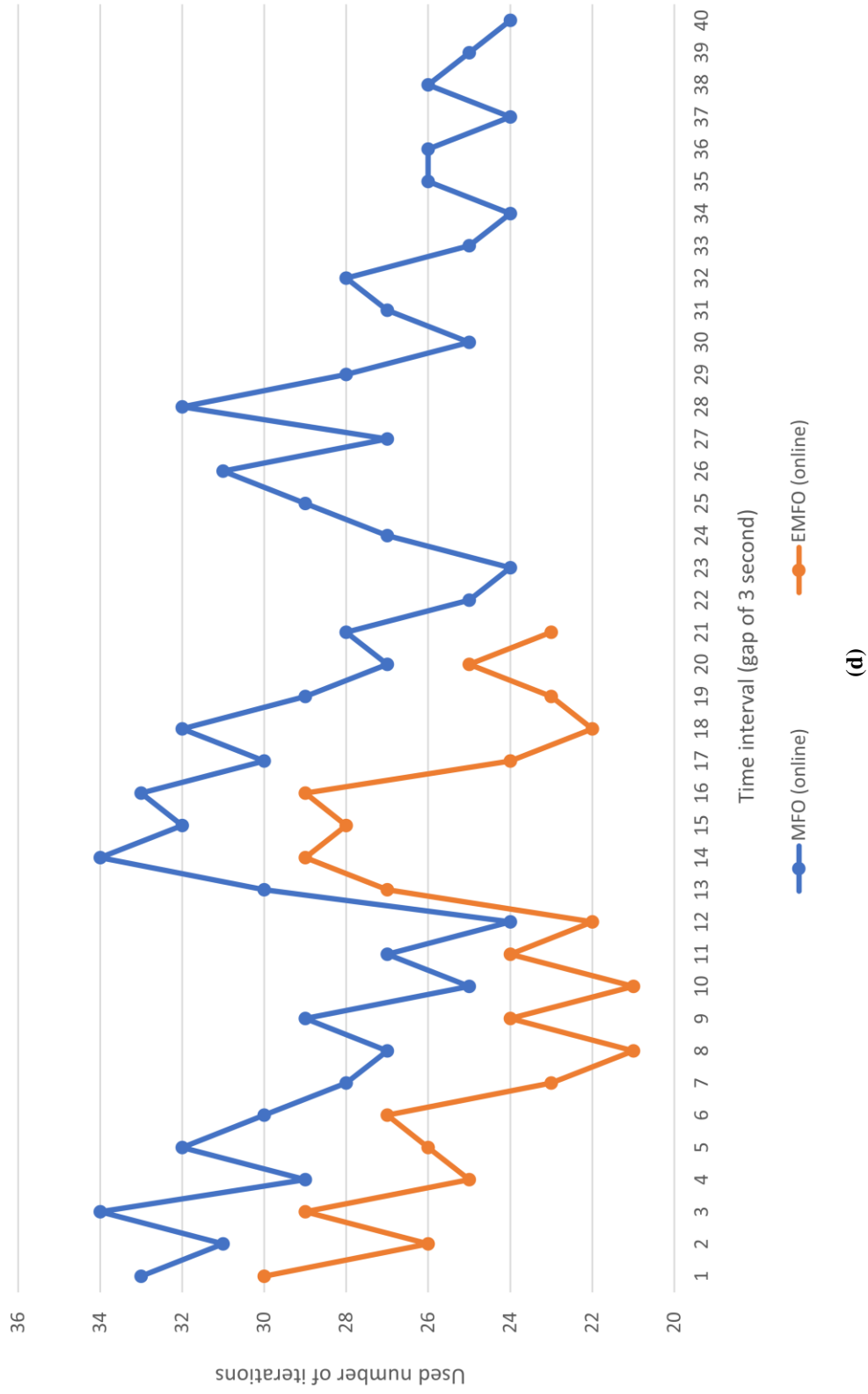


Fig. 4.2 Required number of iterations after every 3 seconds of a time interval for the cases: (a) Case 2, (b) Case 3, (c) Case 4, (d) Case 5.

It is clear from Fig. 4.2 that EMFO converged to the best solution in a smaller number of iterations as compared to the original MFO for each update in every case. This establishes that of all the proposed algorithms, EMFO provides the best solutions in a smaller number of steps as compared to the original MFO.

Further, the investigations were made to study the effect of dead time in the valve and process disturbance on the performance of the online tuned controller. The performances of the controller are also studied in terms of utility consumption. These investigations also included a comparison of the performance of the proposed EMFO and the original MFO in updating the controller gains of the online tuned controller for these conditions.

4.2.1 Effect of system dynamics

The system performance was checked with dead times (considering a dead time of 0, 0.25, and 0.5 seconds in the valve). Further, a comparison was made for the transient performance (in terms of T_s) of the SR-PID controller for the various temperature setpoints using the original MFO (offline and online tuning approach) and EMFO (online tuning approach) algorithms with and without dead time is shown in Tables 4.3–4.5.

Table 4.3 Performances of SR-PID controller using MFO and EMFO algorithms with no dead time.

Cases	Setpoints (°C)	Error	Valve (with no dead time)			Improvement (%)	
			MFO Offline (T_s)	MFO Online (T_s)	EMFO Online (T_s)	MFO (Online) over MFO(Offline)	EMFO (Online) over MFO (Offline)
1	22.5	+3	176	129	104	26.70	40.91
2	22.5	+1	75	47	40	37.33	46.67
3	25	+3	245	181	139	26.82	43.27
4	25	+1	85	58	42	31.76	50.59
5	25	-1	206	174	105	15.53	49.03
6	25	-2	907	531	448	41.46	50.61
7	27.5	+3	419	309	242	26.25	42.24
8	27.5	+1	138	80	56	42.03	59.42
9	27.5	-1	139	81	58	41.73	58.27
10	27.5	-3	419	308	245	26.49	41.53
11	29	+3	818	647	552	20.90	32.52
12	29	+1	156	118	61	24.36	60.89
13	29	-1	93	64	49	31.18	47.31
14	29	-3	382	214	176	43.98	53.93
15	32.5	-1	73	45	38	38.36	47.95
16	32.5	-3	170	126	107	25.88	37.06

Table 4.4 Performances of SR-PID controller using MFO and EMFO algorithms with a dead time of 0.25 sec.

Cases	Setpoints (°C)	Error	Valve (with a dead time of 0.25 second)			Improvement (%)	
			MFO Offline (T_s)	MFO Online (T_s)	EMFO Online (T_s)	MFO (Online) over MFO(Offline)	EMFO (Online) over MFO (Offline)
			1	22.5	+3	199	148
2	22.5	+1	87	56	46	35.63	47.13
3	25	+3	282	213	156	26.07	44.68
4	25	+1	99	70	50	29.29	49.49
5	25	-1	254	196	123	22.83	51.57
6	25	-2	1110	594	478	46.49	56.94
7	27.5	+3	511	352	267	31.12	47.75
8	27.5	+1	157	93	68	40.76	56.69
9	27.5	-1	158	92	69	41.77	56.33
10	27.5	-3	512	349	269	31.84	47.46
11	29	+3	1016	702	587	30.91	42.22
12	29	+1	171	131	78	23.39	54.39
13	29	-1	107	76	57	28.97	46.73
14	29	-3	461	247	192	46.42	58.35
15	32.5	-1	86	52	44	39.53	48.84
16	32.5	-3	198	147	117	25.76	40.91

Table 4.5 Performances of SR-PID controller using MFO and EMFO algorithms with a dead time of 0.5 sec.

Cases	Setpoints (°C)	Error	Valve (with a dead time of 0.5 second)			Improvement (%)	
			MFO Offline (T_s)	MFO Online (T_s)	EMFO Online (T_s)	MFO (Online) over MFO(Offline)	EMFO (Online) over MFO (Offline)
			1	22.5	+3	244	181
2	22.5	+1	98	68	55	30.61	43.88
3	25	+3	345	254	179	26.38	48.12
4	25	+1	108	85	62	21.29	42.59
5	25	-1	314	247	143	21.34	54.46
6	25	-2	1358	713	519	47.49	61.78
7	27.5	+3	629	425	293	32.43	53.42
8	27.5	+1	176	112	84	36.36	52.27
9	27.5	-1	180	110	86	38.89	52.22
10	27.5	-3	632	421	295	33.39	53.32
11	29	+3	1262	918	633	27.26	49.84
12	29	+1	212	155	101	26.89	52.36
13	29	-1	122	90	68	26.23	44.26
14	29	-3	556	308	215	44.60	61.33
15	32.5	-1	94	61	52	35.11	44.68
16	32.5	-3	243	183	133	24.69	45.27

Based on the simulation results, it can be said from [Tables 4.3–4.5](#) that the SR-PID controller tuned online with the EMFO algorithm outperforms the controller with the original MFO algorithm (offline and online tuning approaches). Dead time in the valve makes it take longer for the manipulated variables to change. Because of this, the system will react slowly leading to longer settling times in all circumstances when compared to a system with no dead time. It can be seen from the simulation results that as the dead time of the valve increases, the settling times increase. However, the controller tuned online using the EMFO algorithm exhibits a superior performance as compared to the controller tuned online using the original MFO algorithm.

4.2.2 Effect of process disturbance

The system under consideration was then checked for the effect of disturbance (considered as impulse function in the forward path in the process at $t=30$ seconds) on the controller performance using impulse function. The performances of the controller using MFO and EMFO was tested and compared for the above-considered temperature setpoints in terms of settling time, as shown in [Tables 4.6–4.8](#).

Table 4.6 Performances of SR-PID controller using MFO and EMFO algorithms with disturbance.

Cases	Setpoints (°C)	Error	MFO Offline (T_s)	MFO Online (T_s)	EMFO Online (T_s)	Improvement (%)	
						MFO (Online) over MFO(Offline)	EMFO (Online) over MFO (Offline)
						1	22.5
2	22.5	+1	85	49	40	42.35	52.94
3	25	+3	262	185	141	30.39	46.18
4	25	+1	97	60	43	38.14	55.67
5	25	-1	259	180	108	30.50	58.30
6	25	-2	976	545	454	44.16	53.48
7	27.5	+3	445	316	245	28.99	44.94
8	27.5	+1	159	83	57	47.79	64.15
9	27.5	-1	160	85	59	46.86	63.13
10	27.5	-3	442	317	249	28.28	43.67
11	29	+3	864	656	557	24.07	35.53
12	29	+1	192	123	62	35.94	67.71
13	29	-1	106	67	50	36.79	52.83
14	29	-3	405	219	179	45.93	55.80
15	32.5	-1	82	47	39	42.68	52.44
16	32.5	-3	181	129	109	28.73	39.78

Table 4.7 Performances of SR-PID controller using MFO and EMFO algorithms with disturbance and a dead time of 0.25 sec.

Cases	Setpoints (°C)	Error	Valve (with a dead time of 0.25 second)			Improvement (%)	
			MFO Offline (T_s)	MFO Online (T_s)	EMFO Online (T_s)	MFO (Online) over MFO(Offline)	EMFO (Online) over MFO (Offline)
1	22.5	+3	213	153	118	28.17	44.61
2	22.5	+1	98	59	47	39.81	52.04
3	25	+3	301	219	159	28.54	47.18
4	25	+1	112	74	52	33.93	53.57
5	25	-1	308	204	127	33.77	58.77
6	25	-2	1180	610	485	48.31	58.91
7	27.5	+3	538	361	271	32.91	49.63
8	27.5	+1	179	98	70	45.25	60.89
9	27.5	-1	180	98	71	45.56	60.56
10	27.5	-3	536	360	274	32.84	48.88
11	29	+3	1063	713	593	32.93	44.21
12	29	+1	208	138	80	33.65	61.54
13	29	-1	121	81	59	33.06	51.24
14	29	-3	485	254	196	47.63	59.59
15	32.5	-1	96	55	46	42.71	52.08
16	32.5	-3	210	152	120	27.62	42.86

Table 4.8 Performances of SR-PID controller using MFO and EMFO algorithms with disturbance and a dead time of 0.5 sec.

Cases	Setpoints (°C)	Error	Valve (with a dead time of 0.5 second)			Improvement (%)	
			MFO Offline (T_s)	MFO Online (T_s)	EMFO Online (T_s)	MFO (Online) over MFO(Offline)	EMFO (Online) over MFO (Offline)
1	22.5	+3	260	188	136	28.69	47.69
2	22.5	+1	111	73	57	34.23	48.65
3	25	+3	365	262	184	28.22	49.59
4	25	+1	123	91	66	26.02	46.34
5	25	-1	370	257	149	30.54	59.73
6	25	-2	1430	731	528	48.88	63.08
7	27.5	+3	658	436	299	33.74	54.56
8	27.5	+1	200	119	88	40.51	56.01
9	27.5	-1	204	118	90	42.16	55.88
10	27.5	-3	658	434	302	34.04	54.11
11	29	+3	1311	931	641	28.99	51.11
12	29	+1	251	164	105	34.66	58.17
13	29	-1	138	97	72	29.71	47.83
14	29	-3	582	317	221	45.53	62.03
15	32.5	-1	106	66	55	37.74	48.11
16	32.5	-3	257	190	138	26.07	46.31

The results in Tables 4.6–4.8 show that the EMFO algorithm (online tuning approach) provided better control performance in the presence of disturbance in the process, as compared to the original MFO algorithm (offline and online tuning approach). Based on Tables 4.6–4.8, it can be inferred that in the presence of disturbance into the process, the performance of the control system in terms of settling time will be influenced, and the system will take longer time to reach the desired temperature as compared to the system without disturbance. This time increases further when the dead time in the valve increases.

4.2.3 Utility consumption

The investigation of the controller performance based on the utility consumption is carried out for the above-considered setpoints. The consumption of utility obtained using the controller with MFO and EMFO is presented in Tables 4.9–4.14. It can be observed from the simulation results that the utility consumption of cold and hot water using SR-PID controller tuned online with EMFO algorithm was lesser as compared to controller tuned offline and online with the original MFO algorithm.

Table 4.9 Comparative study of the amount of Q_1 and Q_2 flows using SR-PID controller with MFO (offline and online) and EMFO (online) in case of valve with no dead time.

Cases	Setpoints (°C)	Error	MFO Offline		MFO Online		EMFO Online	
			u_c	u_h	u_c	u_h	u_c	u_h
1	22.5	+3	5.11	13.95	4.22	10.63	3.54	8.87
2	22.5	+1	4.25	8.20	4.01	5.29	3.16	4.42
3	25	+3	5.85	19.81	4.97	15.20	3.80	12.56
4	25	+1	5.21	9.65	4.39	5.50	3.68	4.77
5	25	-1	18.45	7.24	14.22	5.14	10.58	4.65
6	25	-2	69.49	13.91	41.59	8.39	33.14	6.08
7	27.5	+3	7.38	34.46	5.48	25.19	4.38	18.41
8	27.5	+1	5.29	12.59	4.82	7.18	4.09	6.08
9	27.5	-1	13.58	6.06	7.11	4.73	6.02	4.03
10	27.5	-3	33.26	7.14	25.07	5.39	18.3	4.08
11	29	+3	12.58	62.81	10.14	50.33	8.81	40.96
12	29	+1	6.21	14.01	5.54	10.46	4.62	8.93
13	29	-1	9.47	5.64	5.67	4.19	4.76	3.51
14	29	-3	29.49	5.93	17.39	4.31	13.72	3.69
15	32.5	-1	7.97	4.14	5.12	3.91	4.32	3.07
16	32.5	-3	13.58	4.92	10.55	4.14	8.74	3.44

Table 4.10 Comparative study of the amount of Q_1 and Q_2 flows using SR-PID controller with MFO (offline and online) and EMFO (online) in case of valve with a dead time of 0.25 sec.

Cases	Setpoints (°C)	Error	MFO Offline		MFO Online		EMFO Online	
			u_c	u_h	u_c	u_h	u_c	u_h
1	22.5	+3	5.45	15.76	4.47	11.9	3.72	9.84
2	22.5	+1	4.53	9.26	4.25	5.92	3.32	4.91
3	25	+3	6.24	22.37	5.27	17.02	3.99	13.94
4	25	+1	5.56	10.9	4.65	6.16	3.86	5.29
5	25	-1	19.69	8.18	15.07	5.75	11.11	5.16
6	25	-2	74.14	15.71	44.07	9.39	34.79	6.75
7	27.5	+3	7.87	38.92	5.81	28.2	4.61	20.43
8	27.5	+1	5.64	14.22	5.11	8.04	4.29	6.75
9	27.5	-1	14.49	6.84	7.53	5.31	6.32	4.47
10	27.5	-3	35.49	8.06	26.57	6.03	19.21	4.53
11	29	+3	13.42	70.94	10.75	56.34	9.25	45.44
12	29	+1	6.63	15.82	5.87	11.71	4.85	9.91
13	29	-1	10.11	6.37	6.01	4.69	5.01	3.89
14	29	-3	31.46	6.71	18.43	4.82	14.4	4.09
15	32.5	-1	8.51	4.68	5.43	4.38	4.53	3.41
16	32.5	-3	14.49	5.56	11.18	4.63	9.17	3.82

Table 4.11 Comparative study of the amount of Q_1 and Q_2 flows using SR-PID controller with MFO (offline and online) and EMFO (online) in case of valve with a dead time of 0.5 sec.

Cases	Setpoints (°C)	Error	MFO Offline		MFO Online		EMFO Online	
			u_c	u_h	u_c	u_h	u_c	u_h
1	22.5	+3	5.98	18.27	4.81	13.52	3.96	11.08
2	22.5	+1	4.97	10.74	4.57	6.72	3.54	5.52
3	25	+3	6.84	25.94	5.67	19.33	4.25	15.7
4	25	+1	6.11	12.64	4.99	7.01	4.11	5.96
5	25	-1	21.6	9.48	16.21	6.53	11.84	5.81
6	25	-2	81.33	18.21	47.41	10.67	37.08	7.61
7	27.5	+3	8.63	45.13	6.25	32.03	4.91	23.02
8	27.5	+1	6.19	16.49	5.51	9.13	4.57	7.61
9	27.5	-1	15.89	7.93	8.11	6.02	6.74	5.03
10	27.5	-3	38.93	9.35	28.58	6.85	20.47	5.11
11	29	+3	14.72	82.25	11.56	64.01	9.86	51.16
12	29	+1	7.27	18.34	6.31	13.3	5.17	11.16
13	29	-1	11.08	7.39	6.47	5.33	5.33	4.38
14	29	-3	34.51	7.77	19.83	5.48	15.35	4.61
15	32.5	-1	9.32	5.43	5.84	4.98	4.83	3.84
16	32.5	-3	15.89	6.45	12.03	5.26	9.77	4.31

Table 4.12 Comparative study of the amount of Q_1 and Q_2 flows using SR-PID controller with MFO (offline and online) and EMFO (online) in case of disturbance.

Cases	Setpoints (°C)	Error	MFO Offline		MFO Online		EMFO Online	
			u_c	u_h	u_c	u_h	u_c	u_h
1	22.5	+3	5.64	15.51	4.34	11.01	3.62	9.14
2	22.5	+1	4.69	9.12	4.12	5.48	3.17	4.48
3	25	+3	6.46	22.03	5.11	15.75	3.88	12.95
4	25	+1	5.75	10.73	4.51	5.71	3.76	4.92
5	25	-1	20.38	8.05	14.61	5.32	10.81	4.79
6	25	-2	80.92	15.47	42.73	8.69	33.85	6.27
7	27.5	+3	8.15	38.32	5.63	26.1	4.47	18.98
8	27.5	+1	5.84	14.03	4.95	7.44	4.18	6.27
9	27.5	-1	15.03	6.74	7.31	4.91	6.15	4.15
10	27.5	-3	36.73	7.94	25.76	5.58	18.69	4.21
11	29	+3	13.89	72.35	10.42	52.14	9.01	42.23
12	29	+1	6.86	15.58	5.69	10.84	4.72	9.21
13	29	-1	10.46	6.27	5.83	4.34	4.86	3.62
14	29	-3	32.57	6.59	17.87	4.46	14.01	3.81
15	32.5	-1	8.81	4.62	5.26	4.05	4.41	3.17
16	32.5	-3	15.18	5.47	10.84	4.29	8.93	3.55

Table 4.13 Comparative study of the amount of Q_1 and Q_2 flows using SR-PID controller with MFO (offline and online) and EMFO (online) in case of disturbance and valve with a dead time of 0.25 sec.

Cases	Setpoints (°C)	Error	MFO Offline		MFO Online		EMFO Online	
			u_c	u_h	u_c	u_h	u_c	u_h
1	22.5	+3	6.07	17.6	4.68	12.49	3.86	10.23
2	22.5	+1	5.05	10.34	4.45	6.21	3.44	5.09
3	25	+3	6.95	24.99	5.52	17.86	4.14	14.49
4	25	+1	6.21	12.17	4.87	6.46	4.01	5.51
5	25	-1	21.94	9.14	15.78	6.03	11.53	5.36
6	25	-2	82.62	17.55	46.16	9.85	36.09	7.02
7	27.5	+3	8.77	43.47	6.09	29.6	4.77	21.24
8	27.5	+1	6.29	15.88	5.35	8.44	4.45	7.02
9	27.5	-1	16.15	7.64	7.89	5.56	6.56	4.65
10	27.5	-3	39.55	9.03	27.83	6.33	19.93	4.71
11	29	+3	14.96	79.24	11.26	59.13	9.61	47.23
12	29	+1	7.39	17.67	6.15	12.29	5.03	10.31
13	29	-1	11.26	7.11	6.31	4.92	5.19	4.04
14	29	-3	35.06	7.48	19.29	5.06	14.94	4.25
15	32.5	-1	9.47	5.23	5.69	4.61	4.71	3.54
16	32.5	-3	16.15	6.21	11.71	4.86	9.51	3.97

Table 4.14 Comparative study of the amount of Q_1 and Q_2 flows using SR-PID controller with MFO (offline and online) and EMFO (online) in case of disturbance and valve with a dead time of 0.5 sec.

Cases	Setpoints (°C)	Error	MFO Offline		MFO Online		EMFO Online	
			u_c	u_h	u_c	u_h	u_c	u_h
1	22.5	+3	6.72	20.48	5.09	14.32	4.11	11.48
2	22.5	+1	5.59	12.04	4.83	7.12	3.67	5.72
3	25	+3	7.69	29.08	6.01	20.48	4.41	16.26
4	25	+1	6.86	14.17	5.29	7.42	4.26	6.17
5	25	-1	24.29	10.63	17.14	6.92	12.26	6.02
6	25	-2	91.45	20.41	50.13	11.3	38.41	7.87
7	27.5	+3	9.71	50.59	6.61	33.94	5.08	23.83
8	27.5	+1	6.96	18.48	5.82	9.67	4.73	7.87
9	27.5	-1	17.87	8.89	8.57	6.38	6.98	5.21
10	27.5	-3	43.77	10.48	30.22	7.26	21.2	5.28
11	29	+3	16.55	92.2	12.22	67.81	10.21	53.01
12	29	+1	8.17	20.56	6.67	14.09	5.35	11.56
13	29	-1	12.46	8.28	6.84	5.65	5.52	4.54
14	29	-3	38.81	8.71	20.97	5.81	15.9	4.78
15	32.5	-1	10.48	6.09	6.18	5.28	5.01	3.98
16	32.5	-3	17.87	7.23	12.72	5.57	10.12	4.45

The findings in [Tables 4.9–4.14](#) also demonstrate that when we move upward/ downward in the temperature's working range, hot/ cold utility consumption increases for the same desired rise/ fall in the temperature. From [Tables 4.9–4.11](#), it can also be seen that the hot and cold utility consumptions (u_h and u_c) increase as the dead time in the valve increases. The utility consumption increases further when the system with dead time in the valve is subjected to disturbance ([Tables 4.13–4.14](#)).

At the end, it can be said from the above discussion that even in case of online tuning, the EMFO algorithm outperformed the original MFO, IMFO 1, IMFO 2, and IMFO 3 algorithms. It is due to the improved solutions found during the search process in solution space. As a result, the algorithm converged fast into the best solution.

In this chapter, MFO algorithm-based online tuning approach was proposed for optimizing the gain parameters of the SR-PID controller for temperature control of the mixing process. The online tuning approach yielded a better performance as compared to the offline strategy in terms of settling time. By varying the MFO scheme (path, initial population, and flame selection) with

online tuning approach, a further improvement was observed in the performance. It is pertinent to mention that the effect of system dynamics and process disturbance is also taken care of. Further, this work can be extended by the performance investigation of the proposed EMFO algorithm in the real environment.

Validation

This chapter investigates the performances of the online tuned controllers in the simulated real environment. The effect of using the EMFO algorithm for online tuning of the controller in the simulated real environment is studied. An electrical analogous model of the practical environment is simulated for investigation by considering several effects (imperfect insulation, density, viscosity, and compressibility) found in real-time conditions. Further, the system is also investigated with the effect of system dynamics, and process disturbance. Moreover, a comparison of the performances of variable range SR-PID controller tuned online using EMFO algorithm in case of the ideal environment and the practical environment is also studied.

Keeping in view the importance of nature-inspired algorithms and the online tuning approach, the MFO algorithm and its enhanced version (EMFO) based online tuned variable range SR-PID controller for controlling the temperature of the mixing process were discussed in Chapter 4. The results were found encouraging with the proposed EMFO algorithm. Further, the performance of the proposed algorithm in the real environment needs to be investigated. The electrical analogous model has been established to represent the model of thermal system reliably (Aleksiejuk *et al.*, 2018; Chen *et al.*, 2015; Gilaber & Paris, 1988). The electrical systems also have the advantages of good accuracy, compact size, zero leakage, and low interference. Hence, an electrical analogous model was made for validation purposes.

In this chapter, the performance of the EMFO algorithm was investigated by simulating the real environment of the practical system, incorporating the various effects (Gilaber & Paris, 1988) seen in real-time situations, namely imperfect insulation, density, viscosity, and compressibility, via the electrical analogous system (Vishnoi *et al.*, 2022). Since, both the viscosity and the compressibility parameters have a linear relationship with the density, this work considered the

effect of imperfect insulation and density. Moreover, the system was also investigated with regard to the system dynamics and the disturbance in the process.

An enhanced moth flame optimization algorithm was proposed for online tuning of the controller gains, as discussed in Chapter 4. The simulation results established that the proposed EMFO algorithm converged faster to better gain values resulting in improved control actions as compared to the original MFO algorithm. The current work aims to investigate the performance of the MFO and EMFO algorithms in online tuning of the controller gains for the system under consideration in the practical environment. For this, the thermal-electrical analogy was used to make an electrical analogous model of the system. This electrical analogous model was then modified considering the effects of imperfect insulation and density.

5.1 Electrical analogous model

The water tank system can be visualized as a thermal system with the flow of fluid (water) as a heat source. The rate at which heat accumulates in the system due to both the inlet fluids are expressed as:

$$Q_{1in} = G_1 c_p (T_1 - T) \quad (5.1)$$

$$Q_{2in} = G_2 c_p (T_2 - T) \quad (5.2)$$

where G_1 and G_2 represent the mass flow rate of the inflowing fluids (kg/sec), T_1 and T_2 signify the temperature of the inlet fluids ($^{\circ}\text{C}$), T is the temperature of outflow from the tank ($^{\circ}\text{C}$), and c_p denotes the specific heat of the fluid ($\text{J/kg}\cdot^{\circ}\text{C}$).

The rate at which heat is accumulated into the system because of the two inlets is also given by

$$\text{Heat accumulated} = C \frac{d(T)}{dt} \quad (5.3)$$

where C represents the thermal capacitance of the fluid in the tank ($\text{J}/^{\circ}\text{C}$) and is given by

$$C = m \cdot c_p \quad (5.4)$$

where m represents the mass of the fluid in the tank (kg).

Fig. 5.1 depicts the thermal system for the process used in this study.

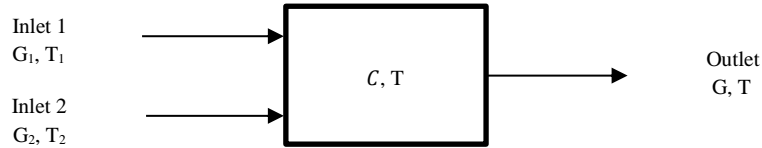


Fig. 5.1 Mixing Process.

From Eq. 5.1–5.3, the heat accumulated in the system can be written as

$$G_1 c_P (T_1 - T) + G_2 c_P (T_2 - T) = C \frac{d(T)}{dt} \tag{5.5}$$

Eq. 5.5 represents a thermal system that can be modeled using an electrical analogy, as seen in Table 5.1.

Table 5.1 Electrical analogous quantities of thermal system.

Thermal (Unit)	Electrical (Unit)
Heat (J)	Charge (C)
Heat flow rate (J/sec.)	Current (A)
Thermal capacitance (J/°C)	Capacitance (F)
Thermal resistance [°C/(J/sec.)]	Resistance (Ω)
Temperature (°C)	Voltage (V)

The fluid that flows from both the inlets is modeled by voltage sources V_1 (equivalent to T_1) and V_2 (equivalent to T_2) with resistances $1/ G_1 c_P$ and $1/ G_2 c_P$ respectively, as presented in Fig. 5.2. The thermal and the equivalent electrical parameters are indicated in standard font style and in bold, respectively.

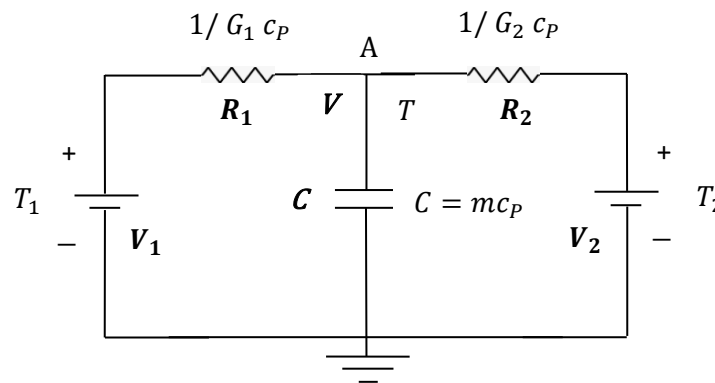


Fig. 5.2 Analogous electrical system.

One of the ends of the thermal capacitance is always connected to the ambient temperature. Therefore, the corresponding capacitance in the electrical model has one end connected to the reference, i.e., the ground. The system under consideration is assumed to be perfectly insulated (infinite thermal resistance). Accordingly, the corresponding resistance in the electrical model is also infinite and hence, no resistance is placed in parallel to the capacitance in the electrical circuit.

Mass flow rate can be determined as follows:

$$\text{Mass flow rate } (G) = \text{Density } (\rho) * \text{Volume flow rate } (Q) \quad (5.6)$$

Therefore, with the effect of density, Eq. 5.5 can be written as follows:

$$Q_1 \rho_1 c_p T_1 + Q_2 \rho_2 c_p T_2 - (Q_1 + Q_2) \rho c_p T = V \rho c_p \frac{d(T)}{dt} \quad (5.7)$$

The temperature has an effect on the density of any fluid. Even in the current work, there is a significant variation in the density of water over the considered operating range of temperature [22.5 32.5] °C. This variation in density will affect the mass flow rate of water at the two inlets which in turn, will have an impact on the actual heat flowing into the system. Therefore, the values of G_1 and G_2 in Fig. 5.2 are calculated using the values of density of water at 20°C and 35°C respectively. Accordingly, the range of values of G_1 and G_2 corresponding to the considered range of volume flow rates for Q_1 and Q_2 are given in Table 5.2. The specific heat of water (c_p) is taken as 4184 J/(kg-°C).

Table 5.2 G_1 and G_2 ranges corresponding to Q_1 and Q_2 .

Range	Volume flow rates		Mass flow rates	
	Q_1	Q_2	G_1 (at $T_1=20^\circ\text{C}$ and $\rho_1 = 0.99820$ kg/l.)	G_2 (at $T_2=35^\circ\text{C}$ and $\rho_2 = 0.994029$ kg/l.)
Min	0.015 l/sec.	0.015 l/sec.	0.01497 kg/sec.	0.01491 kg/sec.
Max	0.075 l/sec.	0.075 l/sec.	0.07487 kg/sec.	0.07455 kg/sec.

The volume flow rates are manipulated by the controllers to control the temperature of the tank, T . The variations in the volume flow rates will affect the mass flow rates G_1 and G_2 which in turn, will affect the resistances R_1 and R_2 of the analogous electrical model shown in Fig. 5.2.

With constant V_1 and V_2 , any change in R_1 and R_2 will affect V , the voltage at node 'A', which corresponds to the tank temperature.

However, in the practical environment, there shall be some heat loss due to imperfect insulation. As a result, the net heat flow into the system (heat in – heat out) shall be equal to the algebraic sum of heat accumulated and heat loss as given below:

$$G_1 c_p (T_1 - T) + G_2 c_p (T_2 - T) = C \frac{d(T)}{dt} + \frac{T}{R}$$

The above equation can be extended as follow:

$$Q_1 \rho c_p T_1 + Q_2 \rho c_p T_2 - (Q_1 + Q_2) \rho c_p T = V \rho c_p \frac{d(T)}{dt} + \frac{T}{R} \quad (5.8)$$

where, $\frac{T}{R}$ represents the heat losses from the tank, and R is thermal resistance.

Accordingly, the analogous electrical system shown in Fig. 5.2 shall be modified as shown in Fig. 5.3. Due to imperfect insulation, thermal resistance will no longer be infinite and will have some finite value. This is represented in Fig. 5.3 by the resistance R in parallel with the capacitance C . The range of G_1 and G_2 will be the same as the considered range of Q_1 and Q_2 because the density has been assumed to be independent of temperature and taken as 1.

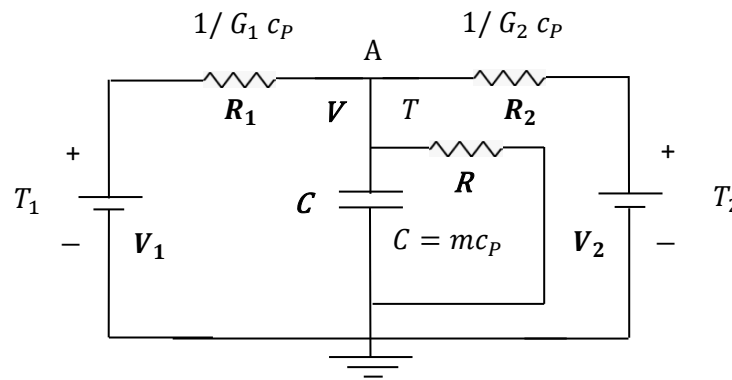


Fig. 5.3 Analogous electrical system with consideration of imperfect insulation.

Both the flow rates Q_1 and Q_2 are varied by the controller to regulate the tank temperature, T . The variations in Q_1 and Q_2 will change G_1 and G_2 , which will affect the resistances R_1 and R_2 in the corresponding electrical circuit model presented in Fig. 5.3.

Further, the system considered the effect of density along with the effect of imperfect insulation. The net heat flow into the system can be defined as:

$$Q_1\rho_1c_pT_1 + Q_2\rho_2c_pT_2 - (Q_1 + Q_2)\rho c_pT = V\rho c_p \frac{d(T)}{dt} + \frac{T}{R} \quad (5.9)$$

Analogous electrical system with consideration of the combined effect will be same as [Fig. 5.3](#). The mass flow rates G_1 and G_2 will be determined using the water density values at 20°C and 35°C respectively, as shown in [Table 5.2](#).

In the combined effect, G_1 and G_2 will be changed due to the effect of density as well as the variations in Q_1 and Q_2 , which will affect the values of R_1 and R_2 in the corresponding equivalent electrical system.

With voltages V_1 and V_2 remaining constant, any change in resistances R_1 and R_2 will impact the voltage V at node 'A' (corresponding to the tank temperature).

5.2 Results and discussion

The variable range SR-PID controller was optimized online using the EMFO algorithm to regulate the temperature of the mixing process in the simulated practical environment. The practical environment was simulated using MATLAB/Simulink by making the electrical analogous model considering the effect of imperfect insulation and density.

In the practical environment (considering the real-time effects, namely imperfect insulation, density, and combination of imperfect insulation and density), the steady-state flow rates for the considered temperature setpoints will be different in comparison to the flow rates which were used previously. However, when we consider the temperature setpoint 32.5°C, the flow rates will exceed the limits (0.015 0.075). Therefore, in this chapter, the temperature setpoint 32.5°C was not considered.

By considering these steady-state flow rates using the individual effects, the performance of the controller tuned online using the EMFO algorithm in the practical environment was investigated for all the considered cases ([Table 5.3](#)), and compared with the controller tuned online using the original MFO algorithm in the same environment.

The system was further investigated for the effect of system dynamics (by considering the dead

time of 0.25 sec. and 0.5 sec.) and the effect of disturbance in the process (by considering an impulse function in the forward path in the process at $t=30$ sec.). The results obtained with MFO and EMFO algorithms in the practical environment were compared in terms of settling time T_s (sec.), as shown in Tables 5.3–5.5.

Table 5.3 Practical environment-based (with the effect of imperfect insulation) performances of SR-PID controller optimized online using MFO and EMFO algorithms.

Cases	Set points (°C)	Error	Without dead time		With a dead time of 0.25 sec.		With a dead time of 0.5 sec.		With disturbance	
			MFO	EMFO	MFO	EMFO	MFO	EMFO	MFO	EMFO
1	22.5	+3	137	111	157	123	191	140	135	109
2	22.5	+1	50	43	59	49	72	59	50	41
3	25	+3	190	147	223	165	265	189	189	145
4	25	+1	62	45	74	54	90	67	61	44
5	25	-1	166	98	185	115	237	133	176	104
6	25	-2	514	433	576	461	693	499	538	448
7	27.5	+3	324	255	368	282	443	311	322	250
8	27.5	+1	85	60	98	73	118	90	85	59
9	27.5	-1	77	54	87	64	104	80	83	57
10	27.5	-3	293	231	333	254	403	278	312	245
11	29	+3	665	568	721	605	939	654	664	563
12	29	+1	124	66	138	84	163	109	125	64
13	29	-1	61	46	72	53	85	63	66	48
14	29	-3	206	168	237	183	297	205	216	175

Table 5.4 Practical environment-based (with the effect of density) performances of SR-PID controller optimized online using MFO and EMFO algorithms.

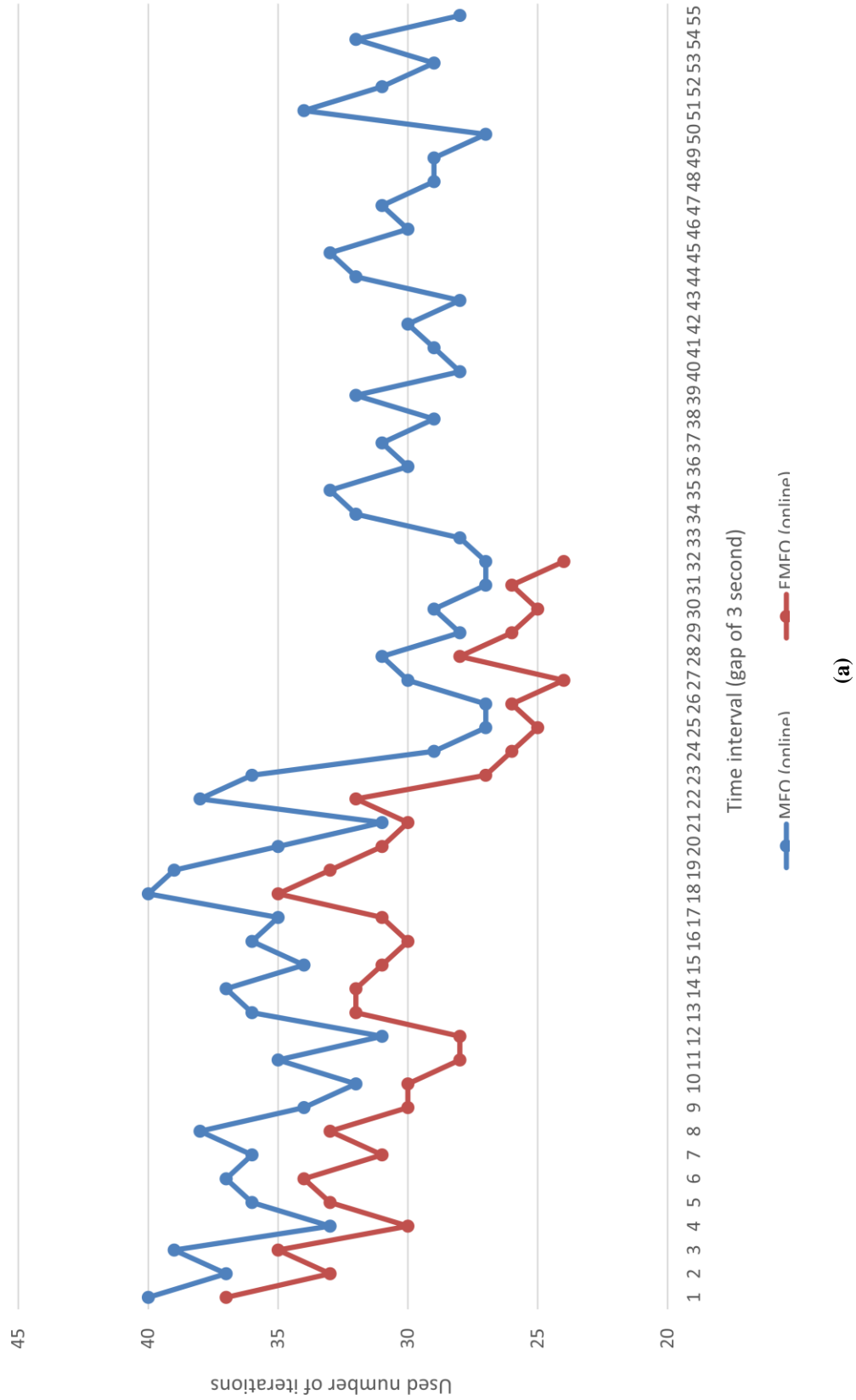
Cases	Set points (°C)	Error	Without dead time		With a dead time of 0.25 sec.		With a dead time of 0.5 sec.		With disturbance	
			MFO	EMFO	MFO	EMFO	MFO	EMFO	MFO	EMFO
1	22.5	+3	131	107	151	118	184	135	134	107
2	22.5	+1	48	41	57	47	70	57	50	40
3	25	+3	183	143	216	160	258	184	187	143
4	25	+1	59	43	72	52	88	65	61	43
5	25	-1	171	101	192	118	243	137	177	106
6	25	-2	525	441	586	470	702	509	540	450
7	27.5	+3	311	248	355	274	429	302	319	248
8	27.5	+1	81	58	95	70	115	87	83	58
9	27.5	-1	80	56	90	67	108	83	84	58
10	27.5	-3	306	240	347	263	417	287	313	246
11	29	+3	654	560	711	596	929	644	660	561
12	29	+1	120	64	134	81	159	105	125	63
13	29	-1	63	48	74	55	87	65	66	49
14	29	-3	211	173	244	189	305	211	216	177

Table 5.5 Practical environment-based (with the combined effect of imperfect insulation and density) performances of SR-PID controller optimized online using MFO and EMFO algorithms.

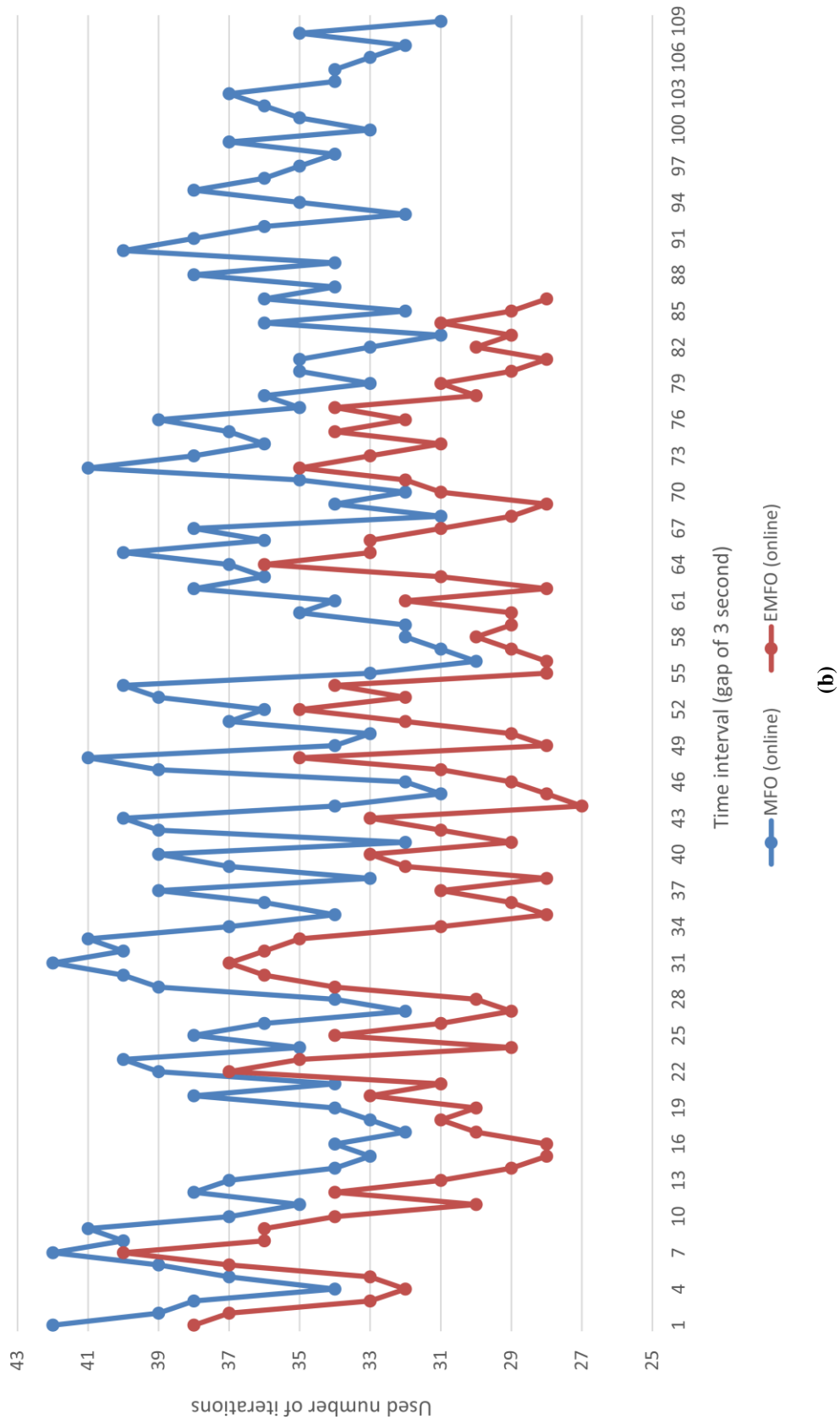
Cases	Set points (°C)	Error	Without dead time		With a dead time of 0.25 sec.		With a dead time of 0.5 sec.		With disturbance	
			MFO	EMFO	MFO	EMFO	MFO	EMFO	MFO	EMFO
1	22.5	+3	139	113	159	125	193	143	136	110
2	22.5	+1	51	44	60	50	73	60	51	41
3	25	+3	192	150	225	168	268	193	190	146
4	25	+1	63	46	75	55	92	69	62	44
5	25	-1	164	96	183	113	234	131	174	103
6	25	-2	510	428	573	456	690	494	535	445
7	27.5	+3	326	258	371	285	446	313	324	252
8	27.5	+1	86	61	99	74	120	92	86	60
9	27.5	-1	76	53	86	63	102	78	82	56
10	27.5	-3	291	229	331	251	401	275	310	243
11	29	+3	668	573	724	610	943	659	667	566
12	29	+1	125	68	140	86	165	112	127	65
13	29	-1	60	45	71	52	84	62	65	47
14	29	-3	204	166	235	181	295	202	215	174

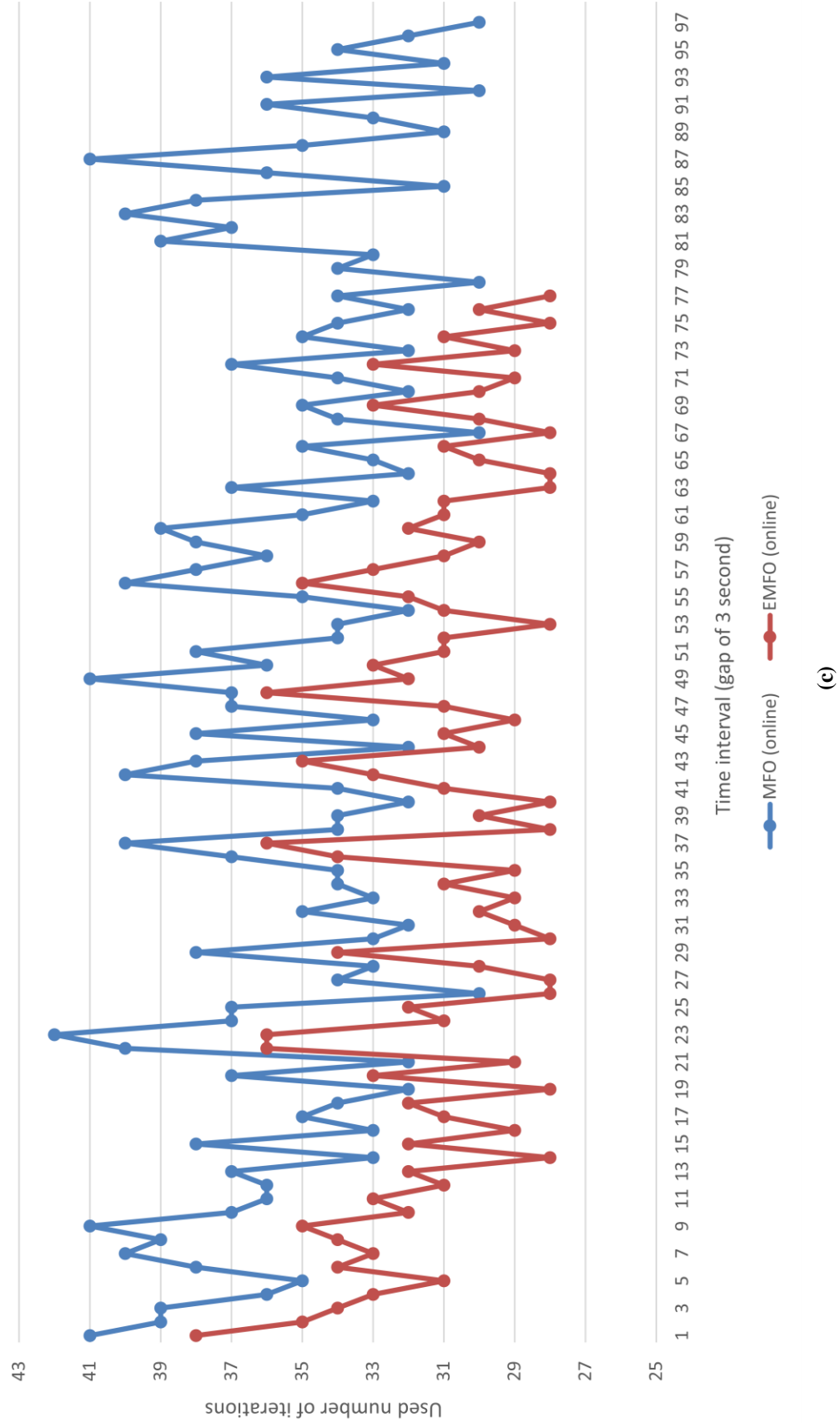
It is clear from the simulation results in [Tables 5.3–5.5](#) that even in the practical environment, the EMFO algorithm outperforms the original MFO algorithm in online tuning of the controller (in all the cases under consideration) because it combines the benefits of three modifications (change the spiral path, opposition learning-based initialization, and change in flames selection) in the original MFO algorithm. Due to this combination, the proposed EMFO algorithm has good initial search agents using a better spiral path to explore good-performing candidates in the search space, leading to the best solution with a faster convergence rate. This guarantees that appropriate controller gains are given to the controller at a definite time interval throughout the process cycle, resulting in a substantial decrease in settling time for the cases considered in [Tables 5.3–5.5](#).

In the simulated practical environment, the controller gains were updated after every three seconds. A comparison was also made between the EMFO and the original MFO algorithms on the basis of the number of iterations required after every three seconds to converge to the respective best solutions. The number of iterations required to converge after every three seconds by both the algorithms in Cases 5, Case 7, Case 10, and Case 12 (combined effect of imperfect insulation and density with zero dead time) is shown in [Fig. 5.4](#).



(a)





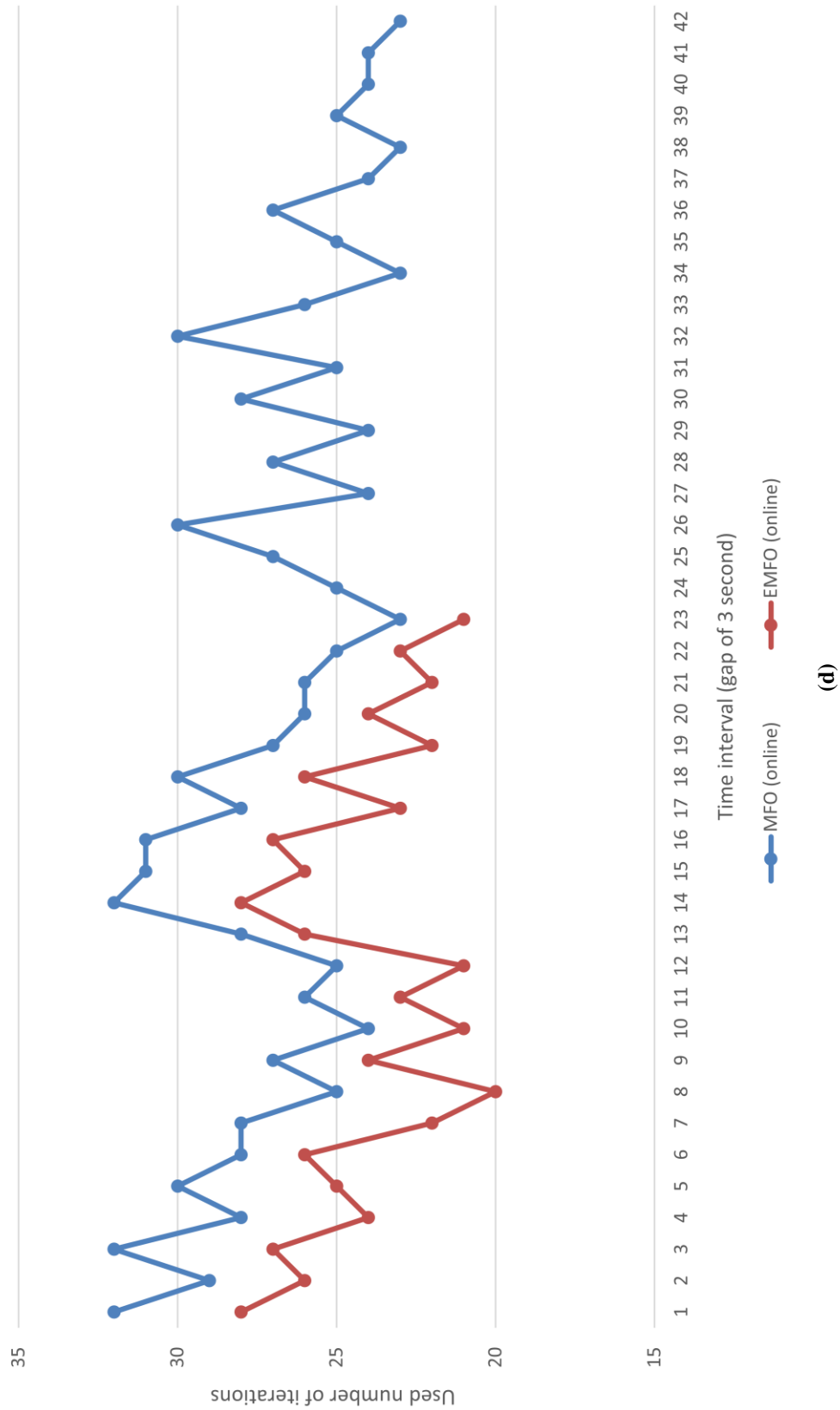


Fig. 5.4 Required number of iterations after every 3 seconds of time interval (in case of the combined effect) for the cases: (a) Case 5, (b) Case 7, (c) Case 10, (d) Case 12.

It can be observed from Fig. 5.4 that even in case of the practical environment, the EMFO algorithm converged to the optimal solution in fewer iterations in comparison to the original MFO algorithm for each updating. It is also clear that the updating of the controller gains in case of the EMFO algorithm stops much earlier as compared to the original MFO algorithm. This happens because the system settles to the new setpoint earlier in case of the EMFO algorithm due to better gain values.

The simulation results clearly establish the superiority of the EMFO algorithm in tuning the controller as compared to the original MFO algorithm even in the practical environment. On an average, improvement in EMFO over MFO was observed as 24.65% (with no dead time), 25.38% (with a dead time of 0.25 sec.), 28.64% (with a dead time of 0.5 sec.), and 26.07% (with disturbance) in case of the ideal environment (shown in Chapter 4), whereas in the practical environment, it has been observed as 24.56% (with no dead time), 25.15% (with a dead time of 0.25 sec.), 28.37% (with a dead time of 0.5 sec.), and 25.79% (with disturbance). Hence, the improvement observed due to the proposed EMFO algorithm over the original MFO algorithm is comparable in the ideal and practical environment. However, for the same optimization algorithm, the settling times were influenced by the environment.

Therefore, an investigation was made to study the effect of imperfect insulation, the effect of density, and the combined effect of imperfect insulation and density on the settling times. The settling times obtained from the simulation of the performance of the system in the practical environment were compared with results for the ideal environment (constant density=1 kg/l. and perfectly insulated tank).

5.2.1 Effect of imperfect insulation

A comparison of the settling times for the system with imperfect insulation and the ideal environment is presented in Table 5.6. It is clear from Table 5.6 that due to imperfect insulation, the settling times of the system have increased for the increasing setpoints and decreased for the decreasing setpoints. This definite pattern in the table is observed due to the heat losses ($\frac{T}{R}$ in Eq. 5.8) from the system on account of imperfect insulation. At the initial temperature (when the flow rates are at their steady-state values), there will be constant heat losses from the system.

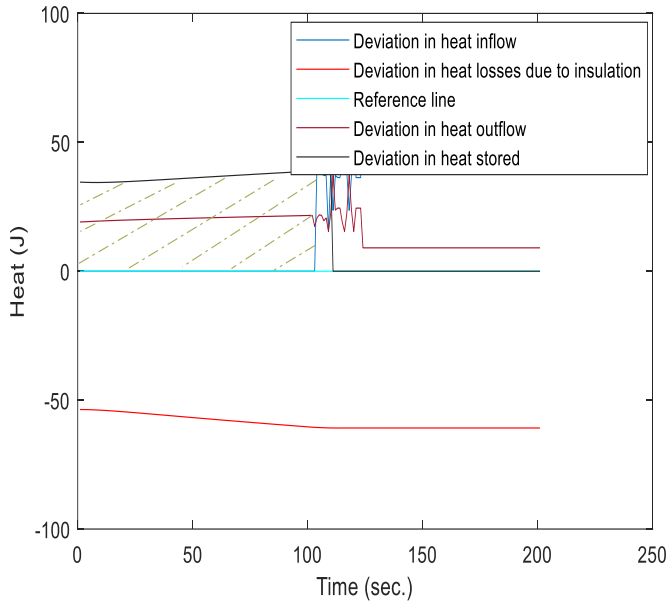
Table 5.6 Performances of SR-PID controller tuned online using EMFO algorithm in case of the ideal environment and the practical environment (with the effect of imperfect insulation).

Cases	Set points (°C)	Error	Without dead time		With a dead time of 0.25 sec.		With a dead time of 0.5 sec.		With disturbance	
			Ideal based	Practical based	Ideal based	Practical based	Ideal based	Practical based	Ideal based	Practical based
1	22.5	+3	104	111	115	123	131	140	106	109
2	22.5	+1	40	43	46	49	55	59	40	41
3	25	+3	139	147	156	165	179	189	141	145
4	25	+1	42	45	50	54	62	67	43	44
5	25	-1	105	98	123	115	143	133	108	104
6	25	-2	448	433	478	461	519	499	454	448
7	27.5	+3	242	255	267	282	293	311	245	250
8	27.5	+1	56	60	68	73	84	90	57	59
9	27.5	-1	58	54	69	64	86	80	59	57
10	27.5	-3	245	231	269	254	295	278	249	245
11	29	+3	552	568	587	605	633	654	557	563
12	29	+1	61	66	78	84	101	109	62	64
13	29	-1	49	46	57	53	68	63	50	48
14	29	-3	176	168	192	183	215	205	179	175

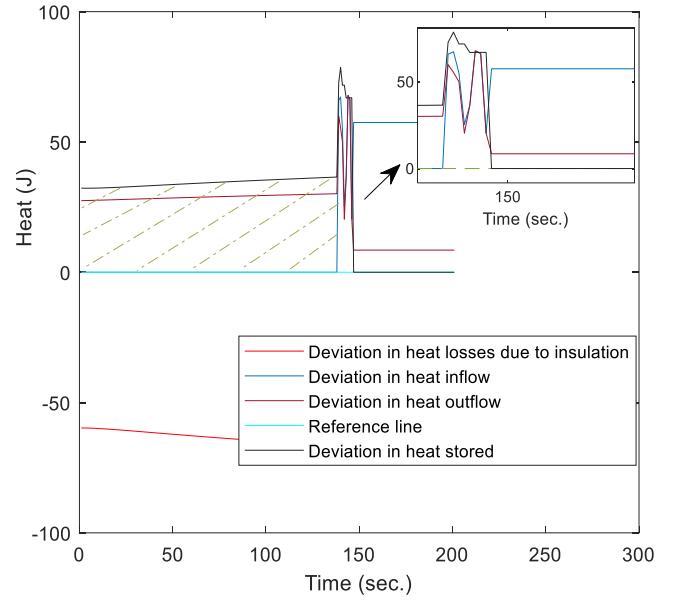
When the temperature is increased from its setpoint, heat losses ($\frac{T}{R}$) from the system will be increased. The heat outflow from the system will be decreased in comparison to the system with perfect insulation due to the slow rise in temperature, as per Eq. 5.8. This slow rise in temperature is because of heat losses. Although, the heat outflow has decreased, but because of the excessive heat losses due to imperfect insulation, the net heat accumulated into the system will be less as compared to the system with perfect insulation.

The graphs for deviation in heat inflow, deviation in heat outflow, deviation in heat losses due to insulation, and the deviation in heat stored are shown in Fig. 5.5 (considered the cases for a system with zero dead time). ‘Deviation in heat inflow’ represents the difference in heat inflows in cases of perfect insulation and imperfect insulation. In the beginning, the deviation in heat inflows is zero due to the same heat inflows in both cases. The same heat inflows are observed in the initial part in spite of the different controller outputs in the two cases as the magnitude of the controller outputs in both cases are such that the volume flow rates are restricted to saturation levels. After some time, the deviation in heat inflows exists due to the variation in the mass flow rates, as clearly shown in Fig. 5.5(b). ‘Deviation in heat outflow’ indicates the difference in heat outflows in the two cases (perfect insulation and imperfect insulation). It exists from the

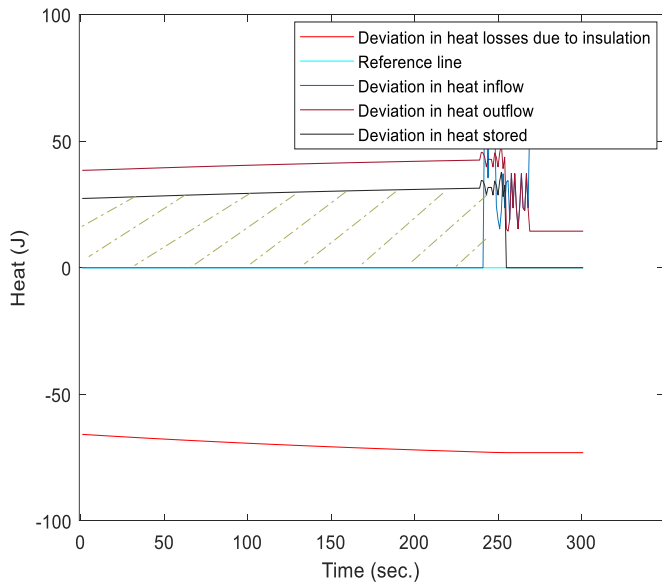
beginning due to the decrease in heat outflow in case of imperfect insulation. ‘Deviation in heat losses due to insulation’ represents the difference in heat losses in the system with perfect insulation (no heat loss) and the system with imperfect insulation.



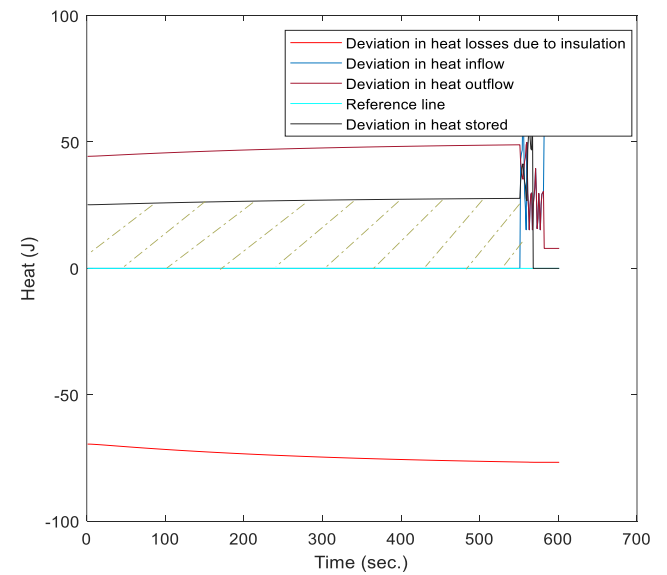
(a)



(b)



(c)



(d)

Fig. 5.5 Heat deficit in the case of the effect of imperfect insulation in various cases: (a) 22.5°C -25.5°C, (b) 25°C - 28°C, (c) 27.5°C -30.5°C, and (d) 29°C -32°C.

The deviations in heat inflows, heat outflows, and heat losses due to the insulation describe the quantity of heat deficit in the case of imperfect insulation. This heat deficit (shaded portion) in the system can be calculated as follows:

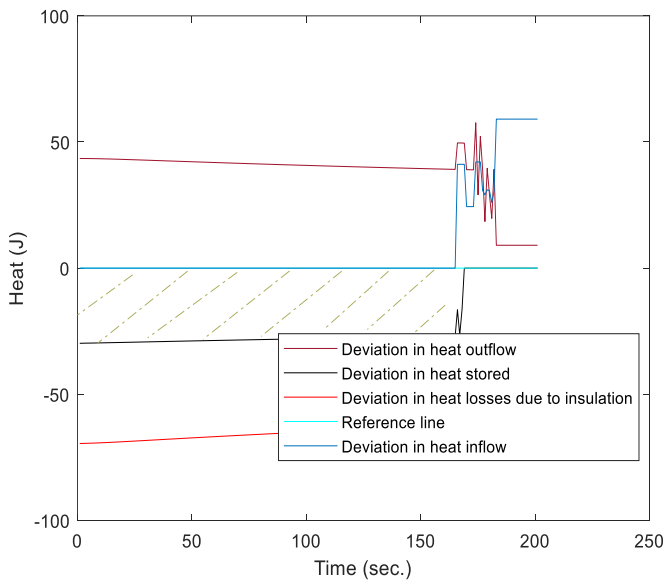
Heat deficit

$$= [(Area\ between\ the\ curves:\ deviation\ in\ heat\ inflows\ and\ deviation\ in\ heat\ outflows) - Area\ under\ the\ curve\ due\ to\ deviation\ in\ heat\ losses]$$

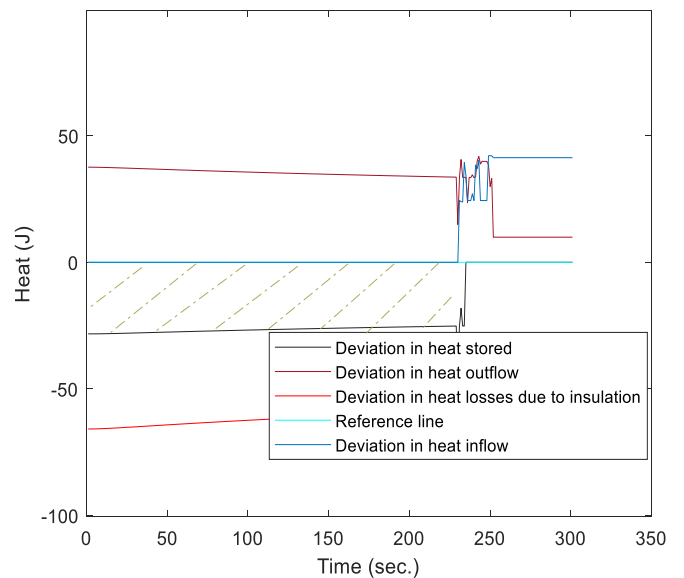
Further, the same heat will be required to be accumulated into the system for achieving the desired temperature setpoints. As a result, the system with imperfect insulation will take more time to reach desired temperature setpoints, as compared to the system with perfect insulation.

The heat deficit in various cases is shown in Fig. 5.5. It can be seen in the figure that the total heat deficit increases with an increase in temperature setpoints while considering the same value of error. As a result, the system will take more time to accumulate the required heat as shown in Table 5.6.

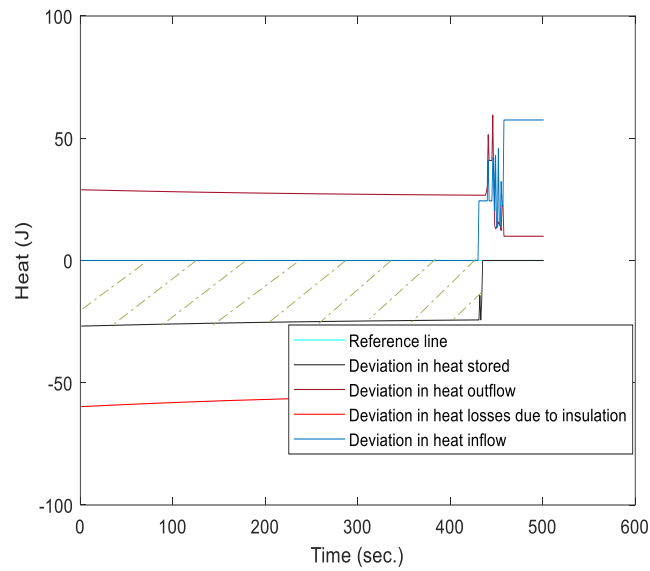
Similarly, for decreasing temperature setpoints, the deviations in heat inflows, heat outflows, and heat losses due to insulation are shown in Fig. 5.6.



(a)



(b)



(c)

Fig. 5.6. Heat accumulation in the case of the effect of imperfect insulation in various cases: (a) 29°C -26°C, (b) 27.5°C -24.5°C, and (c) 25°C -23°C.

The heat outflow from the system will be decreased. Due to decrement in the heat outflow, the system will store more heat as compared to the system with perfect insulation, but due to the excessive heat losses on account of imperfect insulation, the net heat accumulated into the system will be less as compared to the system with perfect insulation. Consequently, the time taken will be less as compared to the system with perfect insulation. Hence, the system with imperfect insulation will take less time to attain the desired temperature setpoint.

5.2.2 Effect of density

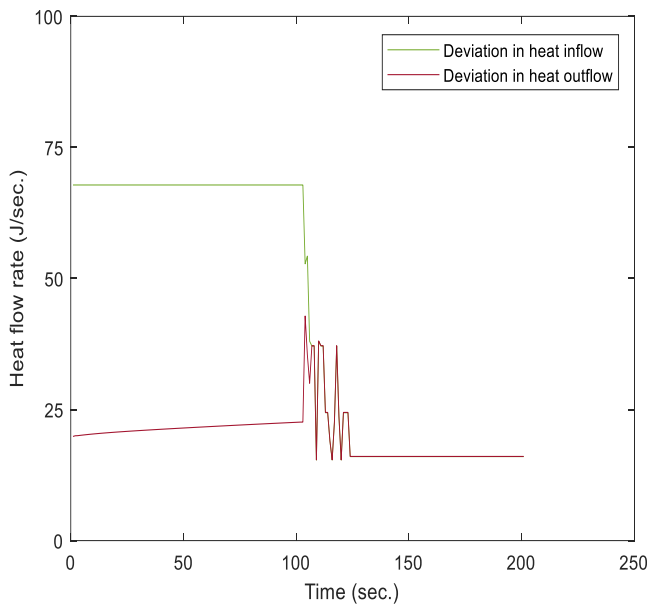
In contrast to the results obtained with the ideal environment assuming density (equal to 1), the settling times achieved from the practical model are found to be increased for increasing temperature setpoints and decreased for decreasing temperature setpoints, as shown in [Table 5.7](#). These variations in settling times are due to the effect of the water density in the practical model. A comparison of the performances obtained with the practical environment and the ideal environment is shown in [Table 5.7](#). As the temperature is raised above the setpoint, the controller first reduces the cold-water volume flow rate to the lower saturation limit and then increases the hot water volume flow rate (if necessary). This leads to a decrease in the cold-water mass flow rate and a rise in the hot water mass flow rate from its initial steady-state values, respectively.

Table 5.7 Performances of SR-PID controller tuned online using EMFO algorithm in case of the ideal environment and the practical environment (with the effect of density).

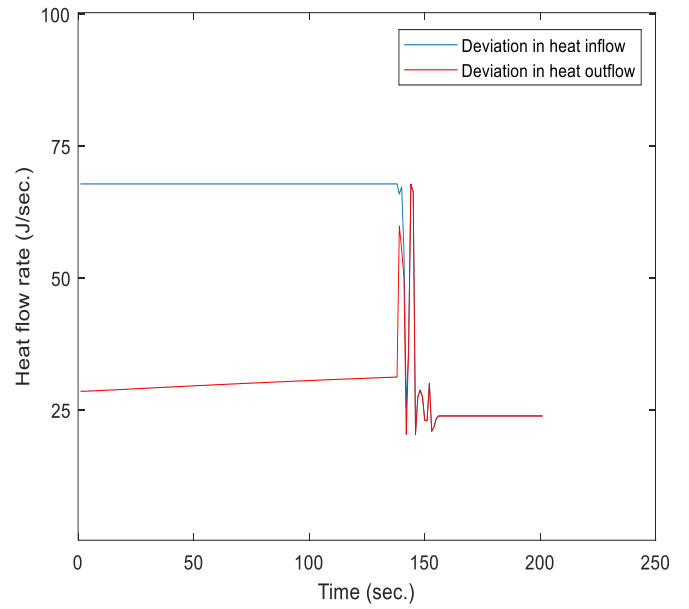
Cases	Set points (°C)	Error	Without dead time		With a dead time of 0.25 sec.		With a dead time of 0.5 sec.		With disturbance	
			Ideal based	Practical based	Ideal based	Practical based	Ideal based	Practical based	Ideal based	Practical based
1	22.5	+3	104	107	115	118	131	135	106	107
2	22.5	+1	40	41	46	47	55	57	40	40
3	25	+3	139	143	156	160	179	184	141	143
4	25	+1	42	43	50	52	62	65	43	43
5	25	-1	105	101	123	118	143	137	108	106
6	25	-2	448	441	478	470	519	509	454	450
7	27.5	+3	242	248	267	274	293	302	245	248
8	27.5	+1	56	58	68	70	84	87	57	58
9	27.5	-1	58	56	69	67	86	83	59	58
10	27.5	-3	245	240	269	263	295	287	249	246
11	29	+3	552	560	587	596	633	644	557	561
12	29	+1	61	64	78	81	101	105	62	63
13	29	-1	49	48	57	55	68	65	50	49
14	29	-3	176	173	192	189	215	211	179	177

Considering the effect of density in the practical model (the density of cold water and hot water is less than 1), it is observed from Eq. 5.7 that the total inlet mass flow rate due to cold water and hot water will be decreased, when volumetric flow rates are at their saturation limits. As the heat flow rate depends on the mass flow rate, the heat inflow (as per Eq. 5.7) will be lower as compared to the system assuming water density to be the same (equal to 1). Similarly, heat outflow from the system will be decreased. Due to the effect of density, the deviation in heat inflows is more than the deviation in heat outflows. As a result, the heat accumulated in the system would be less, as compared to the system assuming constant density (equal to 1).

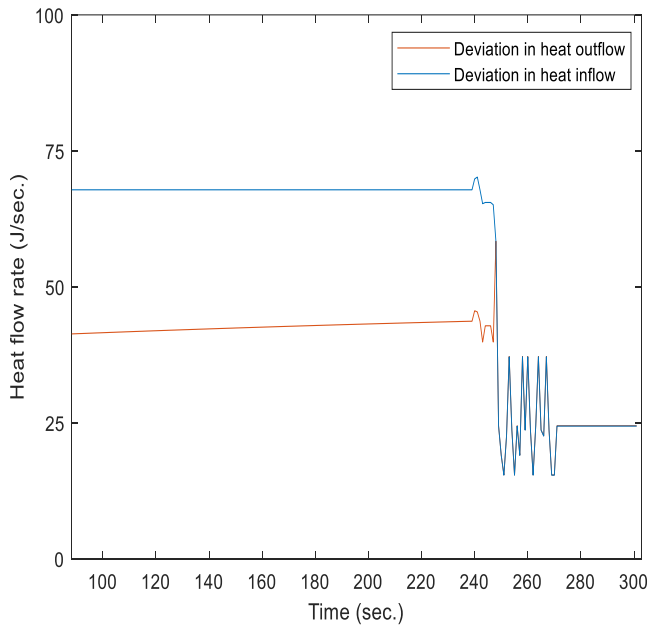
The graphs for deviations in heat inflows and heat outflows are shown in Fig. 5.7. The deviation indicates the difference in heat flows in cases of the system with constant density and the system with the effect of density. In Fig. 5.7, the area between the curves: the deviation in heat inflows, and the deviation in heat outflows describe the quantity of heat deficit in case of the effect of density. Further, the same heat will be required to be accumulated for achieving the desired temperature setpoints. As a result, the system with the effect of density will take more time to reach desired temperature setpoints, as compared to the system assuming constant water density.



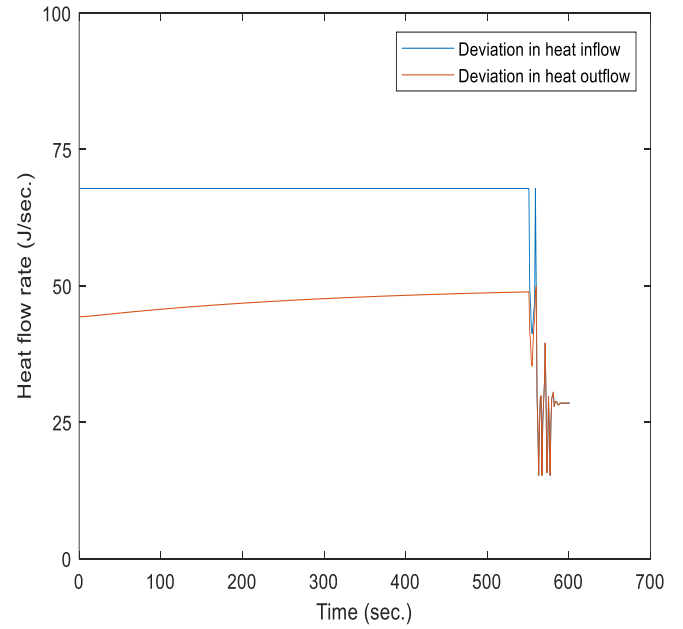
(a)



(b)



(c)



(d)

Fig. 5.7 Heat deficit in the case of the effect of density in various cases: (a) 22.5°C -25.5°C, (b) 25°C -28°C, (c) 27.5°C -30.5°C, and (d) 29°C -32°C.

The heat deficit in several cases is shown in Fig. 5.7. It is clear from the figure that the total heat deficit increases with an increase in temperature while considering the same value of error. As a

result, more heat will be required to be accumulated in the tank for the same, and the system will take more time while increasing the temperature setpoints for the same error value, as shown in Table 5.7.

Similarly, for decreasing temperature setpoints, the deviations in heat inflows and heat outflows are shown in Fig. 5.8.

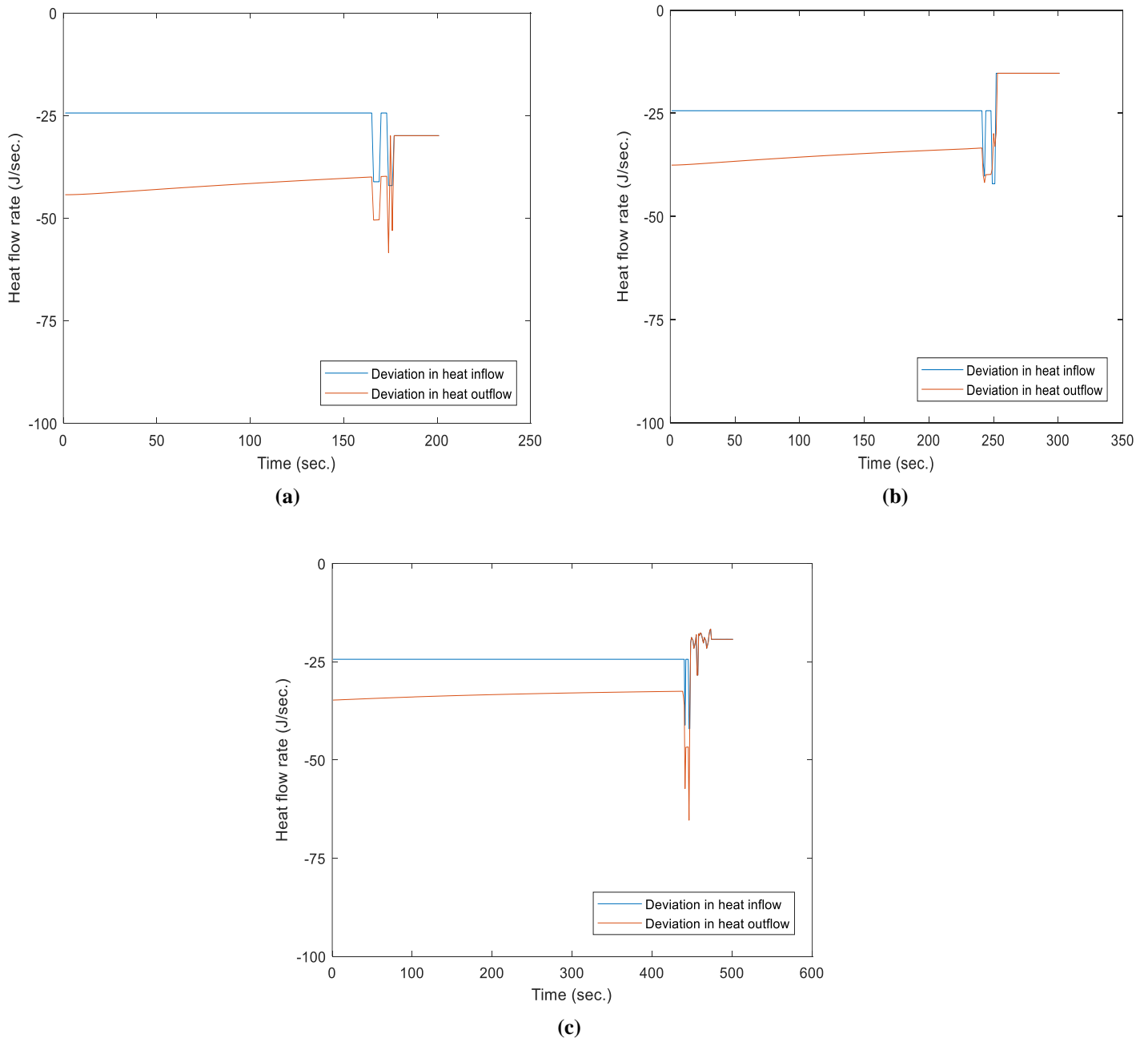


Fig. 5.8 Heat accumulation in the case of the effect of density in various cases: (a) 29°C -26°C, (b) 27.5°C -24.5°C, and (c) 25°C -23°C.

For decreasing temperature setpoints, the controller first decreases the hot water volume flow rate to the least level and then increases the cold-water volume flow rate (if necessary). By considering the effect of density, as per Eq. 5.7, the heat inflow and the heat outflow will be decreased in comparison to the system assuming constant density. The deviation in heat inflows is more than the deviation in heat outflows. This is because of the effect of density. As a result, the net heat accumulated in the system will be less as compared to the system with constant density, and for the same heat dissipation, the system will take less time to reach desired temperature setpoints, as shown in Table 5.7.

5.2.3 Effect of combination of imperfect insulation and density

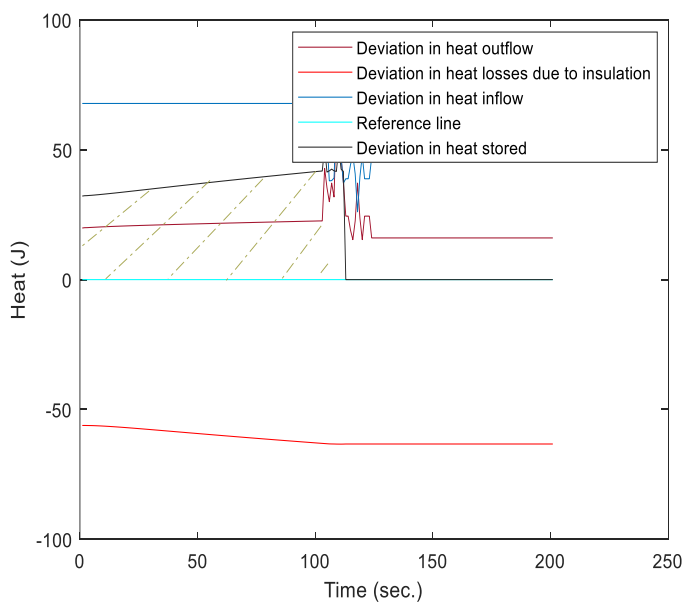
Due to the combined effect of imperfect insulation and density in the system, the settling times are found to be increased for increasing temperature setpoints and decreased for decreasing temperature setpoints. A comparative study of the settling times obtained for the system with the combined effect of imperfect insulation and density, and the system with the ideal environment, is shown in Table 5.8. Temperature affects the density of any fluid. The mass flow rate of water at the two inlets will be affected by a change in density, which will have an effect on the actual heat flowing into the system.

Table 5.8 Performances of SR-PID controller tuned online using EMFO algorithm in case of the ideal environment and the practical environment (with the combined effect of imperfect insulation and density).

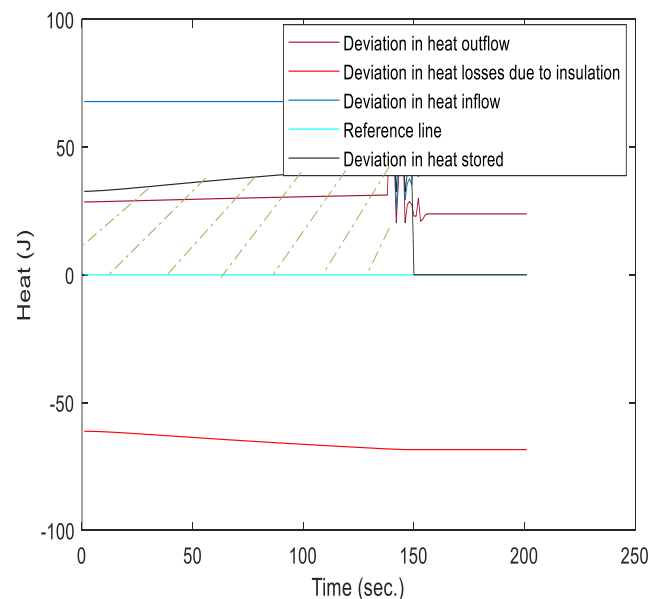
Cases	Set points (°C)	Error	Without dead time		With a dead time of 0.25 sec.		With a dead time of 0.5 sec.		With disturbance	
			Ideal based	Practical based	Ideal based	Practical based	Ideal based	Practical based	Ideal based	Practical based
1	22.5	+3	104	113	115	125	131	143	106	110
2	22.5	+1	40	44	46	50	55	60	40	41
3	25	+3	139	150	156	168	179	193	141	146
4	25	+1	42	46	50	55	62	69	43	44
5	25	-1	105	96	123	113	143	131	108	103
6	25	-2	448	428	478	456	519	494	454	445
7	27.5	+3	242	258	267	285	293	313	245	252
8	27.5	+1	56	61	68	74	84	92	57	60
9	27.5	-1	58	53	69	63	86	78	59	56
10	27.5	-3	245	229	269	251	295	275	249	243
11	29	+3	552	573	587	610	633	659	557	566
12	29	+1	61	68	78	86	101	112	62	65
13	29	-1	49	45	57	52	68	62	50	47
14	29	-3	176	166	192	181	215	202	179	174

When the setpoints are to be increased from the initial temperature, the controller first decreases the volume flow rate of the cold water to the minimum value and then if required, increases the volume flow rate of the hot water. This will result in a decrease in the mass flow rate of the cold water and an increase in the mass flow rate of the hot water, respectively. As the density of cold water is more than that of hot water, the rate of contribution of heat inflows due to the cold water and hot water will be different. The total heat inflow (as per Eq. 5.9) will be lower due to the effect of density in the combined effect as compared to the system with constant water density (equal to 1 kg/l.) and perfect insulation. Similarly, the heat outflow (as per Eq. 5.9) from the system will be decreased due to the effect of density and the slow rise in temperature (due to heat loss). The density for the outflow is fixed as per the initial temperature. The heat losses from the system due to the imperfect insulation will be increased by increasing the temperature from its setpoint. Due to decrement in the heat inflow and the heat outflow, the system ideally should store heat, but because of heat losses due to imperfect insulation, the net heat accumulated into the system decreases as compared to the system with constant density and perfect insulation.

The graphs for the deviations in heat inflows, heat outflows, and heat losses due to insulation are shown in Fig. 5.9, which describe the quantity of heat deficit (shaded portion) in case of the combined effect of imperfect insulation and density.



(a)



(b)

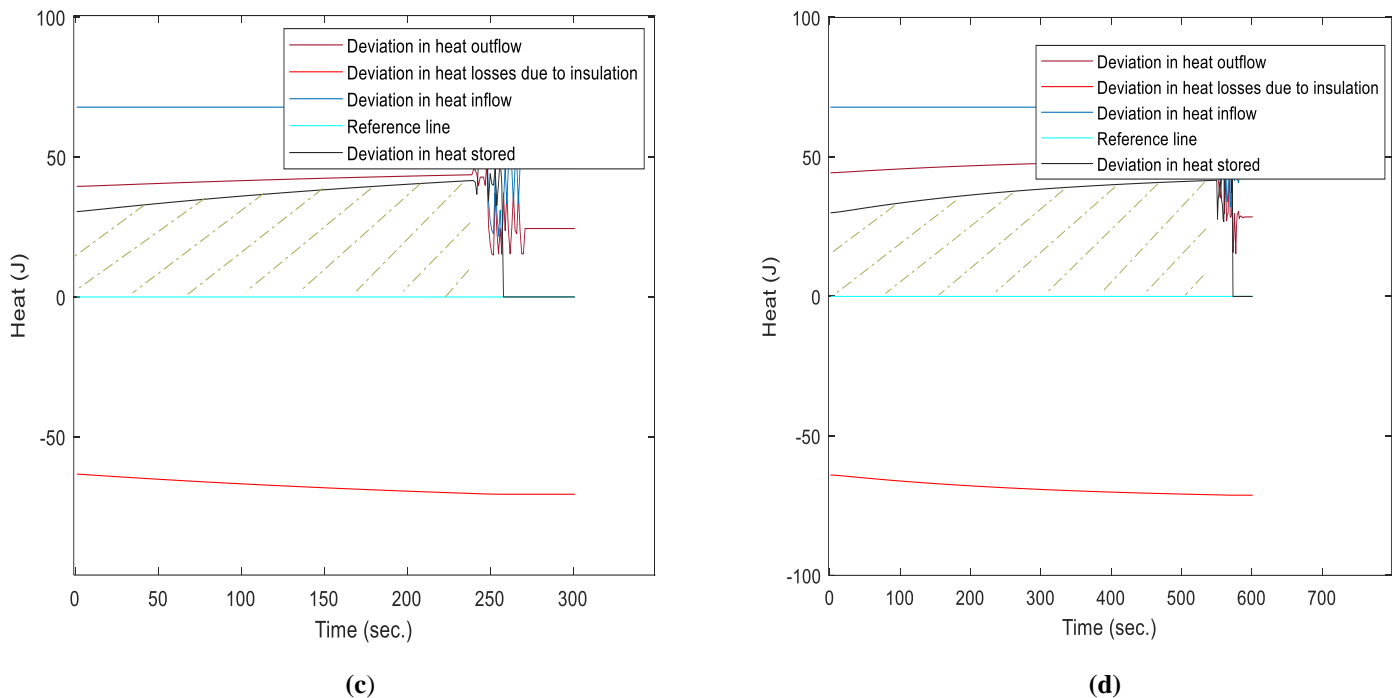


Fig. 5.9 Heat deficit in the case of the combined effect in various cases: (a) 22.5°C -25.5°C, (b) 25°C -28°C, (c) 27.5°C -30.5°C, and (d) 29°C -32°C.

The heat deficit can be calculated as follow:

Heat deficit

$$= [(Area\ between\ the\ curves:\ deviation\ in\ heat\ inflows\ and\ deviation\ in\ heat\ outflows) \\ - Area\ under\ the\ curve\ due\ to\ deviation\ in\ heat\ losses]$$

As a result, the system with the combined effect will take more time to acquire the same heat, as compared to the system with constant density and perfect insulation.

Fig. 5.9 shows the deviations between the practical (system with the combined effect of imperfect insulation and density) and an ideal (system with constant density and perfect insulation). It is observed from Fig. 5.9 that the total heat deficit increases for same desired rise in temperature as one moves from the lower to higher limit of the working range of temperature. It is because of the effect of density (different densities being considered), and the slow rise in temperature. As a result, the system will take more time to store the required heat while increasing the temperature setpoints for the same value of error, as shown in Table 5.8.

In the same way, for decreasing temperature setpoints (shown in Fig. 5.10), less heat will be accumulated as compared to the system with constant density and perfect insulation. It is because of heat losses, a decrease in heat inflow, and a decrease in heat outflow.

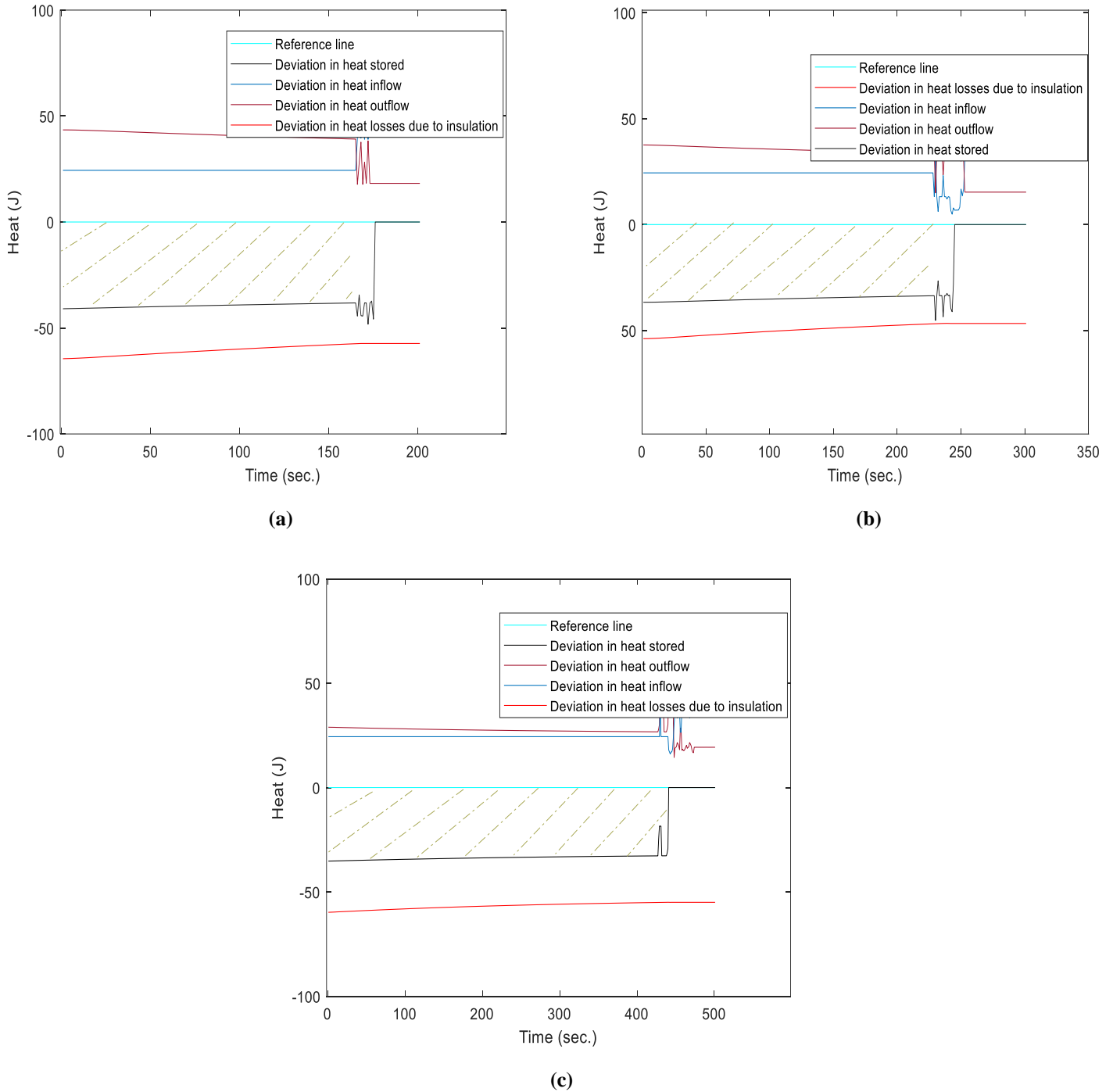


Fig. 5.10 Heat accumulation in the case of the combined effect in various cases: (a) 29°C -26°C, (b) 27.5°C - 24.5°C, and (c) 25°C -23°C.

Thus, the same heat will be dissipated at a faster rate from the system for obtaining the desired temperature setpoints. As a result, the system with the combined effect of imperfect insulation and density will take less time to reach desired temperature setpoints, as compared to the system with constant density and perfect insulation.

Moreover, the investigation of the controller performance is also carried out on the basis of the utility consumptions (u_c and u_h) for the above-considered setpoints in all the circumstances (the effect of imperfect insulation, the effect of density, and the combined effect of imperfect insulation and density). The utility consumptions obtained using the controller tuned online with the EMFO algorithm are shown in Tables 5.9–5.14. These tables provide a comparison of the performance of the proposed EMFO algorithm in the ideal environment and the practical environment.

Table 5.9 Comparative study of the amount of flows using SR-PID controller tuned online with EMFO algorithm in case of the ideal environment and the practical environment (with the effect of imperfect insulation).

Cases	Set points (°C)	Error	Without dead time				With a dead time of 0.25 sec.				With a dead time of 0.5 sec.			
			Ideal based		Practical based		Ideal based		Practical based		Ideal based		Practical based	
			u_c	u_h	u_c	u_h	u_c	u_h	u_c	u_h	u_c	u_h	u_c	u_h
1	22.5	+3	3.54	8.87	3.79	9.58	3.72	9.84	4.02	10.7	3.96	11.1	4.32	12.2
2	22.5	+1	3.16	4.42	3.38	4.78	3.32	4.91	3.59	5.35	3.54	5.52	3.87	6.08
3	25	+3	3.80	12.6	4.07	13.7	3.99	13.9	4.31	15.2	4.25	15.7	4.64	17.3
4	25	+1	3.68	4.77	3.94	5.15	3.86	5.29	4.17	5.77	4.11	5.96	4.49	6.57
5	25	-1	10.6	4.65	9.87	4.33	11.1	5.16	10.3	4.75	11.8	5.81	10.8	5.3
6	25	-2	33.1	6.08	30.9	5.65	34.8	6.75	32.2	6.22	37.1	7.61	33.9	6.95
7	27.5	+3	4.38	18.4	4.69	19.8	4.61	20.4	4.99	22.2	4.91	23.0	5.36	25.2
8	27.5	+1	4.09	6.08	4.38	6.54	4.29	6.75	4.64	7.33	4.57	7.61	4.99	8.34
9	27.5	-1	6.02	4.03	5.6	3.74	6.32	4.47	5.81	4.11	6.74	5.03	6.14	4.59
10	27.5	-3	18.3	4.08	17.1	3.79	19.2	4.53	17.7	4.17	20.5	5.11	18.7	4.66
11	29	+3	8.81	40.9	9.44	43.9	9.25	45.4	10.0	49.2	9.86	51.2	10.8	56.1
12	29	+1	4.62	8.93	4.95	9.6	4.85	9.91	5.25	10.8	5.17	11.2	5.65	12.2
13	29	-1	4.76	3.51	4.45	3.26	5.01	3.89	4.63	3.58	5.33	4.38	4.87	3.99
14	29	-3	13.7	3.69	12.8	3.43	14.4	4.09	13.3	3.76	15.4	4.61	14.1	4.2

Table 5.10 Comparative study of the amount of flows (in case of disturbance) using SR-PID controller tuned online with EMFO algorithm in case of the ideal environment and the practical environment (with the effect of imperfect insulation).

Cases	Set points (°C)	Error	With disturbance			
			Ideal based		Practical based	
			u_c	u_h	u_c	u_h
1	22.5	+3	3.62	9.14	3.69	9.31
2	22.5	+1	3.17	4.48	3.22	4.55
3	25	+3	3.88	12.95	3.95	13.18
4	25	+1	3.76	4.92	3.83	5.01
5	25	-1	10.81	4.79	10.63	4.7
6	25	-2	33.85	6.27	33.27	6.16
7	27.5	+3	4.47	18.98	4.55	19.3
8	27.5	+1	4.18	6.27	4.25	6.38
9	27.5	-1	6.15	4.15	6.04	4.08
10	27.5	-3	18.69	4.21	18.36	4.14
11	29	+3	9.01	42.23	9.17	42.95
12	29	+1	4.72	9.21	4.81	9.35
13	29	-1	4.86	3.62	4.77	3.56
14	29	-3	14.01	3.81	13.76	3.74

Table 5.11 Comparative study of the amount of flows using SR-PID controller tuned online with EMFO algorithm in case of the ideal environment and the practical environment (with the effect of density).

Cases	Set points (°C)	Error	Without dead time				With a dead time of 0.25 sec.				With a dead time of 0.5 sec.			
			Ideal based		Practical based		Ideal based		Practical based		Ideal based		Practical based	
			u_c	u_h	u_c	u_h	u_c	u_h	u_c	u_h	u_c	u_h	u_c	u_h
1	22.5	+3	3.54	8.87	3.6	9.04	3.72	9.84	3.82	10.1	3.96	11.1	4.1	11.5
2	22.5	+1	3.16	4.42	3.21	4.51	3.32	4.91	3.41	5.05	3.54	5.52	3.67	5.74
3	25	+3	3.80	12.6	3.86	12.9	3.99	13.9	4.09	14.3	4.25	15.7	4.4	16.3
4	25	+1	3.68	4.77	3.74	4.86	3.86	5.29	3.96	5.45	4.11	5.96	4.26	6.2
5	25	-1	10.6	4.65	10.4	4.56	11.1	5.16	10.8	5.01	11.8	5.81	11.4	5.59
6	25	-2	33.1	6.08	32.3	5.91	34.8	6.75	33.9	6.56	37.1	7.61	35.8	7.32
7	27.5	+3	4.38	18.4	4.45	18.8	4.61	20.4	4.73	21.1	4.91	23.0	5.09	23.9
8	27.5	+1	4.09	6.08	4.16	6.2	4.29	6.75	4.4	6.95	4.57	7.61	4.73	7.91
9	27.5	-1	6.02	4.03	5.93	3.95	6.32	4.47	6.16	4.34	6.74	5.03	6.51	4.84
10	27.5	-3	18.3	4.08	18.1	4.1	19.2	4.53	18.7	4.4	20.5	5.11	19.8	4.92
11	29	+3	8.81	40.9	8.95	41.7	9.25	45.4	9.49	46.7	9.86	51.2	10.2	53.2
12	29	+1	4.62	8.93	4.69	9.1	4.85	9.91	4.98	10.2	5.17	11.2	5.36	11.6
13	29	-1	4.76	3.51	4.69	3.44	5.01	3.89	4.88	3.78	5.33	4.38	5.14	4.21
14	29	-3	13.7	3.69	13.5	3.62	14.4	4.09	14.0	3.97	15.4	4.61	14.9	4.43

Table 5.12 Comparative study of the amount of flows (in case of disturbance) using SR-PID controller tuned online with EMFO algorithm in case of the ideal environment and the practical environment (with the effect of density).

Cases	Set points (°C)	Error	With disturbance			
			Ideal based		Practical based	
			u_c	u_h	u_c	u_h
1	22.5	+3	3.62	9.14	3.65	9.21
2	22.5	+1	3.17	4.48	3.19	4.51
3	25	+3	3.88	12.95	3.91	13.13
4	25	+1	3.76	4.92	3.78	4.96
5	25	-1	10.81	4.79	10.71	4.73
6	25	-2	33.85	6.27	33.49	6.18
7	27.5	+3	4.47	18.98	4.51	19.12
8	27.5	+1	4.18	6.27	4.21	6.32
9	27.5	-1	6.15	4.15	6.1	4.12
10	27.5	-3	18.69	4.21	18.54	4.15
11	29	+3	9.01	42.23	9.14	42.65
12	29	+1	4.72	9.21	4.76	9.28
13	29	-1	4.86	3.62	4.81	3.58
14	29	-3	14.01	3.81	13.9	3.76

Table 5.13 Comparative study of the amount of flows using SR-PID controller tuned online with EMFO algorithm in case of the ideal environment and the practical environment (with the combined effect of imperfect insulation and density).

Cases	Set points (°C)	Error	Without dead time				With a dead time of 0.25 sec.				With a dead time of 0.5 sec.			
			Ideal based		Practical based		Ideal based		Practical based		Ideal based		Practical based	
			u_c	u_h	u_c	u_h	u_c	u_h	u_c	u_h	u_c	u_h	u_c	u_h
1	22.5	+3	3.54	8.87	3.86	9.77	3.72	9.84	4.1	10.9	3.96	11.1	4.4	12.4
2	22.5	+1	3.16	4.42	3.45	4.87	3.32	4.91	3.66	5.45	3.54	5.52	3.95	6.2
3	25	+3	3.80	12.6	4.15	13.9	3.99	13.9	4.39	15.5	4.25	15.7	4.73	17.6
4	25	+1	3.68	4.77	4.02	5.25	3.86	5.29	4.25	5.88	4.11	5.96	4.58	6.7
5	25	-1	10.6	4.65	9.68	4.25	11.1	5.16	10.1	4.66	11.8	5.81	10.6	5.2
6	25	-2	33.1	6.08	30.3	5.54	34.8	6.75	31.6	6.1	37.1	7.61	33.2	6.81
7	27.5	+3	4.38	18.4	4.78	20.2	4.61	20.4	5.09	22.6	4.91	23.0	5.47	25.7
8	27.5	+1	4.09	6.08	4.47	6.67	4.29	6.75	4.73	7.48	4.57	7.61	5.09	8.51
9	27.5	-1	6.02	4.03	5.49	3.67	6.32	4.47	5.7	4.03	6.74	5.03	6.02	4.5
10	27.5	-3	18.3	4.08	16.8	3.72	19.2	4.53	17.4	4.09	20.5	5.11	18.3	4.57
11	29	+3	8.81	40.9	9.63	44.8	9.25	45.4	10.2	50.2	9.86	51.2	11.0	57.2
12	29	+1	4.62	8.93	5.05	9.79	4.85	9.91	5.35	11.0	5.17	11.2	5.76	12.4
13	29	-1	4.76	3.51	4.36	3.2	5.01	3.89	4.54	3.51	5.33	4.38	4.77	3.91
14	29	-3	13.7	3.69	12.6	3.36	14.4	4.09	13.0	3.69	15.4	4.61	13.8	4.12

Table 5.14 Comparative study of the amount of flows (in case of disturbance) using SR-PID controller tuned online with EMFO algorithm in case of the ideal environment and the practical environment (with the combined effect of imperfect insulation and density).

Cases	Set points (°C)	Error	With disturbance			
			Ideal based		Practical based	
			u_c	u_h	u_c	u_h
1	22.5	+3	3.62	9.14	3.71	9.35
2	22.5	+1	3.17	4.48	3.24	4.57
3	25	+3	3.88	12.95	3.98	13.28
4	25	+1	3.76	4.92	3.84	5.03
5	25	-1	10.81	4.79	10.55	4.66
6	25	-2	33.85	6.27	33.02	6.11
7	27.5	+3	4.47	18.98	4.59	19.44
8	27.5	+1	4.18	6.27	4.28	6.43
9	27.5	-1	6.15	4.15	5.99	4.05
10	27.5	-3	18.69	4.21	18.21	4.11
11	29	+3	9.01	42.23	9.24	43.27
12	29	+1	4.72	9.21	4.85	9.44
13	29	-1	4.86	3.62	4.73	3.53
14	29	-3	14.01	3.81	13.65	3.71

It can be seen from [Tables 5.9–5.14](#) that when we move downward in the working range of temperature, the consumption of cold and hot water utilities (u_h and u_c) was found to be lesser in case of the practical environment as compared to the ideal environment. This is because of the individual effects of imperfect insulation, density, and combination of both. Similarly, the consumption of the utilities was found to be higher in case of the practical environment as we move upward in the working range of temperature.

The simulation results also show that even in case of the practical environment, when we move upward/ downward in the working range of temperature, hot/ cold utility consumption increases for the same desired rise/ fall in the temperature. From [Tables 5.9, 5.11, and 5.13](#), it can also be observed that the hot and cold utility consumptions increase as the dead time in the valve increases. Even in case of disturbance, these utility consumptions increase in all the considered cases ([Tables 5.10, 5.12, and 5.14](#)).

This chapter investigated the performance of the MFO and EMFO algorithm for continuously online updating of the gain parameters of the SR-PID controller in the simulated real environment. To validate the simulation results obtained for an ideal environment, the real environment of the practical system was simulated by making the electrical analogous model,

incorporating several effects considered in practical situations. A comparative study was made between the performances of the EMFO algorithm and the original MFO algorithm-based controller on the basis of settling time. Further, the system was also investigated for the effect of system dynamics and the process disturbance. The results showed the superiority of the EMFO algorithm in the online tuning of the controller in comparison to the MFO algorithm in the practical environment. As a future scope, this study can be validated on the practical system. The work can also be extended by introducing new changes in the MFO algorithm, or by hybridization of the MFO algorithm with other nature-inspired algorithms.

Conclusion and Future Scope

This chapter summarizes the research outcomes and the significant contributions of this dissertation. It also provides the future scope for improvement of the current research work.

6.1 Conclusion

The current work highlighted the conventional PID controller, a standard split range PID controller, and the variable range of split range PID controller for temperature control of a mixing process. Initially, the performance of all the control schemes was investigated using a classical Z-N tuning method. The simulation results obtained were compared in terms of settling time and steady-state error. It was found that the variable range of split range PID controller outperformed conventional PID controller as well as a standard split range PID controller in all the scenarios, i.e., the effect of dead time in the valve, the effect of disturbance in the process, and utility consumption. The steady-state error in the case of the variable range SR-PID controller was found to be minimum (equivalent to zero) in all the scenarios, as shown in Chapter 2. However, the variable range SR-PID controller with the Z-N tuning method produced a response with massive overshoot and a large settling time.

In order to improve the performance of the variable range SR-PID controller, this work used different nature-inspired optimization techniques such as PSO, WOA, and MFO for tuning (offline) the controller gains. The simulation results obtained were compared in terms of settling time. Based on these simulation results, a comparative study was made for the controller performance using the Z-N method, PSO, WOA, and MFO algorithms on the basis of settling time. It was observed that the MFO algorithm performed better as compared to all the other nature-inspired algorithms used and the Z-N classical method, in all the scenarios, i.e., the effect of dead time in the valve, the effect of disturbance in the process, and utility consumption.

To further explore the solution space and enhance the performance of the system, various improved versions of the moth flame optimization algorithm (change in a spiral path, initial population, and flame selection) were proposed for the same. A new version of the MFO (EMFO) algorithm was obtained by combining all the three phases of modification and used to tune the SR-PID controller in offline mode. To show the efficacy of the proposed variants of the MFO algorithm, the performance of the controller using the proposed algorithms was investigated for the different temperature setpoints within the working range, and compared with the controller performance using the original MFO algorithm in terms of settling time. It was observed that the performance with the proposed improved MFO algorithms was better. The results demonstrated that the controller using the EMFO algorithm outperformed the original MFO algorithm in all the scenarios.

Further, an online tuning approach used in this work yielded a better performance as compared to the offline strategy in terms of settling time. Usage of the online tuning approach along with the enhanced MFO algorithm further improved the results.

Furthermore, the performance of the MFO and EMFO algorithm was investigated in the real environment while the SR-PID controller gain parameters were updated continuously online using the same algorithms. To validate the simulation results obtained in an ideal environment, the real environment of the practical system was simulated by making the electrical analogous model incorporating several effects considered in practical situations. A comparison was made between the performances of the EMFO algorithm- and the original MFO algorithm-based controller in terms of settling time. The results showed the superiority of the EMFO algorithm in the online tuning of the controller as compared to the original MFO algorithm in the practical environment. Moreover, the system was also investigated with the effect of system dynamics, process disturbance, and utility consumption.

6.2 Future Scope

The future scope of the current research work is described below:

- ❖ The current work can be extended by considering disturbances in the various sensors that are normally used in the feedback path. Also, the effect of the nature of disturbances entering into the process can be investigated.

- ❖ The overall working range of a variable split range controller used in the present work can be further subdivided in case, more manipulated variables exist in the system. The performance of enhanced MFO algorithms in such systems can be a further area of investigation.
 - ❖ The modified MFO algorithm developed in the work can be used for further performance enhancement in many real-world engineering problems.
 - ❖ This research work can be validated on the practical system.
 - ❖ This work can also be extended by incorporating new modifications in the MFO algorithm, or by the hybridization of the MFO algorithm with other nature-inspired algorithms for subsequent usage in various applications.
-

Research Outcomes

List of Published Research Papers

1. Vishnoi, V., Tiwari, S., & Singla, R. (2022). Performance Investigation of EMFO-Based Perpetual Online Tuned Variable SR-Controller in Simulated Real Environment. *Cybernetics and Systems*, 1-25. **(Taylor & Francis, SCIE indexed) (Published)**
2. Vishnoi, V., Tiwari, S., & Singla, R. (2021). Performance Analysis of Enhanced MFO-Based Online-Tuned Split-Range PID Controller. *Arabian Journal for Science and Engineering*, 46, 9673–9689. **(Springer, SCIE indexed) (Published)**
3. Vishnoi, V., Tiwari, S., & Singla, R. (2021). Performance analysis of moth flame optimization-based split-range PID controller. *MAPAN*, 36(1), 67-79. **(Springer, SCIE indexed) (Published)**
4. Vishnoi, V., Tiwari, S., & Singla, R. (2021). Controller Design for Temperature Control of MISO Water Tank System: Simulation Studies. *International Journal of Cognitive Informatics and Natural Intelligence (IJCINI)*, 15(4), 1-13. **(IGI Global, Scopus and ESCI indexed) (Published)**

References

- Abdel-Basset, M., Manogaran, G., El-Shahat, D., & Mirjalili, S. (2018). A hybrid whale optimization algorithm based on local search strategy for the permutation flow shop scheduling problem. *Future Generation Computer Systems*, 85, 129–145. <https://doi.org/10.1016/j.future.2018.03.020>
- Acharyulu, B. V. S., Mohanty, B., & Hota, P. K. (2020). Analysis of moth flame optimization optimized cascade proportional-integral-proportional-derivative controller with filter for automatic generation control system incorporating solar thermal power plant. *Optimal Control Applications and Methods*, 41(3), 866–881. <https://doi.org/10.1002/oca.2582>
- Adam, E. J., & Marchetti, J. L. (2004). Designing and tuning robust feedforward controllers. *Computers and Chemical Engineering*, 28(9). <https://doi.org/10.1016/j.compchemeng.2004.03.005>
- Ahn, C. W., An, J., & Yoo, J. C. (2012). Estimation of particle swarm distribution algorithms: Combining the benefits of PSO and EDAs. *Information Sciences*, 192, 109–119.
- Aleksiejuk, J., Chochowski, A., & Reshetiuk, V. (2018). Analog model of dynamics of a flat-plate solar collector. *Solar Energy*, 160, 103–116. <https://doi.org/10.1016/j.solener.2017.11.079>
- Ali, M. M., Adewumi, A. O., Blamah, N., & Falowo, O. (2015). Mathematical modeling and optimization of industrial problems. *Journal of Applied Mathematics*, 2015.
- Ali, M., & Ahn, C. W. (2015). Comments on “Optimized gray-scale image watermarking using DWT-SVD and firefly algorithm”. *Expert Systems with Applications*, 42(5), 2392–2394.
- Alibakhshkenari, M., Virdee, B. S., Shukla, P., See, C. H., Abd-Alhameed, R. A., Falcone, F., & Limiti, E. (2020). Improved adaptive impedance matching for RF front-end systems of wireless transceivers. *Scientific Reports*, 10(1), 1–11.
- Allam, D., Yousri, D. A., & Eteiba, M. B. (2016). Parameters extraction of the three diode model for the multi-crystalline solar cell/module using Moth-Flame Optimization Algorithm. *Energy Conversion and Management*, 123, 535–548. <https://doi.org/10.1016/j.enconman.2016.06.052>
- Anaraki, J. R., Samet, S., Eftekhari, M., & Ahn, C. W. (2018). A Fuzzy-Rough Feature Selection Based on Binary Shuffled Frog Leaping Algorithm. *International Journal of Computer and Information Engineering*, 12(9), 722–729.
- Anfal, M., & Abdelhafid, H. (2017). Optimal placement of PMUS in algerian network using a

- hybrid particle swarm–moth flame optimizer (PSO-MFO). *EEA - Electrotehnica, Electronica, Automatica*, 65(3).
- Arora, P., & Gupta, A. (2013). Comparison of Two Different Strategies to Control Particle Size Distribution in Batch Emulsion Polymerization Using PID Controller. *International Journal of Scientific Engineering and Technology*, 2(11), 1138–1143.
- Aruna, R., & Jaya Christa, S. T. (2020). Modeling, system identification and design of fuzzy PID controller for discharge dynamics of metal hydride hydrogen storage bed. *International Journal of Hydrogen Energy*, 45(7), 4703–4719. <https://doi.org/10.1016/j.ijhydene.2019.11.238>
- Åström, K. J., & Hägglund, T. (1984). Automatic tuning of simple regulators with specifications on phase and amplitude margins. *Automatica*, 20(5), 645–651. [https://doi.org/10.1016/0005-1098\(84\)90014-1](https://doi.org/10.1016/0005-1098(84)90014-1)
- Åström, K. J., & Hägglund, T. (1995). PID controllers: theory, design, and tuning. *ISA-The Instrumentation, Systems and Automation Society*.
- Åström, K. J., & Hägglund, T. (2001). The future of PID control. *Control Engineering Practice*, 9(11), 1163–1175. [https://doi.org/10.1016/S0967-0661\(01\)00062-4](https://doi.org/10.1016/S0967-0661(01)00062-4)
- Aziz, M. A. El, Ewees, A. A., & Hassanien, A. E. (2017). Whale Optimization Algorithm and Moth-Flame Optimization for multilevel thresholding image segmentation. *Expert Systems with Applications*, 83, 242–256. <https://doi.org/10.1016/j.eswa.2017.04.023>
- Balaton, M. G., Nagy, L., & Szeifert, F. (2013). Jacket temperature control of a batch reactor using three different temperature levels. *Industrial and Engineering Chemistry Research*, 52(5). <https://doi.org/10.1021/ie300734b>
- Bandopadhyay, J., & Roy, P. K. (2020). Application of hybrid multi-objective moth flame optimization technique for optimal performance of hybrid micro-grid system. *Applied Soft Computing*, 95, 106487. <https://doi.org/10.1016/j.asoc.2020.106487>
- Bansal, J. C., Sharma, H., Jadon, S. S., & Clerc, M. (2014). Spider Monkey Optimization algorithm for numerical optimization. *Memetic Computing*, 6(1), 31–47. <https://doi.org/10.1007/s12293-013-0128-0>
- Baset, M. A., Zhou, Y., & Ismail, M. (2018). An improved cuckoo search algorithm for integer programming problems. *International Journal of Computing Science and Mathematics*, 9(1), 66. <https://doi.org/10.1504/IJCSM.2018.090710>

- Bequette, B. W. (2003). *Process control: modeling, design, and simulation*. Prentice Hall Professional, Upper Saddle River, NJ.
- Bharat, S., Ganguly, A., Chatterjee, R., Basak, B., Sheet, D. K., & Ganguly, A. (2019). A Review on Tuning Methods for PID controller. *Asian Journal of Convergence in Technology (AJCT)*, 5(1), 1–4.
- Bhardwaj, S., & Agarwal, R. (2022). An efficient speaker identification framework based on Mask R-CNN classifier parameter optimized using hosted cuckoo optimization (HCO). *Journal of Ambient Intelligence and Humanized Computing*, 1-13.
- Bhesdadiya, R. H., Trivedi, I. N., Jangir, P., Kumar, A., Jangir, N., & Totlani, R. (2017). A Novel Hybrid Approach Particle Swarm Optimizer with Moth-Flame Optimizer Algorithm. In *Advances in Intelligent Systems and Computing* (Vol. 553, pp. 569–577). https://doi.org/10.1007/978-981-10-3770-2_53
- Bingul, Z., & Karahan, O. (2018). A novel performance criterion approach to optimum design of PID controller using cuckoo search algorithm for AVR system. *Journal of the Franklin Institute*, 355(13), 5534–5559. <https://doi.org/10.1016/j.jfranklin.2018.05.056>
- Buch, H., & Trivedi, I. N. (2019). An Efficient Adaptive Moth Flame Optimization Algorithm for Solving Large-Scale Optimal Power Flow Problem with POZ, Multifuel and Valve-Point Loading Effect. *Iranian Journal of Science and Technology, Transactions of Electrical Engineering*, 43(4), 1031–1051. <https://doi.org/10.1007/s40998-019-00211-9>
- Bucz, Š., & Kozáková, A. (2018). Advanced Methods of PID Controller Tuning for Specified Performance. In *PID Control for Industrial Processes*. <https://doi.org/10.5772/intechopen.76069>
- Cai, X., Gao, X. Z., & Xue, Y. (2016). Improved bat algorithm with optimal forage strategy and random disturbance strategy. *International Journal of Bio-Inspired Computation*, 8(4), 205. <https://doi.org/10.1504/IJBIC.2016.078666>
- Cairone, F., Anandan, P., & Bucolo, M. (2018). Nonlinear systems synchronization for modeling two-phase microfluidics flows. *Nonlinear Dynamics*, 92(1), 75–84. <https://doi.org/10.1007/s11071-017-3819-0>
- Cedro, L., & Wiczorkowski, K. (2019). Optimizing PID controller gains to model the performance of a quadcopter. *Transportation Research Procedia*, 40, 156–169. <https://doi.org/10.1016/j.trpro.2019.07.026>

- Chen, H., Bowels, S., Zhang, B., & Fuhlbrigge, T. (2019). Controller parameter optimization for complex industrial system with uncertainties. *Measurement and Control*, 52(7–8), 888–895. <https://doi.org/10.1177/0020294019830108>
- Chen, Q., Fu, R. H., & Xu, Y. C. (2015). Electrical circuit analogy for heat transfer analysis and optimization in heat exchanger networks. *Applied Energy*, 139, 81–92. <https://doi.org/10.1016/j.apenergy.2014.11.021>
- Chidambaram, M., & Saxena, N. (2018). Refined Ziegler–Nichols Tuning Method for Unstable SISO Systems. In *Advances in Industrial Control* (Issue 9789811077265, pp. 127–150). https://doi.org/10.1007/978-981-10-7727-2_7
- Choi, T. J., Lee, J. H., Youn, H. Y., & Ahn, C. W. (2019). Adaptive differential evolution with elite opposition-based learning and its application to training artificial neural networks. *Fundamenta Informaticae*, 164(2-3), 227–242.
- Chu, S. C., Tsai, P. W., & Pan, J. S. (2006). Cat swarm optimization. In *Pacific Rim International Conference on Artificial Intelligence*, Springer, Berlin, Heidelberg.
- Cohen, G. H., & Coon, G. A. (1953). Theoretical Consideration of Retarded Control., *Trans. ASME*, 75, 827–834.
- Colombino, M., Dall’Anese, E., & Bernstein, A. (2020). Online Optimization as a Feedback Controller: Stability and Tracking. *IEEE Transactions on Control of Network Systems*, 7(1), 422–432. <https://doi.org/10.1109/TCNS.2019.2906916>
- Cortés, P., Muñozuri, J., Onieva, L., & Guadix, J. (2018). A discrete particle swarm optimisation algorithm to operate distributed energy generation networks efficiently. *International Journal of Bio-Inspired Computation*, 12(4), 226–235.
- Coughanowr, D. R., & Koppel, L. B. (1965). Process systems analysis and control (Vol. 2). New York: McGraw-Hill.
- Coughanowr, D. R., & LeBlanc, S. E. (2009). Process Systems Analysis and Control. In *McGraw-Hill*.
- Cui, Zhihua, Li, F., & Zhang, W. (2019). Bat algorithm with principal component analysis. *International Journal of Machine Learning and Cybernetics*, 10(3), 603–622. <https://doi.org/10.1007/s13042-018-0888-4>
- Cui, Zhihua, Sun, B., Wang, G., Xue, Y., & Chen, J. (2017). A novel oriented cuckoo search algorithm to improve DV-Hop performance for cyber–physical systems. *Journal of Parallel*

- and Distributed Computing*, 103, 42–52. <https://doi.org/10.1016/j.jpdc.2016.10.011>
- Cui, Zhihua, Zhang, J., Wang, Y., Cao, Y., Cai, X., Zhang, W., & Chen, J. (2019). A pigeon-inspired optimization algorithm for many-objective optimization problems. *Science China Information Sciences*, 62(7), 70212. <https://doi.org/10.1007/s11432-018-9729-5>
- Cui, Zhiling, Li, C., Huang, J., Wu, Y., & Zhang, L. (2020). An Improved Moth Flame Optimization Algorithm for Minimizing Specific Fuel Consumption of Variable Cycle Engine. *IEEE Access*, 8, 142725–142735. <https://doi.org/10.1109/ACCESS.2020.3001156>
- Dahunsi, O. A., Dangor, M., Pedro, J. O., & Ali, M. M. (2020). Proportional+ integral+ derivative control of nonlinear full-car electrohydraulic suspensions using global and evolutionary optimization techniques. *Journal of Low Frequency Noise, Vibration and Active Control*, 39(2), 393-415.
- Dangor, M., Dahunsi, O. A., Pedro, J. O., & Ali, M. M. (2014). Evolutionary algorithm-based PID controller tuning for nonlinear quarter-car electrohydraulic vehicle suspensions. *Nonlinear dynamics*, 78(4), 2795-2810.
- Dash, S. P., Subhashini, K. R., & Satapathy, J. K. (2020). Optimal location and parametric settings of FACTS devices based on JAYA blended moth flame optimization for transmission loss minimization in power systems. *Microsystem Technologies*, 26(5), 1543–1552. <https://doi.org/10.1007/s00542-019-04692-w>
- Dash, S., Subhashini, K. R., & Satapathy, J. (2020). Efficient utilization of power system network through optimal location of FACTS devices using a proposed hybrid meta-heuristic Ant Lion-Moth Flame-Salp Swarm optimization algorithm. *International Transactions on Electrical Energy Systems*, 30(7), e12402. <https://doi.org/10.1002/2050-7038.12402>
- Davanipour, M., Javanmardi, H., & Goodarzi, N. (2018). Chaotic Self-Tuning PID Controller Based on Fuzzy Wavelet Neural Network Model. *Iranian Journal of Science and Technology, Transactions of Electrical Engineering*, 42(3), 357–366. <https://doi.org/10.1007/s40998-018-0069-1>
- Dey, S., Gupta, N., Pathak, S., Kela, D. H., & Datta, S. (2019). Data-driven design optimization for industrial products. In *Optimization in industry* (pp. 253-267). Springer, Cham.
- Dhyani, A., Panda, M. K., & Jha, B. (2018). Moth-Flame Optimization-Based Fuzzy-PID Controller for Optimal Control of Active Magnetic Bearing System. *Iranian Journal of Science and Technology, Transactions of Electrical Engineering*, 42(4), 451–463.

- <https://doi.org/10.1007/s40998-018-0077-1>
- Ding, T., Chang, L., Li, C., Feng, C., & Zhang, N. (2018). A Mixed-Strategy-Based Whale Optimization Algorithm for Parameter Identification of Hydraulic Turbine Governing Systems with a Delayed Water Hammer Effect. *Energies*, *11*(9), 2367. <https://doi.org/10.3390/en11092367>
- Dinkar, S. K., & Deep, K. (2018). An efficient opposition based Lévy Flight Antlion optimizer for optimization problems. *Journal of Computational Science*, *29*, 119–141. <https://doi.org/10.1016/j.jocs.2018.10.002>
- Dinkar, S. K., & Deep, K. (2019). Accelerated Opposition-Based Antlion Optimizer with Application to Order Reduction of Linear Time-Invariant Systems. *Arabian Journal for Science and Engineering*, *44*(3), 2213–2241. <https://doi.org/10.1007/s13369-018-3370-4>
- Dorigo, M. (1992). *Optimization, learning and natural algorithms*. Politecnico di Milano, Italy.
- Drias, H., Sadeg, S., & Yahi, S. (2005). Cooperative Bees Swarm for Solving the Maximum Weighted Satisfiability Problem. In *Lecture Notes in Computer Science* (Vol. 3512, pp. 318–325). https://doi.org/10.1007/11494669_39
- Eberhart, R. C., Shi, Y., & Kennedy, J. (2001). *Swarm Intelligence*. San Francisco: Morgan Kaufmann Publishers Inc.
- Eberhart R. C., & Shi Y. (2001). Particle swarm optimization: developments, applications and resources. *Proceedings of the 2001 Congress on Evolutionary Computation (IEEE Cat. No.01TH8546)*, *1*, 81–86. <https://doi.org/10.1109/CEC.2001.934374>
- El-Gendy, E. M., Saafan, M. M., Elksas, M. S., Saraya, S. F., & Areed, F. F. G. (2020). Applying hybrid genetic–PSO technique for tuning an adaptive PID controller used in a chemical process. *Soft Computing*, *24*(5), 3455–3474. <https://doi.org/10.1007/s00500-019-04106-z>
- Elattar, E. E., & Elsayed, S. K. (2020). Optimal Location and Sizing of Distributed Generators Based on Renewable Energy Sources Using Modified Moth Flame Optimization Technique. *IEEE Access*, *8*, 109625–109638. <https://doi.org/10.1109/ACCESS.2020.3001758>
- Elaziz, M. A., Ewees, A. A., Ibrahim, R. A., & Lu, S. (2020). Opposition-based moth-flame optimization improved by differential evolution for feature selection. *Mathematics and Computers in Simulation*, *168*, 48–75. <https://doi.org/10.1016/j.matcom.2019.06.017>
- Elsakaan, A. A., El-Sehiemy, R. A., Kaddah, S. S., & Elsaid, M. I. (2018). An enhanced moth-flame optimizer for solving non-smooth economic dispatch problems with emissions. *Energy*,

- 157, 1063–1078. <https://doi.org/10.1016/j.energy.2018.06.088>
- Fan, Q., Chen, Z., Li, Z., Xia, Z., Yu, J., & Wang, D. (2021). A new improved whale optimization algorithm with joint search mechanisms for high-dimensional global optimization problems. *Engineering with Computers*, 37(3), 1851–1878. <https://doi.org/10.1007/s00366-019-00917-8>
- Fei, W., Hexiang, B., Deyu, L., & Jianjun, W. (2020). Energy-Efficient Clustering Algorithm in Underwater Sensor Networks Based on Fuzzy C Means and Moth-Flame Optimization Method. *IEEE Access*, 8, 97474–97484. <https://doi.org/10.1109/ACCESS.2020.2997066>
- Figueiredo, E. M. N., Ludermir, T. B., & Bastos-Filho, C. J. A. (2016). Many Objective Particle Swarm Optimization. *Information Sciences*, 374, 115–134. <https://doi.org/10.1016/j.ins.2016.09.026>
- Fonseca, R. R., Schmitz, J. E., Fileti, A. M. F., & da Silva, F. V. (2013). A fuzzy–split range control system applied to a fermentation process. *Bioresource technology*, 142, 475–482.
- Gaidhane, P. J., & Nigam, M. J. (2018). A hybrid grey wolf optimizer and artificial bee colony algorithm for enhancing the performance of complex systems. *Journal of Computational Science*, 27, 284–302. <https://doi.org/10.1016/j.jocs.2018.06.008>
- Gaing, Z.-L. (2004). A Particle Swarm Optimization Approach for Optimum Design of PID Controller in AVR System. *IEEE Transactions on Energy Conversion*, 19(2), 384–391. <https://doi.org/10.1109/TEC.2003.821821>
- Gandomi, A. H., & Alavi, A. H. (2012). Krill herd: A new bio-inspired optimization algorithm. *Communications in Nonlinear Science and Numerical Simulation*, 17(12), 4831–4845. <https://doi.org/10.1016/j.cnsns.2012.05.010>
- Gang, W. (2020). ESO-Based Terminal Sliding Mode Control for Uncertain Full-Car Active Suspension Systems. *International Journal of Automotive Technology*, 21(3), 691–702. <https://doi.org/10.1007/s12239-020-0067-y>
- Gao, S.-Z., Wu, X.-F., Luan, L.-L., Wang, J.-S., & Wang, G.-C. (2018). PSO optimal control of model-free adaptive control for PVC polymerization process. *International Journal of Automation and Computing*, 15(4), 482–491. <https://doi.org/10.1007/s11633-016-0973-7>
- Garg, M., Arora, A., & Gupta, S. (2021). An efficient human identification through iris recognition system. *Journal of Signal Processing Systems*, 93(6), 701–708.
- George, M. A., Kamat, D. V., & Indiran, T. (2021). OTA-C Realization of An Optimized FOPID

- Controller for BLDC Motor Speed Control. *IETE Journal of Research*.
<https://doi.org/10.1080/03772063.2021.1951380>
- Gholizadeh, S., Davoudi, H., & Fattahi, F. (2017). Design of steel frames by an enhanced moth-flame optimization algorithm. *Steel and Composite Structures*, 24(1), 129–140.
<https://doi.org/10.12989/scs.2017.24.1.129>
- Gilaber, P., & Paris, J. (1988). A thermal-electrical analogy model of heat exchanges in a solar greenhouse. *Applied Energy*, 31(2), 133–160. [https://doi.org/10.1016/0306-2619\(88\)90026-8](https://doi.org/10.1016/0306-2619(88)90026-8)
- Glover, F. (1989). Tabu Search—Part I. *ORSA Journal on Computing*, 1(3).
<https://doi.org/10.1287/ijoc.1.3.190>
- Goud, H., & Swarnkar, P. (2019). Investigations on Metaheuristic Algorithm for Designing Adaptive PID Controller for Continuous Stirred Tank Reactor. *MAPAN*, 34(1), 113–119.
<https://doi.org/10.1007/s12647-018-00300-w>
- Greeshma, M. C., Bhagyalakshmi, R., & Jaganatha Pandian, B. (2019). Temperature control using ACO and PSO based PD. *Journal of Advanced Research in Dynamical and Control Systems*, 11, 2332–2340.
- Guha, D., Roy, P. K., & Banerjee, S. (2016). Application of Krill Herd Algorithm for Optimum Design of Load Frequency Controller for Multi-Area Power System Network with Generation Rate Constraint. In *Advances in Intelligent Systems and Computing* (pp. 245–257).
https://doi.org/10.1007/978-81-322-2695-6_22
- Gupta, P., Rana, K. P. S., Kumar, V., & Mishra, P. (2015). Split-range control of a Jacketed CSTR using self-tuning fuzzy PI controller. *2015 International Conference on Advances in Computer Engineering and Applications*, 527–533. <https://doi.org/10.1109/ICACEA.2015.7164750>
- Gupta, R., Dhindsa, I. S., & Agarwal, R. (2021). Surface Electromyogram Feature Set Optimization for Lower Limb Activity Classification. *IETE Journal of Research*, 1-15.
- Gupta, S., & Deep, K. (2019). An Efficient Grey Wolf Optimizer with Opposition-Based Learning and Chaotic Local Search for Integer and Mixed-Integer Optimization Problems. *Arabian Journal for Science and Engineering*, 44(8), 7277–7296. <https://doi.org/10.1007/s13369-019-03806-w>
- Gupta, T. K., & Raza, K. (2019). Optimization of ANN Architecture: A Review on Nature-Inspired Techniques. In *Machine Learning in Bio-Signal Analysis and Diagnostic Imaging* (pp. 159–182). Elsevier. <https://doi.org/10.1016/B978-0-12-816086-2.00007-2>

- Guzmán, J. L., & Hägglund, T. (2021). Tuning rules for feedforward control from measurable disturbances combined with PID control: a review. *International Journal of Control*. <https://doi.org/10.1080/00207179.2021.1978537>
- Guzmán, J. L., & Hägglund, T. (2011). Simple tuning rules for feedforward compensators. *Journal of Process Control*, 21, 92–102.
- Hang, C. C., Astrom, K. J., & Wang, Q. G. (2002). Relay feedback auto-tuning of process controllers - A tutorial review. In *Journal of Process Control* (Vol. 12, Issue 1). [https://doi.org/10.1016/S0959-1524\(01\)00025-7](https://doi.org/10.1016/S0959-1524(01)00025-7)
- Hassanien, A. E., Gaber, T., Mokhtar, U., & Hefny, H. (2017). An improved moth flame optimization algorithm based on rough sets for tomato diseases detection. *Computers and Electronics in Agriculture*, 136, 86–96. <https://doi.org/10.1016/j.compag.2017.02.026>
- Helmi, A., & Alenany, A. (2020). An enhanced Moth-flame optimization algorithm for permutation-based problems. *Evolutionary Intelligence*, 13(4), 741–764. <https://doi.org/10.1007/s12065-020-00389-6>
- Hermawanto, D., Putri, C. C., Dwisetyo, B., Prasasti, N. R., Palupi, M. R., & Rusjadi, D. (2020). Development of Decibel Stepped Attenuator for Automated Sound Level Meter Calibration. *MAPAN*, 35(1), 81–86. <https://doi.org/10.1007/s12647-019-00332-w>
- Hernández-Alvarado, R., García-Valdovinos, L., Salgado-Jiménez, T., Gómez-Espinosa, A., & Fonseca-Navarro, F. (2016). Neural Network-Based Self-Tuning PID Control for Underwater Vehicles. *Sensors*, 16(9), 1429. <https://doi.org/10.3390/s16091429>
- Ho-Huu, V., Nguyen-Thoi, T., Nguyen-Thoi, M. H., & Le-Anh, L. (2015). An improved constrained differential evolution using discrete variables (D-ICDE) for layout optimization of truss structures. *Expert Systems with Applications*, 42(20), 7057–7069. <https://doi.org/10.1016/j.eswa.2015.04.072>
- Hongwei, L. I., Jianyong, L. I. U., Liang, C. H. E. N., Jingbo, B. A. I., Yangyang, S. U. N., & Kai, L. U. (2019). Chaos-enhanced moth-flame optimization algorithm for global optimization. *Journal of Systems Engineering and Electronics*, 30(6), 1144–1159. <https://doi.org/10.21629/JSEE.2019.06.10>
- Jain, P., & Saxena, A. (2019). An opposition theory enabled moth flame optimizer for strategic bidding in uniform spot energy market. *Engineering Science and Technology, an International Journal*, 22(4), 1047–1067. <https://doi.org/10.1016/j.jestch.2019.03.005>

- Jain, S., Kumar, A., & Bajaj, V. (2017, April). QRS complex detection using cuckoo search optimization algorithm. In *2017 International Conference on Communication and Signal Processing (ICCSP)* (pp. 0091-0095). IEEE.
- Jangir, N., Pandya, M. H., Trivedi, I. N., Bhesdadiya, R. H., Jangir, P., & Kumar, A. (2016). Moth-Flame optimization Algorithm for solving real challenging constrained engineering optimization problems. *2016 IEEE Students' Conference on Electrical, Electronics and Computer Science (SCEECS)*, 1–5. <https://doi.org/10.1109/SCEECS.2016.7509293>
- Jangir P. (2017). Optimal power flow using a hybrid particle swarm optimizer with moth flame optimizer. *Global Journal of Research in Engineering*, *17*, 524–542.
- Jangir, P. (2018). Non-Dominated Sorting Moth Flame Optimizer: A Novel Multi-Objective Optimization Algorithm for Solving Engineering Design Problems. *Engineering Technology Open Access Journal*, *2*(1), 555579. <https://doi.org/10.19080/ETOAJ.2018.02.555579>
- Jha, S. K., & Kumar, D. (2020). Assessment of Battery Energy Storage System with Hybrid Renewable Energy Sources to Voltage Control of Islanded Microgrid Considering Demand-Side Management Capability. *Iranian Journal of Science and Technology, Transactions of Electrical Engineering*, *44*(2), 861–877. <https://doi.org/10.1007/s40998-019-00273-9>
- Jia, H., Ma, J., & Song, W. (2019). Multilevel Thresholding Segmentation for Color Image Using Modified Moth-Flame Optimization. *IEEE Access*, *7*, 44097–44134. <https://doi.org/10.1109/ACCESS.2019.2908718>
- Kamalapathi, K., Priyadarshi, N., Padmanaban, S., Holm-Nielsen, J., Azam, F., Umayal, C., & Ramachandaramurthy, V. (2018). A Hybrid Moth-Flame Fuzzy Logic Controller Based Integrated Cuk Converter Fed Brushless DC Motor for Power Factor Correction. *Electronics*, *7*(11), 288. <https://doi.org/10.3390/electronics7110288>
- Kapri, R. K., Rathore, K., Dubey, P. K., Mehrotra, R., & Sharma, P. (2020). Optimization of Control Parameters of PMT-Based Photon Counting System. *MAPAN - Journal of Metrology Society of India*, *35*(2), 177–182. <https://doi.org/10.1007/s12647-019-00357-1>
- Karaboga, D., & Basturk, B. (2007). A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm. *Journal of Global Optimization*, *39*(3), 459–471. <https://doi.org/10.1007/s10898-007-9149-x>
- Karam, Z. A., & Awad, O. A. (2020). Design of Active Fractional PID Controller Based on Whale's Optimization Algorithm for Stabilizing a Quarter Vehicle Suspension System. *Periodica*

- Polytechnica Electrical Engineering and Computer Science*, 64(3), 247–263.
<https://doi.org/10.3311/PPee.14904>
- Karroum, K., Lin, Y., Chiang, Y.-Y., Ben Maissa, Y., El Haziti, M., Sokolov, A., & Delbarre, H. (2020). A Review of Air Quality Modeling. *MAPAN*, 35(2), 287–300.
<https://doi.org/10.1007/s12647-020-00371-8>
- Kaur, K., Singh, U., & Salgotra, R. (2020). An enhanced moth flame optimization. *Neural Computing and Applications*, 32(7), 2315–2349. <https://doi.org/10.1007/s00521-018-3821-6>
- Kennedy, J., Eberhart, R. (1995). Particle swarm optimization. *In Proceedings of ICNN'95-International Conference on Neural Networks*, 1942–1948.
- Khalilpourazari, S., & Khalilpourazary, S. (2019). An efficient hybrid algorithm based on Water Cycle and Moth-Flame Optimization algorithms for solving numerical and constrained engineering optimization problems. *Soft Computing*, 23(5), 1699–1722.
<https://doi.org/10.1007/s00500-017-2894-y>
- Khare, Y. B., & Singh, Y. (2010). PID control of heat exchanger system. *International Journal of Computer Applications*, 8(6), 22-27.
- Kirkpatrick, S., Gelatt, C. D., & Vecchi, M. P. (1983). Optimization by simulated annealing. *Science*, 220(4598). <https://doi.org/10.1126/science.220.4598.671>
- Kofinas, P., & Dounis, A. I. (2018). Fuzzy Q-Learning Agent for Online Tuning of PID Controller for DC Motor Speed Control. *Algorithms*, 11(10), 148. <https://doi.org/10.3390/a11100148>
- Kofinas, P., & Dounis, A. I. (2019). Online Tuning of a PID Controller with a Fuzzy Reinforcement Learning MAS for Flow Rate Control of a Desalination Unit. *Electronics*, 8(2), 231. <https://doi.org/10.3390/electronics8020231>
- Korashy, A., Kamel, S., Alquthami, T., & Jurado, F. (2020). Optimal Coordination of Standard and Non-Standard Direction Overcurrent Relays Using an Improved Moth-Flame Optimization. *IEEE Access*, 8, 87378–87392. <https://doi.org/10.1109/ACCESS.2020.2992566>
- Králová, J., & Doležel, P. (2009). DIFFERENT APPROACHES TO CONTROL OF TISO THERMAL SYSTEM. *Transactions of the VŠB – Technical University of Ostrava, Mechanical Series*, 2, 73–78.
- Krishnamoorthy, D. (2020). *Real-Time Optimization as a Feedback Control Problem - A Review*.
- Kumar, R., Rab, S., Pant, B. D., Maji, S., & Mishra, R. S. (2019). FEA-Based Design Studies for Development of Diaphragm Force Transducers. *MAPAN*, 34(2), 179–187.

- <https://doi.org/10.1007/s12647-018-0292-2>
- Kumar, V., Chhabra, J. K., & Kumar, D. (2014). Parameter adaptive harmony search algorithm for unimodal and multimodal optimization problems. *Journal of Computational Science*, 5(2), 144-155.
- Laddimath, R. S., & Patil, N. S. (2019). Artificial Neural Network Technique for Statistical Downscaling of Global Climate Model. *MAPAN*, 34(1), 121–127. <https://doi.org/10.1007/s12647-018-00299-0>
- Li, C., Niu, Z., Song, Z., Li, B., Fan, J., & Liu, P. X. (2018). A Double Evolutionary Learning Moth-Flame Optimization for Real-Parameter Global Optimization Problems. *IEEE Access*, 6, 76700–76727. <https://doi.org/10.1109/ACCESS.2018.2884130>
- Li, W. K., Wang, W. L., & Li, L. (2018). Optimization of Water Resources Utilization by Multi-Objective Moth-Flame Algorithm. *Water Resources Management*, 32(10), 3303–3316. <https://doi.org/10.1007/s11269-018-1992-7>
- Li, X. L., Shao, Z. J., & Qian, J. X. (2002). Optimizing method based on autonomous animats: Fish-swarm Algorithm. *Xitong Gongcheng Lilun Yu Shijian/System Engineering Theory and Practice*, 22(11), 32–38.
- Li, Yaru, Wang, Z., Cheng, Y., Tang, Y., & Shang, Z. (2019). Application of vision measurement model with an improved moth-flame optimization algorithm. *Optics Express*, 27(15), 20800. <https://doi.org/10.1364/OE.27.020800>
- Li, Yu, Zhu, X., & Liu, J. (2020). An Improved Moth-Flame Optimization Algorithm for Engineering Problems. *Symmetry*, 12(8), 1234. <https://doi.org/10.3390/sym12081234>
- Li, Zhifu, Zeng, J., Chen, Y., Ma, G., & Liu, G. (2021). Death mechanism-based moth–flame optimization with improved flame generation mechanism for global optimization tasks. *Expert Systems with Applications*, 183, 115436. <https://doi.org/10.1016/j.eswa.2021.115436>
- Li, Zhiming, Zhou, Y., Zhang, S., & Song, J. (2016). Lévy-Flight Moth-Flame Algorithm for Function Optimization and Engineering Design Problems. *Mathematical Problems in Engineering*, 2016, 1–22. <https://doi.org/10.1155/2016/1423930>
- Lin, G.-Q., Li, L.-L., Tseng, M.-L., Liu, H.-M., Yuan, D.-D., & Tan, R. R. (2020). An improved moth-flame optimization algorithm for support vector machine prediction of photovoltaic power generation. *Journal of Cleaner Production*, 253, 119966. <https://doi.org/10.1016/j.jclepro.2020.119966>

- Loucif, F., Kechida, S., & Sebbagh, A. (2020). Whale optimizer algorithm to tune PID controller for the trajectory tracking control of robot manipulator. *Journal of the Brazilian Society of Mechanical Sciences and Engineering*, 42(1), 1–11. <https://doi.org/10.1007/s40430-019-2074-3>
- Luo, J. C., & Lee, E. B. (1999, June). Mixed disturbance action for stabilized control systems (necessary conditions). In *Proceedings of the 1999 American Control Conference (Cat. No. 99CH36251)* (Vol. 4, pp. 2563-2567). IEEE.
- Luo, Q., Yang, X., & Zhou, Y. (2019). Nature-inspired approach: An enhanced moth swarm algorithm for global optimization. *Mathematics and Computers in Simulation*, 159, 57–92. <https://doi.org/10.1016/j.matcom.2018.10.011>
- Ma, L., Wang, C., Xie, N., Shi, M., Ye, Y., & Wang, L. (2021). Moth-flame optimization algorithm based on diversity and mutation strategy. *Applied Intelligence*, 51(8), 5836–5872. <https://doi.org/10.1007/s10489-020-02081-9>
- Mahitthimahawong, S., Tarapoom, N., Skogestad, S., Srinophakun, T., & Lertbumrungsuk, V. (2016). Application of Passivity Concept for Split Range Control of Heat Exchanger Networks. *Journal of Chemical Engineering & Process Technology*, 07(01), 274. <https://doi.org/10.4172/2157-7048.1000274>
- Mahmood, Q. A., Nawaf, A. T., Esmael, M. N., Abdulateef, L. T., & Dahham, O. S. (2018, December). PID temperature control of demineralized water tank. In *IOP Conference Series: Materials Science and Engineering* (Vol. 454, No. 1, p. 012031). IOP Publishing.
- Marlin, T. E. (1995). *Process control: Designing Processes and Control Systems for Dynamic Performance*. Chemical Engineering Series, McGraw-Hill International Editions: New York.
- Maurya, L., Mahapatra, P. K., & Kumar, A. (2017). A social spider optimized image fusion approach for contrast enhancement and brightness preservation. *Applied Soft Computing*, 52, 575-592.
- Memon, F., & Shao, C. (2020). An Optimal Approach to Online Tuning Method for PID Type Iterative Learning Control. *International Journal of Control, Automation and Systems*, 18(8), 1926–1935. <https://doi.org/10.1007/s12555-018-0840-0>
- Mien, T. L., An, V. Van, & Tam, B. T. (2020). A Fuzzy-PID Controller Combined with PSO Algorithm for the Resistance Furnace. *Advances in Science, Technology and Engineering Systems Journal*, 5(3), 568–575. <https://doi.org/10.25046/aj050371>

- Mirjalili, S. (2015). Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm. *Knowledge-Based Systems*, 89, 228–249. <https://doi.org/10.1016/j.knosys.2015.07.006>
- Mirjalili, S., & Lewis, A. (2016). The Whale Optimization Algorithm. *Advances in Engineering Software*, 95, 51–67. <https://doi.org/10.1016/j.advengsoft.2016.01.008>
- Mohanty, B. (2019). Performance analysis of moth flame optimization algorithm for AGC system. *International Journal of Modelling and Simulation*, 39(2), 73–87. <https://doi.org/10.1080/02286203.2018.1476799>
- Mohanty, B., Acharyulu, B. V. S., & Hota, P. K. (2018). Moth-flame optimization algorithm optimized dual-mode controller for multiarea hybrid sources AGC system. *Optimal Control Applications and Methods*, 39(2), 720–734. <https://doi.org/10.1002/oca.2373>
- Mohanty, D., & Panda, S. (2021). A modified moth flame optimisation technique tuned adaptive fuzzy logic PID controller for frequency regulation of an autonomous power system. *International Journal of Sustainable Energy*, 40(1), 41–68. <https://doi.org/10.1080/14786451.2020.1787412>
- Montoya-Ríos, A. P., García-Mañas, F., Guzmán, J. L., & Rodríguez, F. (2020). Simple tuning rules for feedforward compensators applied to greenhouse daytime temperature control using natural ventilation. *Agronomy*, 10(9). <https://doi.org/10.3390/agronomy10091327>
- Mosaad, A. M., Attia, M. A., & Abdelaziz, A. Y. (2019). Whale optimization algorithm to tune PID and PIDA controllers on AVR system. *Ain Shams Engineering Journal*, 10(4), 755–767. <https://doi.org/10.1016/j.asej.2019.07.004>
- Nagaraj, S., Raju, G. S. V. P., & Srinadth, V. (2015). Data Encryption and Authentication Using Public Key Approach. *Procedia Computer Science*, 48(C), 126–132. <https://doi.org/10.1016/j.procs.2015.04.161>
- Nagy, Z. K. (2007). Model based control of a yeast fermentation bioreactor using optimally designed artificial neural networks. *Chemical Engineering Journal*, 127(1–3), 95–109. <https://doi.org/10.1016/j.cej.2006.10.015>
- Naimi, A., Deng, J., Sheikh-Akbari, A., SR, S., & Arul, J. (2022). Input-Output Feedback Linearization Control for a PWR Nuclear Power Plant.
- Ng Shin Mei, R., Sulaiman, M. H., Mustaffa, Z., & Daniyal, H. (2017). Optimal reactive power dispatch solution by loss minimization using moth-flame optimization technique. *Applied Soft*

- Computing*, 59, 210–222. <https://doi.org/10.1016/j.asoc.2017.05.057>
- Nikolic, K. P. (2015). Stochastic Search Algorithms for Identification, Optimization, and Training of Artificial Neural Networks. *Advances in Artificial Neural Systems*, 2015. <https://doi.org/10.1155/2015/931379>
- Ning, J., Zhang, Q., Zhang, C., & Zhang, B. (2018). A best-path-updating information-guided ant colony optimization algorithm. *Information Sciences*, 433–434, 142–162. <https://doi.org/10.1016/j.ins.2017.12.047>
- Odili, J. B., Kahar, M. N. M., Anwar, S., & Ali, M. (2017). Tutorials on African buffalo optimization for solving the travelling salesman problem. *International Journal of Software Engineering and Computer Systems*, 3(3), 120-128.
- Passino, K. M. (2002). Biomimicry of bacterial foraging for distributed optimization and control. *IEEE Control Systems*, 22(3), 52–67. <https://doi.org/10.1109/MCS.2002.1004010>
- Pathak, V. K., & Singh, A. K. (2017). Effective Form Error Assessment Using Improved Particle Swarm Optimization. *MAPAN*, 32(4), 279–292. <https://doi.org/10.1007/s12647-017-0225-5>
- Pecora, L. M., & Carroll, T. L. (2015). Synchronization of chaotic systems. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 25(9), 097611. <https://doi.org/10.1063/1.4917383>
- Pedro, J. O., Dangor, M., Dahunsi, O. A., & Ali, M. M. (2018). Dynamic neural network-based feedback linearization control of full-car suspensions using PSO. *Applied Soft Computing*, 70, 723-736.
- Pelusi, D., Mascella, R., Tallini, L., Nayak, J., Naik, B., & Deng, Y. (2020). An Improved Moth-Flame Optimization algorithm with hybrid search phase. *Knowledge-Based Systems*, 191, 105277. <https://doi.org/10.1016/j.knosys.2019.105277>
- Poddar, S., Narkhede, P., Kumar, V., & Kumar, A. (2017). PSO aided adaptive complementary filter for attitude estimation. *Journal of Intelligent & Robotic Systems*, 87(3), 531-543.
- Rab, S., Yadav, S., Zafer, A., Haleem, A., Dubey, P. K., Singh, J., Kumar, R., Sharma, R., & Kumar, L. (2019). Comparison of Monte Carlo Simulation, Least Square Fitting and Calibration Factor Methods for the Evaluation of Measurement Uncertainty Using Direct Pressure Indicating Devices. *MAPAN*, 34(3), 305–315. <https://doi.org/10.1007/s12647-019-00333-9>
- Rahnamayan, S., Tizhoosh, H. R., & Salama, M. M. A. (2008). Opposition-Based Differential Evolution. *IEEE Transactions on Evolutionary Computation*, 12(1), 64–79.

- <https://doi.org/10.1109/TEVC.2007.894200>
- Rahnamayan, Shahryar, Tizhoosh, H. R., & Salama, M. M. A. (2008). Opposition versus randomness in soft computing techniques. *Applied Soft Computing*, 8(2), 906–918. <https://doi.org/10.1016/j.asoc.2007.07.010>
- Rajesh, R. (2019). Optimal tuning of FOPID controller based on PSO algorithm with reference model for a single conical tank system. *SN Applied Sciences*, 1(7), 1–14. <https://doi.org/10.1007/s42452-019-0754-3>
- Ramachandran, A., Rustum, R., & Adeloye, A. J. (2019). Review of anaerobic digestion modeling and optimization using nature-inspired techniques. In *Processes* (Vol. 7, Issue 12). <https://doi.org/10.3390/PR7120953>
- Ranjan, R., & Chhabra, J. K. (2022). Automatic Data Clustering using Dynamic Crow Search Algorithm. *EAI Endorsed Transactions on Context-aware Systems and Applications*, 8, e5–e5.
- Rashid, M. F. F. A., Rose, A. N. M., Mohamed, N. M. Z. N., & Romlay, F. R. M. (2019). Improved moth flame optimization algorithm to optimize cost-oriented two-sided assembly line balancing. *Engineering Computations*, 37(2), 638–663. <https://doi.org/10.1108/EC-12-2018-0593>
- Reddy, S., Panwar, L. K., Panigrahi, B. K., & Kumar, R. (2018). Solution to unit commitment in power system operation planning using binary coded modified moth flame optimization algorithm (BMMFOA): A flame selection based computational technique. *Journal of Computational Science*, 25, 298–317. <https://doi.org/10.1016/j.jocs.2017.04.011>
- Reddy, K. N., & Bojja, P. (2020). A new hybrid optimization method combining moth–flame optimization and teaching–learning-based optimization algorithms for visual tracking. *Soft Computing*, 24(24), 18321–18347. <https://doi.org/10.1007/s00500-020-05032-1>
- Reyes-Lúa, A., & Skogestad, S. (2019). Multiple-Input Single-Output Control for Extending the Steady-State Operating Range—Use of Controllers with Different Setpoints. *Processes*, 7(12), 941. <https://doi.org/10.3390/pr7120941>
- Reyes-Lúa, A., & Skogestad, S. (2020). Multi-input single-output control for extending the operating range: Generalized split range control using the baton strategy. *Journal of Process Control*, 91, 1–11. <https://doi.org/10.1016/j.jprocont.2020.05.001>
- Reyes-Lúa, A., Zotică, C., & Skogestad, S. (2018). Optimal Operation with Changing Active Constraint Regions using Classical Advanced Control. *IFAC-PapersOnLine*, 51(18), 440–

445. <https://doi.org/10.1016/j.ifacol.2018.09.340>
- Rodríguez, C., Aranda-Escolástico, E., Guzmán, J. L., Berenguel, M., & Hägglund, T. (2020). Revisiting the simplified internal model control tuning rules for low-order controllers: Feedforward controller. *IET Control Theory and Applications*, 14(12). <https://doi.org/10.1049/iet-cta.2019.0823>
- Sachdeva, J., Kumar, V., Gupta, I., Khandelwal, N., & Ahuja, C. K. (2011, December). Multiclass brain tumor classification using GA-SVM. In *2011 Developments in E-systems Engineering* (pp. 182-187). IEEE.
- Safarzadeh, O., & Noori-kalkhoran, O. (2021). A fractional PID controller based on fractional point kinetic model and particle swarm optimization for power regulation of SMART reactor. *Nuclear Engineering and Design*, 377, 111137. <https://doi.org/10.1016/j.nucengdes.2021.111137>
- Saidala, R. K., & Devarakonda, N. (2018). Improved Whale Optimization Algorithm Case Study: Clinical Data of Anaemic Pregnant Woman. In *Advances in Intelligent Systems and Computing* (Vol. 542, pp. 271–281). https://doi.org/10.1007/978-981-10-3223-3_25
- Sapre, S., & Mini, S. (2019). Opposition-based moth flame optimization with Cauchy mutation and evolutionary boundary constraint handling for global optimization. *Soft Computing*, 23(15), 6023–6041. <https://doi.org/10.1007/s00500-018-3586-y>
- Sapre, S., & Mini, S. (2021). A differential moth flame optimization algorithm for mobile sink trajectory. *Peer-to-Peer Networking and Applications*, 14(1), 44–57. <https://doi.org/10.1007/s12083-020-00947-w>
- Savsani, V., & Tawhid, M. A. (2017). Non-dominated sorting moth flame optimization (NS-MFO) for multi-objective problems. *Engineering Applications of Artificial Intelligence*, 63, 20–32. <https://doi.org/10.1016/j.engappai.2017.04.018>
- Saxena, S. C., Kumar, V., & Waghmare, L. M. (2002). Cascade control of interconnected system using neural network. *IETE journal of research*, 48(6), 461-469.
- Sayed, G. I., Darwish, A., & Hassanien, A. E. (2020). Binary Whale Optimization Algorithm and Binary Moth Flame Optimization with Clustering Algorithms for Clinical Breast Cancer Diagnoses. *Journal of Classification*, 37(1), 66–96. <https://doi.org/10.1007/s00357-018-9297-3>
- Sayed, G. I., & Hassanien, A. E. (2018). A hybrid SA-MFO algorithm for function optimization

- and engineering design problems. *Complex & Intelligent Systems*, 4(3), 195–212. <https://doi.org/10.1007/s40747-018-0066-z>
- Seborg, D., Edgar, T., Mellicamp, D., & Doyle III, F. (2011). *Process Dynamics and Control* (3rd Edition). In *John Wiley & Sons*.
- Seborg, D. E., Edgar, T. F., Mellichamp, D. A., & Wiley, H. (2008). *Process Dynamics and Control*, 2nd.
- Sen, R., Pati, C., Dutta, S., & Sen, R. (2015). Comparison Between Three Tuning Methods of PID Control for High Precision Positioning Stage. *MAPAN*, 30(1), 65–70. <https://doi.org/10.1007/s12647-014-0123-z>
- Sethi, D. (2020). An Approach to Optimize Homogeneous and Heterogeneous Routing Protocols in WSN Using Sink Mobility. *MAPAN*, 35(2), 241–250. <https://doi.org/10.1007/s12647-020-00366-5>
- Sharma, M., & Chhabra, J. K. (2021). An efficient hybrid PSO polygamous crossover based clustering algorithm. *Evolutionary Intelligence*, 14(3), 1213–1231.
- Sharma, R., & Saha, A. (2020). An integrated approach of class testing using firefly and moth flame optimization algorithm. *Journal of Information and Optimization Sciences*, 41(2), 599–612. <https://doi.org/10.1080/02522667.2020.1733192>
- Sharma, Y., & Saikia, L. C. (2015). Automatic generation control of a multi-area ST – Thermal power system using Grey Wolf Optimizer algorithm based classical controllers. *International Journal of Electrical Power & Energy Systems*, 73, 853–862. <https://doi.org/10.1016/j.ijepes.2015.06.005>
- Shehab, M., Alshawabkah, H., Abualigah, L., & AL-Madi, N. (2021). Enhanced a hybrid moth-flame optimization algorithm using new selection schemes. *Engineering with Computers*, 37(4), 2931–2956. <https://doi.org/10.1007/s00366-020-00971-7>
- Sheng, H., Li, C., Wang, H., Yan, Z., Xiong, Y., Cao, Z., & Kuang, Q. (2019). Parameters Extraction of Photovoltaic Models Using an Improved Moth-Flame Optimization. *Energies*, 12(18), 3527. <https://doi.org/10.3390/en12183527>
- Shinskey, F. G. (1996). *Process Control Systems: Application, Design, and Tuning*. In *Book*.
- Siddique, N., & Adeli, H. (2015). Nature Inspired Computing: An Overview and Some Future Directions. *Cognitive Computation*, 7(6), 706–714. [https://doi.org/10.1007/s12559-015-9370-](https://doi.org/10.1007/s12559-015-9370-8)

- Singh, R. K., Gangwar, S., Singh, D. K., & Pathak, V. K. (2019). A novel hybridization of artificial neural network and moth-flame optimization (ANN–MFO) for multi-objective optimization in magnetic abrasive finishing of aluminium 6060. *Journal of the Brazilian Society of Mechanical Sciences and Engineering*, *41*(6), 270. <https://doi.org/10.1007/s40430-019-1778-8>
- Singh, S. K. (2009). *Process Control: Concepts Dynamics And Applications*. PHI Learning Pvt. Ltd.
- Singla, S. K., & Arora, A. S. (2012). Optimizing the rotation and translation of fingerprint images using genetic algorithm. *Applied Artificial Intelligence*, *26*(6), 541-553.
- Sohane, A., & Agarwal, R. (2022). A Single Platform for Classification and Prediction using a Hybrid Bioinspired and Deep Neural Network (PSO-LSTM). *MAPAN*, *37*(1), 47-58.
- Solihin, M. I., Tack, L. F., & Kean, M. L. (2011). Tuning of PID Controller Using Particle Swarm Optimization (PSO). *International Journal on Advanced Science, Engineering and Information Technology*, *1*(4), 458–461. <https://doi.org/10.18517/ijaseit.1.4.93>
- Soliman, Khorshid, A.-E.-E. (2016). Modified Moth-Flame Optimization Algorithms for Terrorism Prediction. *International Journal of Application or Innovation in Engineering & Management*, *5*(7), 47–58.
- Soto, G. J. A., & Hernandez-Riveros, J. A. (2019). Evolutionary split range controller for a refrigeration system. *ICINCO 2019 - Proceedings of the 16th International Conference on Informatics in Control, Automation and Robotics*, *1*. <https://doi.org/10.5220/0007930803410351>
- Su, S., Wang, J., Fan, W., & Yin, X. (2007). Good Lattice Swarm Algorithm for Constrained Engineering Design Optimization. *2007 International Conference on Wireless Communications, Networking and Mobile Computing*, 6415–6418. <https://doi.org/10.1109/WICOM.2007.1575>
- Suja, K. R. (2021). Mitigation of power quality issues in smart grid using levy flight based moth flame optimization algorithm. *Journal of Ambient Intelligence and Humanized Computing*, *12*(10), 9209–9228. <https://doi.org/10.1007/s12652-020-02626-3>
- Sun, W., Wang, J., & Wei, X. (2018). An Improved Whale Optimization Algorithm Based on Different Searching Paths and Perceptual Disturbance. *Symmetry*, *10*(6), 210. <https://doi.org/10.3390/sym10060210>

- Taher, M. A., Kamel, S., Jurado, F., & Ebeed, M. (2019). An improved moth-flame optimization algorithm for solving optimal power flow problem. *International Transactions on Electrical Energy Systems*, 29(3), e2743. <https://doi.org/10.1002/etep.2743>
- Tamilselvan, G. M., & Aarthy, P. (2017). Online tuning of fuzzy logic controller using Kalman algorithm for conical tank system. *Journal of Applied Research and Technology*, 15(5), 492–503. <https://doi.org/10.1016/j.jart.2017.05.004>
- Tang, R., Fong, S., Yang, X.-S., & Deb, S. (2012). Wolf search algorithm with ephemeral memory. *Seventh International Conference on Digital Information Management (ICDIM 2012)*, 165–172. <https://doi.org/10.1109/ICDIM.2012.6360147>
- Teodorovic, D., & Dell'Orco, M. (2005). Bee Colony Optimization-Cooperative Learning Approach to Complex Transportation Problems. *Advanced OR and AI Methods in Transportation*, 51, 60.
- Thamallah, A., Sakly, A., & M'Sahli, F. (2019). A new constrained PSO for fuzzy predictive control of Quadruple-Tank process. *Measurement*, 136, 93–104. <https://doi.org/10.1016/j.measurement.2018.12.050>
- Tizhoosh, H. R. (2005). Opposition-Based Learning: A New Scheme for Machine Intelligence. *International Conference on Computational Intelligence for Modelling, Control and Automation and International Conference on Intelligent Agents, Web Technologies and Internet Commerce (CIMCA-IAWTIC'06)*, 1, 695–701. <https://doi.org/10.1109/CIMCA.2005.1631345>
- Veronesi, M., Guzmán, J. L., Visioli, A., & Hägglund, T. (2017). Closed-loop tuning rules for feedforward compensator gains. *IFAC-PapersOnLine*, 50(1). <https://doi.org/10.1016/j.ifacol.2017.08.1186>
- Vikas & Nanda S. J. (2016). Multi-objective moth flame optimization. In: *2016 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*. IEEE, 2470–2476.
- Vishnoi, V., Tiwari, S., & Singla, R. K. (2021). Controller Design for Temperature Control of MISO Water Tank System: Simulation Studies. *International Journal of Cognitive Informatics and Natural Intelligence (IJCINI)*, 15(4), 1-13.
- Vishnoi, V., Tiwari, S., & Singla, R. (2021a). Performance analysis of moth flame optimization-based split-range PID controller. *MAPAN*, 36(1), 67-79.

- Vishnoi, V., Tiwari, S., & Singla, R. (2021b). performance analysis of enhanced MFO-based online-tuned split-range PID controller. *Arabian Journal for Science and Engineering*, 46(10), 9673-9689.
- Vishnoi, V., Tiwari, S., & Singla, R. (2022). Performance Investigation of EMFO-Based Perpetual Online Tuned Variable SR-Controller in Simulated Real Environment. *Cybernetics and Systems*, 1-25.
- Waghmare, L. M., Saxena, S. C., & Kumar, V. (2005). Inferential MRAC neural controller for temperature control of CST process. *networks*, 457(168), 235.
- Wang, B. (2015). A novel artificial bee colony algorithm based on modified search strategy and generalized opposition-based learning. *Journal of Intelligent & Fuzzy Systems*, 28(3), 1023–1037. <https://doi.org/10.3233/IFS-141386>
- Wang, H., Wu, Z., Rahnamayan, S., Sun, H., Liu, Y., & Pan, J. (2014). Multi-strategy ensemble artificial bee colony algorithm. *Information Sciences*, 279, 587–603. <https://doi.org/10.1016/j.ins.2014.04.013>
- Wang, M., Chen, H., Yang, B., Zhao, X., Hu, L., Cai, Z., Huang, H., & Tong, C. (2017). Toward an optimal kernel extreme learning machine using a chaotic moth-flame optimization strategy with applications in medical diagnoses. *Neurocomputing*, 267, 69–84. <https://doi.org/10.1016/j.neucom.2017.04.060>
- Wu, Z., Shen, D., Shang, M., & Qi, S. (2019). Parameter Identification of Single-Phase Inverter Based on Improved Moth Flame Optimization Algorithm. *Electric Power Components and Systems*, 47(4–5), 456–469. <https://doi.org/10.1080/15325008.2019.1607922>
- Xia, J., Zhang, H., Li, R., Chen, H., Turabieh, H., Mafarja, M., & Pan, Z. (2021). Generalized Oppositional Moth Flame Optimization with Crossover Strategy: An Approach for Medical Diagnosis. *Journal of Bionic Engineering*, 18(4), 991–1010. <https://doi.org/10.1007/s42235-021-0068-1>
- Xiong, D., Chen, X., & Yang, J. (2018). Optimization of a Low-Voltage Load Switch for a Smart Meter Based on a Double Response Surface Model. *MAPAN*, 33(3), 261–270. <https://doi.org/10.1007/s12647-018-0264-6>
- Xu, L., Li, Y., Li, K., Beng, G. H., Jiang, Z., Wang, C., & Liu, N. (2018). Enhanced Moth-flame Optimization Based on Cultural Learning and Gaussian Mutation. *Journal of Bionic Engineering*, 15(4). <https://doi.org/10.1007/s42235-018-0063-3>

- Xu, Y., Chen, H., Heidari, A. A., Luo, J., Zhang, Q., Zhao, X., & Li, C. (2019). An efficient chaotic mutative moth-flame-inspired optimizer for global optimization tasks. *Expert Systems with Applications*, *129*, 135–155. <https://doi.org/10.1016/j.eswa.2019.03.043>
- Xu, Y., Chen, H., Luo, J., Zhang, Q., Jiao, S., & Zhang, X. (2019). Enhanced Moth-flame optimizer with mutation strategy for global optimization. *Information Sciences*, *492*, 181–203. <https://doi.org/10.1016/j.ins.2019.04.022>
- Xu, Z., Yu, Y., Yachi, H., Ji, J., Todo, Y., & Gao, S. (2018). A Novel Memetic Whale Optimization Algorithm for Optimization. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics): Vol. 10941 LNCS* (pp. 384–396). https://doi.org/10.1007/978-3-319-93815-8_37
- Yalsavar, M., Karimaghaee, P., Sheikh-Akbari, A., Khooban, M. H., Dehmeshki, J., & Al-Majeed, S. (2022). Kernel Parameter Optimization for Support Vector Machine Based on Sliding Mode Control. *IEEE Access*, *10*, 17003-17017.
- Yang, C., Ji, J., Liu, J., & Yin, B. (2016). Bacterial foraging optimization using novel chemotaxis and conjugation strategies. *Information Sciences*, *363*, 72–95. <https://doi.org/10.1016/j.ins.2016.04.046>
- Yang, W., Wang, J., & Wang, R. (2017). Research and Application of a Novel Hybrid Model Based on Data Selection and Artificial Intelligence Algorithm for Short Term Load Forecasting. *Entropy*, *19*(2), 52. <https://doi.org/10.3390/e19020052>
- Yang, X.-S. (2008). *Nature-Inspired Metaheuristic Algorithms*. Luniver Press, UK.
- Yang, X.-S. (2010). A New Metaheuristic Bat-Inspired Algorithm. In *Studies in Computational Intelligence* (Vol. 284, pp. 65–74). https://doi.org/10.1007/978-3-642-12538-6_6
- Yang, X.-S. (2012). Nature-Inspired Metaheuristic Algorithms: Success and New Challenges. *Journal of Computer Engineering and Information Technology*, *01*(01). <https://doi.org/10.4172/2324-9307.1000e101>
- Yang, X.-S., Lees, J. M., & Morley, C. T. (2006). Application of Virtual Ant Algorithms in the Optimization of CFRP Shear Strengthened Precracked Structures. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics): Vol. 3991 LNCS* (pp. 834–837). https://doi.org/10.1007/11758501_117
- Yang, X.-S., & Suash Deb. (2009). Cuckoo Search via Levy flights. *2009 World Congress*

- on *Nature & Biologically Inspired Computing (NaBIC)*, 210–214. <https://doi.org/10.1109/NABIC.2009.5393690>
- Yang, X. S. (2010). Firefly algorithm, stochastic test functions and design optimisation. *International Journal of Bio-Inspired Computation*, 2(2), 78–84. <https://doi.org/10.1504/IJBIC.2010.032124>
- Yang, X. S., & Deb, S. (2010). Eagle strategy using Lévy walk and firefly algorithms for stochastic optimization. *Studies in Computational Intelligence*, 284. https://doi.org/10.1007/978-3-642-12538-6_9
- Yewale, A., Methekar, R., & Agrawal, S. (2020). Dynamic analysis and split range control for maximization of operating range of continuous microbial fuel cell. *Chinese Journal of Chemical Engineering*, 28(9), 2368–2381. <https://doi.org/10.1016/j.cjche.2020.06.030>
- Yu, C., Heidari, A. A., & Chen, H. (2020). A quantum-behaved simulated annealing algorithm-based moth-flame optimization method. *Applied Mathematical Modelling*, 87, 1–19. <https://doi.org/10.1016/j.apm.2020.04.019>
- Yu, G., & Feng, Y. (2018). Improving firefly algorithm using hybrid strategies. *International Journal of Computing Science and Mathematics*, 9(2), 163–170. <https://doi.org/10.1504/IJCSM.2018.091749>
- Zarachoff, M., Sheikh-Akbari, A., & Monekosso, D. (2018, October). 2d multi-band PCA and its application for ear recognition. In *2018 IEEE International Conference on Imaging Systems and Techniques (IST)* (pp. 1-5). IEEE.
- Zeng, W., Gao, H., & Jing, W. (2014). An improved particle swarm optimization. *Information Technology Journal*, 13(16), 2560.
- Zeng, W., Zhu, W., Hui, T., Chen, L., Xie, J., & Yu, T. (2020). An IMC-PID controller with Particle Swarm Optimization algorithm for MSBR core power control. *Nuclear Engineering and Design*, 360, 110513. <https://doi.org/10.1016/j.nucengdes.2020.110513>
- Zhang, H., Heidari, A. A., Wang, M., Zhang, L., Chen, H., & Li, C. (2020). Orthogonal Nelder-Mead moth flame method for parameters identification of photovoltaic modules. *Energy Conversion and Management*, 211, 112764. <https://doi.org/10.1016/j.enconman.2020.112764>
- Zhang, L., Mistry, K., Neoh, S. C., & Lim, C. P. (2016). Intelligent facial emotion recognition using moth-firefly optimization. *Knowledge-Based Systems*, 111, 248–267. <https://doi.org/10.1016/j.knosys.2016.08.018>

- Zhang, M., Wang, H., Cui, Z., & Chen, J. (2018). Hybrid multi-objective cuckoo search with dynamical local search. *Memetic Computing*, *10*(2), 199–208. <https://doi.org/10.1007/s12293-017-0237-2>
- Zhang, X. J., Xiao, F., & Li, S. (2012). Performance study of a constant temperature and humidity air-conditioning system with temperature and humidity independent control device. *Energy and Buildings*, *49*, 640–646. <https://doi.org/10.1016/j.enbuild.2012.03.020>
- Zhang, Z., Qin, H., Yao, L., Liu, Y., Jiang, Z., Feng, Z., & Ouyang, S. (2020). Improved Multi-objective Moth-flame Optimization Algorithm based on R-domination for cascade reservoirs operation. *Journal of Hydrology*, *581*, 124431. <https://doi.org/10.1016/j.jhydrol.2019.124431>
- Zhao, X., Fang, Y., Liu, L., Li, J., & Xu, M. (2020). An improved moth-flame optimization algorithm with orthogonal opposition-based learning and modified position updating mechanism of moths for global optimization problems. *Applied Intelligence*, *50*(12). <https://doi.org/10.1007/s10489-020-01793-2>
- Zhong, M., & Long, W. (2017). Whale optimization algorithm with nonlinear control parameter. *MATEC Web of Conferences*, *139*, 00157. <https://doi.org/10.1051/mateconf/201713900157>
- Zhou, J., & Dong, S. (2018). Hybrid glowworm swarm optimization for task scheduling in the cloud environment. *Engineering Optimization*, *50*(6), 949–964. <https://doi.org/10.1080/0305215X.2017.1361418>
- Zhou, X., Gao, H., Jia, Y., Li, L., Zhao, L., & Yu, R. (2019). Parameter Optimization on FNN/PID Compound Controller for a Three-Axis Inertially Stabilized Platform for Aerial Remote Sensing Applications. *Journal of Sensors*, *2019*, 1–15. <https://doi.org/10.1155/2019/5067081>
- Ziegler, J. G., & Nichols, N. B. (1942). Optimum settings for automatic controllers. *Trans. ASME*, *64*, 759–765.

Summary of MFO variants:**Table 1** Literature relevant to the MFO variants published in the last decade

Research Group	Methods or modifications	Improvement
<i>Li et al.</i> (Li et al., 2021)	Used opposition learning-based flame generation mechanism, Differential evolution algorithm, Shuffled frog leaping algorithm (SFLA) based local search mechanism, and death mechanism in MFO algorithm	Obtained high-quality flames, improved the population diversity and global search capability, and also decreased the probability of the moth's entrapment in local optima
<i>Ma et al.</i> (Ma et al., 2021)	Introduced an inertia weight of diversity feedback control in the MFO algorithm and added a small probability mutation after position updating step	Overcome the premature convergence problems, balanced the global search ability and exploitation of the algorithm, and improved the convergence speed
<i>Mohanty and Panda</i> (Mohanty & Panda, 2021)	Changed in the convergence constant and the shape of the logarithmic spiral value	Proposed modified MFO algorithm promotes exploitation
<i>Sapre and Mini</i> (Sapre & Mini, 2021)	Proposed differential MFO (DMFO) algorithm (hybridization of differential evolution and MFO)	Balanced the exploration and exploitation capability of the DMFO algorithm
<i>Shehab et al.</i> (Shehab et al., 2021)	Proposed MFOHC, i.e., hybridization of MFO and local-based algorithm with hill climbing (HC) approach, and also used another six popular selection schemes	Improved the speed of the searching, the learning strategy to find the generation of search agent solutions, the quality of the selected solution, and also increased the diversity.
<i>Suja</i> (Suja, 2021)	Proposed levy flight-based moth flame optimization algorithm	Improved the global searching capability of the MFO algorithm and maintained the stability of the system

Appendix I

<i>Xia et al.</i> (Xia et al., 2021)	Proposed generalized oppositional learning-based MFO with crossover strategy	Increased diversity of the population, obtained high-quality moths in initialization and improved the exploitation and exploration capability of the MFO algorithm
<i>Bandopadhyay and Roy</i> (Bandopadhyay & Roy, 2020)	Used hybrid multi-objective moth flame optimization (HMOMFO) method, and also integrated PSO and levy flight strategy with the MFO algorithm	Enhanced exploitation phase without compromising exploration phase and obtained the better candidate solution
<i>Cui et al.</i> (Zhiling Cui et al., 2020)	Used modified moth flame optimization algorithm with adaptive Lévy-Flight perturbations	Improved the ability of global search and provided excellent performance in terms of escaping from the local minima
<i>Dash et al.</i> (Dash et al., 2020)	Proposed hybridization of JAYA algorithm and MFO algorithm	JMFO algorithm showed faster convergence towards the solution
<i>Dash et al.</i> (Dash et al., 2020a)	Proposed a hybridization of moth flame optimization, ant lion optimization, and salp swarm algorithm	Improved significantly in performance as compared to ALO, MFO, and SSA. The suggested hybrid method enabled the exploration of the capabilities of each algorithm.
<i>Elattar et al.</i> (Elattar & Elsayed, 2020)	Proposed a new equation for reducing the flame number and convergence constant	Provided an effective balance between the exploitation and exploration phase and sped up the convergence of the algorithm.
<i>Elaziz et al.</i> (Elaziz et al., 2020)	Used opposition learning theory for initialization	Escaped from entrapment in local optima and accelerated the global convergence speed
<i>Fei et al.</i> (Fei et al., 2020)	Proposed the hybridization of fuzzy c-means (FCM) and the MFO method.	Provided higher convergent speed and thus reduced the energy consumption of the network.

Appendix I

Helmi and Alenany (Helmi & Alenany, 2020)	Used Gaussian mutation-based MFO, Lévy-flight based MFO, and Cross MFO (COMFO), an algorithm based on permutation-based problems (PBP)	Escaped from local minima regions and premature convergence
Kaur et al. (Kaur et al., 2020)	Added Cauchy distribution function, and influence of best flame (changes in moth updating equation)	Modification ensured the balancing between exploration and exploitation and also helped in improving the convergence speed
Korashy et al. (Korashy et al., 2020)	To increase the performance of the MFO algorithm, leadership hierarchy of the grey wolf optimizer (GWO) algorithm was proposed	Minimized the total operating time of relays resulting in the improved optimal solution by utilizing the proposed MFO algorithm
Li Yu et al. (Li Yu et al., 2020)	Introduced two new approaches (Lévy flight and dimension by-dimension evaluation) in the MFO algorithm	Improved global exploration capability and convergence speed and also provided an effective balance between the local and global search
Lin et al. (Lin et al., 2020)	Used modified position updating equation based on inertia weight and applied Cauchy mutation strategy for individuals of the moth population	Improved convergence rate and found the optimal global solution with a greater probability
Pelusi et al. (Pelusi et al., 2020)	Changed the moths' positions according to their corresponding flames, and the best flame is as follows: $M_i^k = D_i^{k-1} \cdot e^{bt} \cdot \cos(2\pi t) + w \cdot F_i^{k-1} + (1 - w) \cdot M_{best}$	Improved exploration and exploitation phase
Reddy and Bojja (Reddy & Bojja, 2020)	Used hybridization of MFO and teaching learning-based optimization (TLBO) algorithm	Improved the proficiency of exploration in MFO algorithm and competency of exploitation in TLBO algorithm
Sayed et al. (Sayed et al., 2020)	Proposed binary version of both moth flame optimization and whale optimization algorithm	Gave the best selection of the features using binary MFO algorithm in determining clustering while finding the highest results regarding external

Appendix I

		validity measurements using binary WOA
Sharma and Saha (R. Sharma & Saha, 2020)	Used hybridization of firefly and moth flame algorithm	Produced less redundant and reduced test paths as compared to FA and MFO algorithm
Yu et al. (Yu et al., 2020)	Proposed simulated annealing based MFO	Increased the benefit of the MFO algorithm in the local exploitation process.
Zhang Z. et al. (Zhang Z. et al., 2020)	Used updated formula, the inspiration of moth linear flight path and flame population update strategy	Enhanced the ability of the Moth-flame optimization algorithm (MFO) to overcome falling into the local optimum
Zhang H. et al. (Zhang H. et al., 2020)	Proposed Nelder-Mead simplex (NMS) and orthogonal learning (OL) schemes-based moth flame optimization algorithm	Provided better accuracy and fast convergence rate
Zhao et al. (Zhao et al., 2020)	Flames were generated by orthogonal opposition-based learning	Improved effectiveness of algorithm and provided better performance in terms of balancing exploration and exploitation capabilities as well as avoiding local optima
Buch and Trivedi (Buch & Trivedi, 2019)	Introduced adaptive moth flame optimization algorithm that included the step size on the basis of the best and worst position of moths as well as the current position of moth	Achieved the optimal solution with a faster convergence rate
Hongwei et al. (Hongwei et al., 2019)	Proposed Chaos enhanced MFO algorithm. Chaotic maps used in initialization population and boundary handling components	Increased the diversity of the population
Jain and Saxena (Jain & Saxena, 2019)	Used opposition learning theory for initialization	Escaped from entrapment in local optima and improved convergence rate

Appendix I

<i>Jia et al.</i> (Jia et al., 2019)	Used thresholding heuristic scheme that was included in MFO algorithm, and also applied the inertia weight in moths position updating equation	Improved convergence speed, accuracy, stability, and efficiency,
<i>Khalilpourazari and Khalilpourazary</i> (Khalilpourazari & Khalilpourazary, 2019)	Proposed the hybridization of moth flame optimization algorithm (MFO) and water cycle algorithm (WCA)	MFO algorithm was used to enhance exploitation, and the WCA algorithm was used to enhance exploration
<i>Li et al.</i> (Li et al., 2019)	Changed in moth updating equation for improvement	Increased the convergence speed and improved the search accuracy
<i>Luo et al.</i> (Luo et al., 2019)	Used elite opposition-based learning in the MFO algorithm	Enhanced diversity of the population and its exploration ability
<i>Rashid et al.</i> (Rashid et al., 2019)	Improved MFO algorithm by a different selection of flames	Improved the ability to guide the search direction towards a better solution
<i>Sapre and Mini</i> (Sapre & Mini, 2019)	Moth flame optimization with Cauchy mutation and evolutionary boundary constraint handling was presented	Enhanced the capabilities of exploration and exploitation, avoided entrapment in local optima, and also increased convergence rate
<i>Sheng et al.</i> (Sheng et al., 2019)	The double flames generation scheme was used for producing two distinct forms of target flames to guide the flight of moths, namely, local flame and global flame	Improved global exploration and local exploitation
<i>Singh et al.</i> (Singh et al., 2019)	Proposed hybridization of Artificial neural network (ANN) and MFO algorithm	Overcome the limitations of conventional training methods, including poor convergence speed and local optima stagnation
<i>Taher et al.</i> (Taher et al., 2019)	Used Archimedean and Hyperbolic spiral path	Attained optimal power flow solution with fast convergence
<i>Wu et al.</i> (Wu et al., 2019)	Used the hybridization of the Levy-flight scheme and the moth flame optimization algorithm	Improved the global optimization ability and converged fast to get the best solution

Appendix I

<i>Xu Y. et al.</i> (Xu Y. et al., 2019)	Proposed gaussian mutation and chaotic based MFO	Increased population diversity and improved solution quality and convergence speed
<i>Xu Y. et al.</i> (Xu Y. et al., 2019a)	Used Lévy mutation (LM), Gaussian mutation (GM), Cauchy mutation (CM) in the basic MFO algorithm	Improved neighborhood informed capability, global exploration ability, and enhanced the randomness of search agents' movement
<i>Elsakaan et al.</i> (Elsakaan et al., 2018)	Used the advantages of the original MFO algorithm and levy flight strategy	Increased the diversity of the population
<i>Jangir and Trivedi</i> (Jangir & Trivedi, 2018)	Used non-dominated sorting moth flame optimizer (NSMFO)	Proved effectiveness on the basis of generalized distance and efficiency in terms of execution time and obtained the optimal solution with faster convergence
<i>Kamalapathi et al.</i> (Kamalapathi et al., 2018)	Used hybridization of Moth flame optimization and fuzzy logic controller	Improved the dynamic behavior of the Brushless DC motor in terms of supply voltage and line current harmonics and examined the effectiveness of the proposed technique
<i>Li C. et al.</i> (Li C. et al., 2018)	Used dynamic flame guidance (DFG) and differential evolution flame generation (DEFG) techniques	Improved convergence speed and global search capability
<i>Li W. K. et al.</i> (Li W. K. et al., 2018)	Integrated two schemes, namely an indicator-based selection scheme and an opposition learning-based scheme in the MFO algorithm	Maintained diversity and accelerated the convergence
<i>Reddy et al.</i> (Reddy et al., 2018)	Used the various ways for the selection of flames to update the position of the last remaining moths	Improved exploitation capabilities of MFO approach, and improved solving UC problem

Appendix I

Sayed and Hassanien (Sayed & Hassanien, 2018)	Proposed the hybridization of simulated annealing and moth flame optimization algorithm	Overcome the limitations of the original MFO algorithm, such as slow convergence rate and low chances to obtain the optimal solution
Xu L. et al. (Xu L. et al., 2018)	Proposed improved version of moth-flame optimization algorithm based on gaussian mutation and cultural learning strategy	Enhanced exploitation and exploration performances
Anfal and Abdelhafid (Anfal & Abdelhafid, 2017)	Used hybridization of PSO and MFO algorithm	Obtained the best solution by reducing the number of PMU installations and increasing measurement redundancy.
Aziz et al. (Aziz et al., 2017)	Used moth-flame optimization (MFO) algorithm and whale optimization algorithm (WOA)	The results obtained from MFO were superior to WOA, as well as it provided an effective balance between exploitation and exploration in each picture
Bhesdadiya et al. (Bhesdadiya et al., 2017)	Used the hybridization of particle swarm optimization (PSO) and moth flame optimization (MFO) algorithms.	MFO algorithm was used for enhancement in exploration while PSO algorithm was utilized for enhancement in exploitation
Gholizadeh et al. (Gholizadeh et al., 2017)	A new equation was proposed for updating the position based on the best knowledge gained by the search agents	Decreased the probability of trapping into local optima and improved the convergence speed
Hassanien et al. (Hassanien et al., 2017)	Reported a hybridization of moth flame optimization and rough set theory for feature selection (MFORSFS)	Provided better performance by choosing the best features describing the tomato's leaf in terms of robustness and convergence speed
Jangir (Jangir P., 2017)	Proposed Hybrid Algorithm Particle Swarm Optimization-Moth Flame Optimizer (HPSO-MFO) method	Improved convergence speed, also addressed power flow concerns (reduced fuel cost, enhanced voltage profile, and voltage stability, minimized active and reactive power losses)

Appendix I

Savsani and Tawhid (Savsani & Tawhid, 2017)	A non-dominated moth flame optimization technique was proposed	The efficacy of the algorithm was tested using multi-objective benchmark functions and distinctive features, and the diversity among the best solutions was maintained.
Wang et al. (M. Wang et al., 2017)	Used kernel extreme learning machine based on the chaotic MFO technique	Improved convergence speed and avoids trap into the local optimum.
Yang et al. (W. Yang et al., 2017)	An effective approach for short-term load forecasting in power system was proposed	Enhanced performance by minimizing the original data's inherent complexity
Li et al. (Li et al., 2016)	Used Levy-flight scheme in the moth flame optimization algorithm	Obtained a better trade-off between exploration and exploitation, making the algorithm more efficient and faster
Soliman et al. (Soliman et al., 2016)	Used Archimedean and Hyperbolic spiral path	Improved the searching capability and convergence speed
Vikas and Nanda (Vikas & Nanda, 2016)	Exploration and exploitation phases of the MFO algorithm were used for archive grid, sorting, and non-dominance of solutions	Achieved the best solution and ensured effective convergence